Towards Safe, Realistic Testbed for Robotic Systems with Human Interaction

Bhoram Lee[1], Jonathan Brookshire[1,2], Rhys Yahata[3], and Supun Samarasekera[1]

[1]*Center for Vision Technologies, SRI International (e-mail:* <first>.<last>@sri.com*).*
[2]*LatentAI (e-mail:* jon@latentai.com*)*
[3]*University of Southern California, Institute for Creative Technologies (e-mail:* ryahata@ict.usc.edu*)*

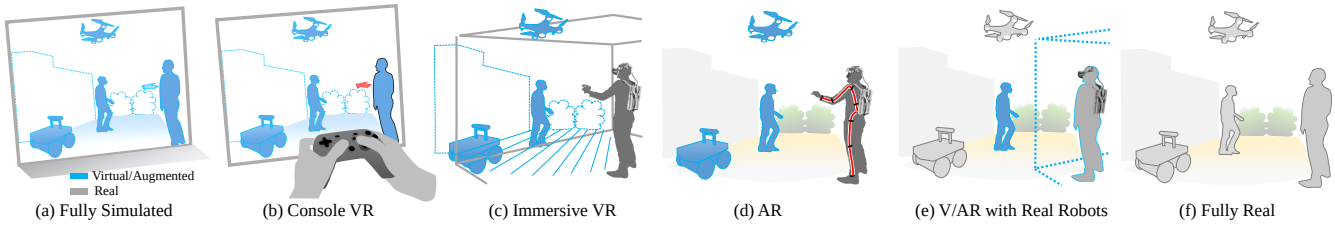

Fig. 1: Multi-Robot System test environments with Mixed-Reality interfaces with varied realism of human behavior. Virtual components are illustrated in blue (best viewed in color): (a) Fully simulated environment, (b-e) human characters are controlled by real humans (via (b) console-based Virtual Reality (VR) interfaces, (c) immersive VR interfaces, (d) Augmented Reality (AR) interfaces, and (e) VR/AR with real robots in real environment, and (f) fully real environment.

***Abstract*—Simulation has been a necessary, safe testbed for robotics systems (RS). However, testing in simulation alone is not enough for robotic systems operating in close proximity, or interacting directly with, humans, because simulated humans are very limited. Furthermore, testing with real humans can be unsafe and costly. As recent advances in machine learning are being brought to physical robotic systems, how to collect data as well as evaluate them with human interactions safely yet realistically is a critical question. This paper presents a Mixed-Reality (MR) system toward human-centered development of robotic systems emphasizing benefits as a data collection and testbed tool. MR testbeds allow humans to interact with various levels of virtuality to maintain both realism and safety. We detail the advantages and limitations of these different levels of realism or virtualization, and report our MR-based RS testbed implemented using off-the-shelf MR devices with the Unity game engine and ROS. We demonstrate our testbed in a multi-robot, multi-person tracking and monitoring application. We share our vision and insights earned during the development and data collection.**

I. Introduction

Given the increasing adoption of robotic technologies in diverse areas, we can expect demands to grow for autonomous robots that can operate alongside humans to perform complex tasks such as transportation, surveillance, rescue, and disaster response. Although we have witnessed successful algorithms to control and operate robots, an open question still remains when it comes to their interaction with humans [1]. How can we develop and test robotic systems operating alongside humans in a safe, yet realistic, environment? Since the ultimate use of physical robots would happen in the real world, enabling them to learn to deal with humans is essential. However, it is challenging in two aspects. First, operating physical robotic systems with humans can be unsafe and costly. Second, human models in simulation are very limited. Even though simulation is a necessary, safe testbed for complex robotics systems, there will exist a large gap between development and reality if simulation is the only tool. Therefore, we need a means to reduce the disparity before deploying robots in real world. Efforts are being made to learn safe algorithms [2], but we still lack a good way to collect data of good realism and verify safety before deployment. Few studies have provided a systematic view on this matter.

In this paper, we present a mixed-reality testbed design for robotic systems as one solution to increase behavioral realism while safely verifying robotic performance directly using human input. After reviewing related past work in § II, we identify and describe the benefits and limitations of using MR interfaces for different levels of virtualization/realism in § III. Details of our MR testbed implementation, in both hardware and software, are provided in § IV. We present experimental demonstration of the system for example applications of multi-robot, multi-person tracking and activity detection in § V, and summarize our approach discussing future directions in § VI.

