Mixed Reality for Robotics

Wolfgang Hönig[1], Christina Milanes[1], Lisa Scaria[1], Thai Phan[2], Mark Bolas[2], Nora Ayanian[1]

Abstract— **Mixed Reality can be a valuable tool for research and development in robotics. In this work, we refine the definition of Mixed Reality to accommodate seamless interaction between physical and virtual objects in any number of physical or virtual environments. In particular, we show that Mixed Reality can reduce the gap between simulation and implementation by enabling the prototyping of algorithms on a combination of physical and virtual objects, including robots, sensors, and humans. Robots can be enhanced with additional virtual capabilities, or can interact with humans without sharing physical space. We demonstrate Mixed Reality with three representative experiments, each of which highlights the advantages of our approach. We also provide a testbed for Mixed Reality with three different virtual robotics environments in combination with the Crazyflie 2.0 quadcopter.**

I. INTRODUCTION

When robots operate in shared environments with humans, they are expected to behave predictably, operate safely, and complete the task even with the uncertainty inherent with human interaction. Preparing such a system for deployment often requires testing the robots in an environment shared with humans in order to resolve any unanticipated robot behaviors or reactions, which could be potentially dangerous to the human. In the case of a multi-robot system, uncertainty compounds and opportunities for error multiply, increasing the need for exhaustive testing in the shared environment but at the same time increasing the possibility of harm to both the robots and the human. Finally, as the number of components of the system (humans, robots, etc.) increases, controlling and debugging the system becomes more difficult.

Allowing system components to operate in a combination of physical and virtual environments can provide a safer and simpler way to test these interactions, not only by separating the system components, but also by allowing a gradual transition of the system components into shared physical environments. Such a *Mixed Reality* platform is a powerful testing tool that can address these issues and has been used to varying degrees in robotics and other fields. In this work, we introduce Mixed Reality as a tool for multi-robot research and discuss the necessary components for effective use. We will demonstrate three practical applications using different simulators to showcase the benefits of the Mixed Reality approach to simulation and development.

The central contribution of this work is to establish Mixed Reality as a tool for research in robotics. To that end, we redefine Mixed Reality, and identify and describe the benefits of using Mixed Reality in robotics and multi-robot systems. We present novel use-cases which show the capabilities and benefits of Mixed Reality. A secondary contribution of this work is to provide a testbed for Mixed Reality multi-robot research using small UAVs such as Bitcraze Crazyflie 2.0 [1], and provide open-source access to relevant source code to enable other researchers to build on top of our model.

II. RELATED WORK

While we will refine this definition in the following section, the first published definition of *Mixed Reality* (MR) was given by Milgram and Kishino as the merging of physical and virtual worlds [2]. In their definition, *Augmented Reality* (AR) and *Augmented Virtuality* (AV) are seen as special instances of MR. In Augmented Reality, virtual objects are projected onto the physical environment, while in Augmented Virtuality, physical objects are incorporated into a virtual environment.

AR and AV have been used to help overcome challenges faced in implementing robotic systems. AR systems have been implemented in the form of a video overlay feed for a multi-robot system to display state information and internal data [3], [4]. With the addition of an overhead camera to do tracking, the pose of the physical robots can be incorporated into a virtual environment. Such a system has been suggested for educational purposes or to simplify debugging between simulation and practical experiments [5].

An implementation closer to a true MR approach merges virtual and physical sensor readings, allowing a robot to sense physical and virtual objects at the same time [6]. This approach allows testing a physical robot in unknown environments and simplifies the addition of obstacles.

Another unique usage of MR is in robot teleoperation. Freund and Rossman report a system that allows a human operator to manipulate objects in a virtual environment, which are then translated and executed by a robot in a physical environment [7], while [8] outlines a system which enables head-coupled virtual reality viewing. Similarly, it is possible to use a MR approach for so called tele-immersive environments. Such a setup allows humans to collaborate even if they are in different physical spaces [9]. While this previous work discusses human-human interaction only, we will show that it is beneficial to extend this use-case specifically to include robots.

In Milgram and Kishino's broad definition of Mixed Reality, anything containing pure or virtual elements may be considered as MR. Prior work in MR focuses on solving specific subproblems by incorporating physical or vir-

---

[1] Authors are with the Department of Computer Science, University of Southern California, USA {`whoenig, memilane, lscaria, ayanian`}`@usc.edu`

[2] Authors are with the Institute for Creative Technologies, University of Southern California, USA {`tphan, bolas`}`@ict.usc.edu`

This work was supported in part by ARO W911NF-14-D-0005 and ONR N00014-14-1-0734.

tual components to enhance a singular physical or virtual workspace. We propose instead that a Mixed Reality system should allow bidirectional feedback between multiple virtual and/or physical environments. This would allow combining the advantages of both virtual and physical environments in a real-time setting. By presenting several use-cases of an MR workflow, we will show the usefulness of such complete framework in robotics research.

III. MIXED REALITY COMPONENTS

Mixed Reality creates a space in which both physical and virtual elements co-exist, allowing for easy interaction between the two. Rather than one space being secondary to another (like in Augmented Reality and Augmented Virtuality), our take on MR blurs the boundaries between environments, creating one larger space where components from both worlds can communicate in real-time. This enables elements in one world to react directly to what is happening in another via direct data communication as opposed to a reconfiguration or modification of existing components.

MR is often known by its applications in virtual reality. However, MR can be particularly useful in robotics applications, as it creates a platform that allows for flexible development and testing of control algorithms. Since users can select which elements are physical and which are