II. Related Work

Mixed-Reality, referring the virtuality continuum as in [3], is a broad topic that involves many different technologies and research fields. We currently experience challenges in leveraging MR in robotics due to the lack of generic frameworks and guidelines of MR user interfaces (UI) as well as limited technologies of MR. Nonetheless, the number of

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. 140D6319C00007.
The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

studies involving "robotics AND (MR or VR or AR)[1]" has increased at an exponential rate during the past couple of decades [4]. As more and more of the technological and UI challenges are overcome, the integration of MR with robotic systems is expected to become prevalent. We embrace the immaturity as a great potential to expand with the synergy of MR and robotics.

One important aspect of leveraging MR in robotics has been mediating communication between humans and robotic systems. It is usually difficult for a person to give direct commands or parameters in the input space of robotic systems, especially when the dimension is high (e.g., manipulators or multiple robots). MR has been considered promising and enabled intuitive visualization of spatial information for some time [5]. Various MR systems have been suggested to provide operators with an easier, higher-level, or multi-modal control over robot [6–10] compared to simple teleoperation without a MR component. In development phases, MR overlays of real and simulated scenes have provided intuitive programming and debugging environments [11–14]. The effectiveness of gathering human inputs via MR interfaces are also presented for human expert demonstrations in the robotic learning domain [15–17]. The types of communication are not restricted to supervisory purposes. More and more collaborative and interactive forms of MR-based communication have appeared in Human-Robot Interactions (HRI) studies [18–21].

With growing processing power and easy access to the off-the-shelf MR interfaces, robotics systems with MR components are increasingly common across various stages of development and deployment [22]. Another noticeable aspect of the trend is that the use of MR in robotics has been extended beyond data visualization. Researchers have suggested ideas for tight, yet flexible, integration of MR with complex robotics systems for development and test [23–26]. The flexibility of MR technology enables gradual integration of virtual and real components with robotic systems at lower risks of physical damages as pointed out in [27].

We can further exploit MR technology as a tool to collect data and test autonomous multi-robot systems with realistic human behaviors. A recent study [28] emphasized the importance of VR in the development of safety in autonomous vehicles. A specific implementation presented in [29] used a MR-based framework for verifying and validating an autonomous vehicle's performance in the presence of pedestrians. The idea was to create a virtual environment shared between the vehicle and human pedestrian. This way, the testbed provided a way to test algorithms under real human behavior. While appreciating the previous studies discussing and revealing the value of MR-based testbeds, we contribute a systematic view of MR-based robotics testbed for broader applications and share our implementation as guidelines.

III. MR for RS with Realism

The goal of this section is to identify pros and cons of potential scenarios of MR implementations as a testbed for RS to be deployed in the real world. In a MR environment, there are virtual and real entities that interact with each other, spatially mapped from physical to virtual or vice-a-versa. Varied ways to implement MR interfaces allow different levels of virtuality or reality as depicted in Fig. 1. We categorized six levels of virtuality/reality based on entities. The following paragraphs are organized to briefly define entities and the six configurations and then detail the features of each configuration, which are summarized in relative scales in Table I.

A. Entities and Configuration

There are three main entities—robots, humans, and the environment (Table I, col. 2-col. 4), and each can be either virtual or real. Virtual robots are widely used for software-in-the-loop (SITL) testing. Their sensor data is calculated using sensor models, and command output is used to simulate actuators. Real robots are physically present in the world, and can be tested alongside virtual entities using a hardware-in-the-loop (HITL) approach. Humans can interface with the system in several ways. Virtual humans mean programmed human characters. Real humans can be immersed in the virtual scene using Console, VR, and AR interfaces, or physically present in the real world. We define the environment to include anything except robots and characters (e.g., buildings, ground, etc.). Based on the combination of virtualization levels of the three entities, we may consider (a) fully simulated, (b)-(d) MR with virtual robots, (e) MR with real robots, and (f) fully real configurations.

B. Features

Now let us look into the 'feature' dimension of the configurations. Column 5 to 14 of Table I are the key aspects to consider in developing MR testbeds. The markings indicate how strong the features are for each configuration relative to others. An empty cell means the feature (column) is not available for the configuration (row).

Safety, Realism, and Cost: The first four items (Table I, col. 5 to 8) are related to humans. The foremost concern when deploying or testing robots near humans is safety (col. 5). Robots can physically harm people only when they share the same environment as in (f). We consider this as the final deployment, rather than a developmental, stage.

The second feature (col. 6), transfer [30], occurs when a human behaves in the virtual world as she/he would in the real world. Among the MR interfaces, a joystick usually provides the lowest level of transfer, and VR and AR modalities are capable of a better transfer. Photo-realism, higher frame rates, or audio effects may affect transfer as well. If the goal of a study is to extrapolate human behavior in the virtual world to the real world then a high level of transfer is important.

Column 7 and 8 are closely related to transfer. When attempting to leverage body motion in algorithms, many VR systems can be used to track limbs and the head. When accompanied by full body motion capture, AR systems can provide similar or better resolution. Many game engines enable character animation systems which can be leveraged by

[1]VR:Virtual Reality, AR: Augmented Reality

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)
	Entities			**Features**									
Configuration	Robots	Humans	Environment	Human safety	Transfer	Full-body motion capture fidelity	Multi-modal experience	Expense	Experimental repeatability	Robot perception fidelity	Rapid/remote data collection	Ground truth availablity	Requires aligned environments
(a)	Virt	Simulated	Virt	x				$	xx	x	xxx	xx	
(b)	Virt	Console-based	Virt	x	x		x	$	x	x	xx	x	
(c)	Virt	VR-based	Virt	x	xx	x	x	$$	x	x	x	x	
(d)	Virt	AR-based	Virt+Real	x	xxx	xx	x	$$$	x	x	x	x	x
(e)	Real	Simulated, console, AR, VR	Virt+Real	x	xxx	xx	x	$$$$		xx			x
(f)	Real	Real	Real		xxxx	xxx	xx	$$$		xxx			

TABLE I: Various configurations (column 1) correspond to Fig. 1. More marks indicate relative importance for that configuration. For example, configuration (e) with real robots and AR/VR hardware would typically cost more than a configuration (a) simulation-only setup.

console and simulated humans. Non-visual sensory modalities are hard to experience in the virtual environment (e.g. haptic or olfactory sense). Sound/voice can be captured and played in a limited scope with a microphone and headsets. While realistic sounds is hard to be provided with virtual entities (e.g. sound of flying drones), human voice can be easily obtained for console systems, VR, and AR systems (col. 8).

Overall, as real components increase (going from (a) to (f)), realism improves in terms of transfer and fidelity. Improved realism, however, comes with added cost and expert knowledge (col. 9). Depending on the development phase or the purpose of study, we may benefit from a proper choice of mixed-reality setting. For example, console interfaces, consisting of just a laptop, joystick, and headset are relatively inexpensive and easily deployable, compared with an AR system. Thus, it will be possible to recruit more human players with the same budget, yet at the cost of transfer quality.

Quantity and Quality of Data: The following four items (col. 10 to 13) are related to data we collect from the system.

The completely virtual approach of configuration (a) affords significant experimental repeatability (col. 10). This is very common and useful way to obtain a massive dataset for training as well as to test low-level algorithms especially in the initial stages of development. Data collection can be conducted faster than real time, possibly in parallel, without any spatial restriction (col. 12). This is often important when trying to assemble large datasets. In configurations (b)-(d) where the virtual characters are controlled by human players, the data collection process will not be faster than real time nor parallelizable. They also present logistic challenges (recruiting, facility management, Internal Review Board approvals, etc.) that slow the process. Nevertheless, data with human inputs are valuable and there are three ways to use it: First, the data can be added to the dataset for training or development. The data size would be small, yet this may be useful for refinement or meta-learning [31]. Second, the recorded human inputs may be used to improve the programmed human characters for configuration (a). Third, the data can be used for evaluation of the algorithm. As we envision continued learning problems of a longer term, all the usages of human data are meaningful.

Perception fidelity (col. 11) refers to how closely the module imitates or amplifies reality. Although synthesized sensor data is limited, depending on the fidelity of the simulator, it is possible to yield a high level of robot perception fidelity. If a high-fidelity simulator is employed, perception algorithms used in the real world can be used in simulation, and vice-a-versa [32, 33]. In configuration (e), a HITL configuration, uses real robot hardware infused with synthetic sensor data. This sensor data might be a real video feed overlaid with virtual entities (i.e., AR for the robot) or it might use completely synthetic sensor data.

One of the advantages with the virtual entities is ubiquitous ground truth information (col. 13). This information is useful while development as well as for quantitative evaluation of algorithms. Recorded human inputs via MR interfaces may contain measurements/observations that mismatches with the player's true intended behavior. Therefore, it is important to collect any report or feedback from human players together with data from devices.

Technical Consideration: Besides limitations of MR interfaces as UI, a technical consideration for the MR testbed where virtual and real entities are combined is that all entities must spatially align (col. 14). Photogrammetric reconstructions [34, 35] enable us to maintain alignment from the real environments to simulated virtual worlds. A low quality alignment may degrade the user experience of human players or alter robots' behavior in an unexpected way.

IV. System Implementation

We have implemented a MR-based RS testbed supporting configurations (a)-(d) presented in the previous section.[2] In this section, details of our system implementation are reported as a concrete successful example.

[2]The system can be extended to support (e) and (f) with real robots, but the current scope includes (a)-(d).

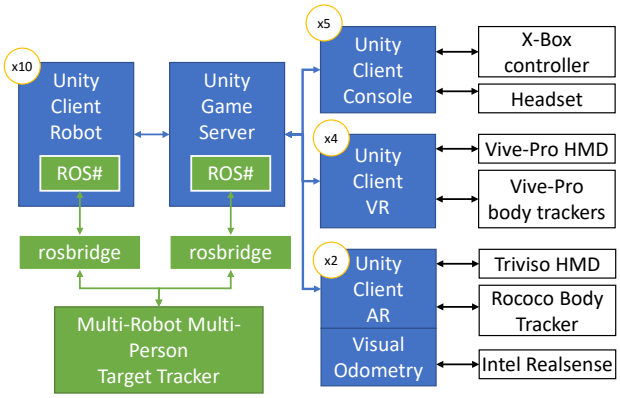

Fig. 2: The system architecture includes ROS (green) and Unity (blue) elements. We performed experiments up to ten robot clients, five console clients, four VR clients, and two AR clients. We also include several non-player characters (NPCs), thus mixing configurations (a)-(d) in Fig. 1 and Table I

Our system uses the Unity game engine [36] as our virtual environment. We make use of Unity's plug-in capability to add custom C# components that enable connections to a ROS [37] environment (Fig. 2). We use a mix of Windows and Linux machines, relying on ROS# [38] and rosbridge [39] to provide Unity-to-JSON and JSON-to-ROS, respectively, connections. The adoption of Unity with ROS is found in other VR HRI studies such as [40, 41]

Our architecture consists of a central Game Server and multiple clients connected to the server (Fig. 2). The game clients include robot clients, and Console/AR/VR clients. Additionally, we also implemented a special type of client that is not associated with an entity, but affords a third party the opportunity to move around the environment. We have virtual environments derived from photogrammetric reconstructions of parts of SRI's Princeton campus (spanning over $180\times120m^2$) and parts of Muscatatuck Urban Training Center, IN [34, 35].

A. Game Server

The Game Server is responsible for maintaining the game state, game time, and scenario/terrain loading. Our game code follows the standard model from Unity's First-Person Shooter (FPS) Sample [42]. We added several key features: (1) The Game Server publishes a single ground truth message at 10Hz that includes: 3D pose and twist of all characters and robots, skeletal pose and activity information of all characters, camera images as well as camera parameters, and more. (2) Autonomous non-player characters (NPCs) are controlled by behavior trees, allowing us to scale up the number of characters in our scene. (3) ROS replay capabilities allow us to examine the experiments afterwards. Also, captured body motions from the AR and VR players can be replayed on the NPCs. This replay capability gives us another way to add motion realism. (4) We add the ability to play prerecorded audio announcements from the robots and to communicate between human players. Incorporating the Dissonance VoIP package to emulate a half-duplex, hand-held radio network, users can select a radio channel and speak to each other.

Fig. 3: The Real/Virtual Environment: actual site (left) and its virtual replica (right) as a mesh model obtained after a prior mapping process.

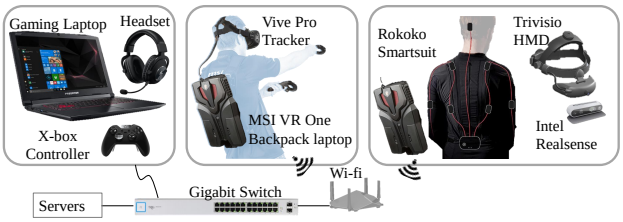

Fig. 4: Equipment and network connections for console (left), VR (middle), and AR (right).

B. Robot clients

Each robot subscribes to a ROS `RobotCmd` message through the rosbridge into JSON, and then into a C# class. The `RobotCmd` allows the publisher to control the robots. As our application was not focused on developing navigation, localization, or obstacle avoidance algorithms, we leveraged Unity's built-in NavMesh navigation system [43] to handle all three issues.

Each robot publishes a `RobotState` message, which includes its pose, twist, camera information, image stream (640x480, 10Hz), audio stream, and ground truth bounding boxes of all characters. We also perform a single ray cast between the center of each character and the camera to ensure that they are not entirely occluded. We use this ground truth to quantify our tracker's performance (see § V).

C. Console clients

The console-based interfaces (Fig. 4, left) include a gaming laptop running Windows 10, Xbox controller [44], and a headset. Using the controller, the player can walk, run, take a single-knee crouch, pickup/deposit an object, and talk on either of two radio channels. Full body motion (walking, running, crouching) was provided by Unity's Mecanim animation system. We were also able to use console interfaces over the Internet to continue testing remotely during COVID-19 closures. While being located in California, Denver, and New Jersey, the players experienced no noticeable lag.

D. VR clients

As suggested in Fig. 4, middle, VR players wear an MSI VR One [45] computational backpack running Windows 10 and a VIVE Pro HMD [46]. They hold a VIVE controller in each hand, and trackers are mounted on the front waist and left and right feet. The position and orientation of the player's head, hands, waist and feet are used as inverse kinematics (IK)

Fig. 5: Experiment with human players (five console players and two VR players are seen in the figure.)

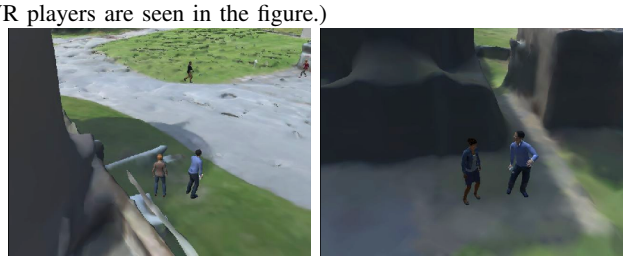

Fig. 6: Human players in the virtual environment captured from virtual UAVs. Each player is playing his/her role (left). Note that the character on the right (male, blue shirt) is in a natural pose, being controlled by a VR interface.

targets, which the IK system (FinalIK [47]) uses to solve the pose of the virtual character's skeleton. The players can freely walk around a 3.5m x 3.5m play area. To address the physical tracking space limit, typical joystick based locomotion is mapped to one of the VIVE controller touchpads.

E. AR clients

The AR interfaces (Fig. 4, right) are also wearable devices including a MSI VR One backpack for computing and a Rokoko Smartsuit [48] for full-body motion capture. The AR headset is a custom system consisting of a Trivisio HMD [49] paired with a physically mounted Intel Realsense D435 that provides video see-through capabilities. Video frames and IMU data from the D435 are fused with GPS via visual odometry [50] to provide head and position and orientation. While aligning the real/virtual worlds, we found that some locations in our 3D models had ground height errors which would cause the AR characters to "float" above the ground. To compensate, we clamp the character's feet to the virtual world's ground and adjust the height of the head pose accordingly.

V. Demonstration: Multi-Robot Multi-Person Tracking

In the previous sections, we reviewed how using MR benefits RS (§III) in general and how our MR testbed is implemented (§IV). Now, we present our demonstration of the system for a specific application, Multi-Robot Multi-Person Tracking. Note that this section is to showcase a specific example of the utility of the MR testbed. The application is still under development and our algorithms for planning, perception, and tracking are not the main contributions.

Let us briefly describe the problem of multi-robot multi-target tracking we consider. We assume several UAVs and several UGVs with perfect communication operate in a bounded, structured environment. The number of robots are smaller than the number of targets, and the numbers are fixed during an experiment. The robots move under a centralized planner to

Fig. 7: Unity presented our tracker with challenging viewing conditions including (from left to right) saturated images, pitched camera angles, low resolution, and partially occluded people.

Fig. 8: Sample tracking results from different views. The same players a tracked from different perspectives. "CO" and "VR" indicate a console and VR player, respectively.

survey the people equally and as much as possible. They actively search for and follow people by load balancing and switching people. The robot tasks also include monitoring target tracks and activities. Our tracking application has several requirements: (1) human safety, (2) short-term activity recognition which require full-body motion, (3) stressing our algorithms and system by varying the numbers of robots (10+) and people (40+) over an area larger than can be covered by static robots, and (4) collect data with real people for refinement and evaluation of the algorithms.

Our experiments include a combination of console-based, VR-based, and AR-based users. Thus there is no safety concern due to operating live robots near people. The IRB for this study was reviewed by Office of Integrity at SRI and approved by the Office of Research Protections at U.S Army Medical Research and Development Command. The shared environment used for the demonstration is shown in Fig. 3. The NPCs were programmed to randomly wander or travel among waypoints at random speeds.

A. Visual Tracking

We use the YOLO3 [51] to provide detection bounding boxes; we found the algorithm to perform reasonably well on the Unity-rendered characters (see Fig. 7). The bounding boxes update a short-term tracker, using the Hungarian algorithm for data association. A Kalman filter is used to filter the tracks with a constant velocity prediction. These short term "tracklets" can handle short tracking gaps.

If tracking gaps are sufficiently long, then a constant velocity assumption and geometry are insufficient to re-associate tracklets and reacquire people. Thus, we use visual features computed from multiple frames of each tracklet using a deep neural network. The normalized inner products of the feature vectors is used to assess the similarity of the people. The neural network has a Resnet50 [52] design with an output

of 2,000 features (this is then used as the re-acquisition feature vector). Using the ground truth positions and labels of all people in the scene, we evaluated our algorithm. On 40 minutes of data, the algorithm assigned the correct label to 13 people 93% of the time (see Fig. 8 for two example frames).

B. Activity Recognition

Our tracking application also classifies people's short-term activities, including walking, running, turning, standing, sitting, throwing, holding objects, putting on a backpack, lifting objects, and kneeling. To ease the training data requirements, we use a skeleton-based activity recognition, Openpose [53], to extract a low-dimensional description (the location of joints, hands, feet, eyes, ears and nose) of each person. Activities are inferred from time histories skeletons. Activity classification is performed by a modified version of Spatio-Temporal Graph Convolution Neural Network (ST-GCN) [54], a deep but relatively compact neural network for activity recognition.

Although some activities were available from HMDB [55], public datasets for the holding binoculars, backpack, lifting, and kneeling activities were unavailable. Instead, we used the AR and full-body recording capabilities to collect additional data. We had 6 participants perform 4 repetitions of each activity and record the skeletons in ROS. We then expanded this small dataset by generating training images by randomly selecting a virtual location, a camera angle/position, and a character avatar, and we replayed our captured body motions. The final classification accuracy was comparable to the original HMDB activities, between 76%-94%. Because our network is trained only on skeletons, which use the same real and virtual representation, we expect this method to transfer to the real world.

C. Statistics of Human Behavior

Leveraging human data from our experiments, we were able to fine-tune algorithms of tracking, searching, and following tasks. For example, Fig. 9 shows the ground truth distribution of how far people travelled from their current position after 10 and 20 seconds during the experiment. This implies that the last-known position is still the most likely location to find a person in a short term. Based on this, when a new person is assigned to a robot and the person's location is unknown, it uses the person's last known position as a starting point and begins to radially search outward until they are found or a timeout occurs.

In another experiment, we performed a simple system test involving interactions between robots and humans. Specifically, random players were provided direction to a new location by the audio announcement from the robot. In this test, some players had been assigned a focused task (e.g., keeping a point of interest under surveillance), while others were given more general direction (e.g., plow the field). The players were not instructed how to respond to the robot's requests. Fig. 10 shows the statistics of how the two groups reacted: the tasked players were more likely to ignore the request while the general players were more likely to follow.

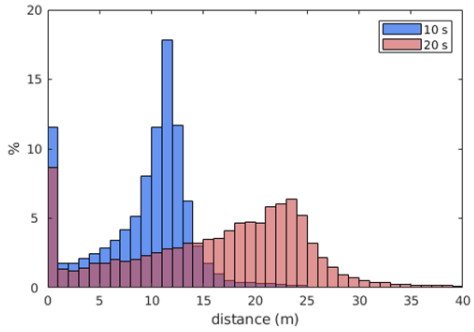

Fig. 9: Distribution of ground truth distance travelled from last-known position. Distance informs tracker, search, and follow algorithm tuning. This data was collected from real people interacting with the system via console, VR, and AR interfaces.

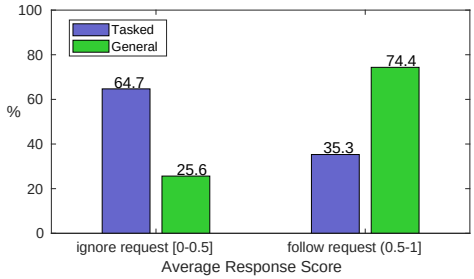

Fig. 10: Players with focused, specific tasks were more likely to ignore the robot's request to move from the area, than players with more general assignments.

VI. Discussion

We have presented an approach toward safe yet realistic MR-based testbed for robotic Systems with human interactions. We have identified six configurations and provided a summary of their features and requirements, which may serve as a concise guideline for readers to choose and develop their MR testbed. We have also reported our MR-based RS testbed system architecture in detail, and our experimental demonstration of the system for a multi-robot multi-person tracking and monitoring application.

While developing our system, we also have seen limiting aspects of the MR technology especially regarding the user's experience within the virtual environment. For example, some users kept adjusting the wearable interfaces resulting in unwanted gestures. Some users of VR systems do experience motion sickness. Any study which uses VR/AR HMDs would need to be prepared to recognize the symptoms and accommodate users with motion sickness or fatigue. Another point of concern is that the VR systems and our custom AR system are not easy to sanitize to be shared among multiple users.

Our study was driven by a safety requirement and a need to have real humans interact with the system: MR offered the best of both worlds. We believe that MR testbeds will become increasingly common given a need for close interaction between robots and humans, increasing levels of autonomy, increasing demands for data with human inputs, and less expensive VR/AR hardware.

REFERENCES

[1] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, 2019.

[2] M. El-Shamouty, X. Wu, S. Yang, M. Albus, and M. F. Huber, "Towards safe human-robot collaboration using deep reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4899–4905, 2020.

[3] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE Transanctions on Information and Systems*, vol. 77, no. 12, pp. 1321–1329, 1994.

[4] D. Szafir, "Mediating human-robot interactions with virtual, augmented, and mixed reality," in *International Conference on Human-Computer Interaction*, pp. 124–149, 2019.

[5] G. C. Burdea, "Invited review: the synergy between virtual reality and robotics," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 400–410, 1999.

[6] P. Milgram, S. Zhai, D. Drascic, and J. Grodski, "Applications of augmented reality for human-robot communication," in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 1467–1472, 1993.

[7] J. Rofmann, "Virtual reality as a control and supervision tool for autonomous systems," in *Proceedings of the International Conference of Intelligent Autonomous Systems*, p. 344, 1995.

[8] A. Yew, S. Ong, and A. Nee, "Immersive augmented reality environment for the teleoperation of maintenance robots," *Procedia CIRP*, vol. 61, pp. 305–310, 2017.

[9] B. Huang, D. Bayazit, D. Ullman, N. Gopalan, and S. Tellex, "Flight, camera, action! using natural language and mixed reality to control a drone," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6949–6956, 2019.

[10] J. Aleotti, G. Micconi, S. Caselli, G. Benassi, N. Zambelli, M. Bettelli, and A. Zappettini, "Detection of nuclear sources by uav teleoperation using a visuo-haptic augmented reality interface," *Sensors*, vol. 17, no. 10, 2017.

[11] F. Cao and B. Shepherd, "Mimic: a robot planning environment integrating real and simulated worlds," in *Proceedings of the IEEE International Symposium on Intelligent Control*, pp. 459–464, 1989.

[12] T. Pettersen, J. Pretlove, C. Skourup, T. Engedal, and T. Lokstad, "Augmented reality for programming industrial robots," in *IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, pp. 319–320, 2003.

[13] H. Liu, Y. Zhang, W. Si, X. Xie, Y. Zhu, and S.-C. Zhu, "Interactive robot knowledge patching using augmented reality," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1947–1954, 2018.

[14] B. Hoppenstedt, T. Witte, J. Ruof, K. Kammerer, M. Tichy, M. Reichert, and R. Pryss, "Debugging quadrocopter trajectories in mixed reality," in *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pp. 43–50, 2019.

[15] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, 2018.

[16] M. B. Luebbers, C. Brooks, M. J. Kim, D. Szafir, and B. Hayes, "Augmented reality interface for constrained learning from demonstration," in *Proceedings of the International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*, pp. 11–14, 2019.

[17] A. Jackson, B. D. Northcutt, and G. Sukthankar, "The benefits of immersive demonstrations for teaching robots," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 326–334, 2019.

[18] M. Dragone, T. Holz, and G. M. O'Hare, "Using mixed reality agents as social interfaces for robots," in *IEEE International Symposium on Robot and Human Interactive Communication*, pp. 1161–1166, 2007.

[19] S. A. Green, M. Billinghurst, X. Chen, and J. G. Chase, "Human-robot collaboration: A literature review and augmented reality approach in design," *International journal of advanced robotic systems*, vol. 5, no. 1, p. 1, 2008.

[20] S. Tellex, R. Knepper, A. Li, D. Rus, and N. Roy, "Asking for help using inverse semantics," 2014.

[21] M. Walker, H. Hedayati, J. Lee, and D. Szafir, "Communicating robot motion intent with augmented reality," in *ACM/IEEE International Conference on Human-Robot Interaction*, pp. 316–324, 2018.

[22] Z. Makhataeva and H. A. Varol, "Augmented reality for robotics: A review," *Robotics*, vol. 9, no. 2, p. 21, 2020.

[23] A. H. Göktoğan and S. Sukkarieh, "An augmented reality system for multi-uav missions," in *the Simulation Conference and Exhibition, SimTect*, 2005.

[24] I. Y.-H. Chen, B. MacDonald, and B. Wunsche, "Mixed reality simulation for mobile robots," in *IEEE International Conference on Robotics and Automation*, pp. 232–237, 2009.

[25] I. Y. H. Chen, B. Macdonald, and B. Wuensche, "A flexible mixed reality simulation framework for software development in robotics," 2011.

[26] M. Jakob, M. Pechoucek, M. Cap, P. Novák, and O. Vanek, "Mixed-reality testbeds for incremental development of hart applications," *IEEE Intelligent Systems*, vol. 27, no. 2, pp. 19–25, 2012.

[27] W. Hoenig, C. Milanes, L. Scaria, T. Phan, M. Bolas, and N. Ayanian, "Mixed reality for robotics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5382–5387.

[28] A. M. Nascimento, A. C. M. Queiroz, L. F. Vismari, J. N. Bailenson, P. S. Cugnasca, J. B. C. Junior, and J. R. de Almeida, "The role of virtual reality in autonomous vehicles' safety," in *IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pp. 50–507, IEEE, 2019.

[29] M. R. Zofka, S. Ulbrich, D. Karl, T. Fleck, R. Kohlhaas, A. Rönnau, R. Dillmann, and J. M. Zöllner, "Traffic participants in the loop: A mixed reality-based interaction testbed for the verification and validation of autonomous vehicles," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3583–3590, 2018.

[30] F. Rose, E. Attree, B. Brooks, D. Parslow, P. Penn, and N. Ambihaipahan, "Training in virtual environments: Transfer to real world tasks and equivalence to real task training," *Ergonomics*, vol. 43, pp. 494–511, 05 2000.

[31] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*, pp. 1126–1135, PMLR, 2017.

[32] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," *CoRR*, vol. abs/1605.06457, 2016.

[33] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," *CoRR*, vol. abs/1812.07252, 2018.

[34] B. C. Matei, H. S. Sawhney, S. Samarasekera, J. Kim, and R. Kumar, "Building segmentation for densely built urban regions using aerial lidar data," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008.

[35] R. Hadsell, B. Matei, G. Salgian, A. Das, T. Oskiper, and S. Samarasekera, "Complex terrain mapping with multi-camera visual odometry and realtime drift correction," in *International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*, pp. 493–500, 2012.

[36] J. K. Haas, "A history of the unity game engine," 2014.

[37] "Robotic Operating System." https://www.ros.org. Last Accessed: 2021-9-13.

[38] "ROS#." https://github.com/siemens/ros-sharp. Last Accessed: 2021-9-13.

[39] C. Crick, G. Jay, S. Osentosiki, B. Pitzer, O. C. Jenkins, and C. Crick, "Rosbridge: Ros for non-ros users," in *International Symposium on Robotics Research (ISRR*, 2011.

[40] M. Mara, K. Meyer, M. Heiml, H. Pichler, R. Haring, B. Krenn, S. Gross, B. Reiterer, and T. Layer-Wagner, "CoBot Studio VR: a virtual reality game environment for transdisciplinary research on interpretability and trust in human-robot collaboration," 2021.

[41] T. Inamura and Y. Mizuchi, "Sigverse: A cloud-based vr platform for research on multimodal human-robot interaction," *Frontiers in Robotics and AI*, vol. 8, 2021.

[42] "FPS Sample." https://unity.com/fps-sample. Last Accessed: 2021-9-13.

[43] "Building a NavMesh." https://docs.unity3d.com/Manual/nav-BuildingNavMesh.html. Last Accessed: 2021-9-13.

[44] "Microsoft Xbox." https://www.xbox.com. Last Accessed: 2021-9-13.

[45] "MSI VR One Backpack - MR READY." https://mr.msi.com/Backpacks/vrone. Last Accessed: 2021-9-13.

[46] "VIVE Pro." https://www.vive.com. Last Accessed: 2021-9-13.

[47] "FinalIK." https://assetstore.unity.com/packages/tools/animation/final-ik-14290. Last Accessed: 2021-9-13.

[48] "Rokoko - Motion Capture System - Smartsuit Pro." https://www.rokoko.com. Last Accessed: 2021-9-13.

[49] "Triviso SXGA61." https://www.trivisio.com/hmd-nte. Last Accessed: 2021-9-13.

[50] T. Oskiper, S. Samarasekera, and R. Kumar, "Multi-sensor navigation algorithm using monocular camera, imu and gps for large scale augmented reality," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 71–80, 2012.
[51] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.
[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.
[53] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2019.
[54] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," 2018.
[55] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *IEEE International Conference on Computer Vision (ICCV)*, 2011.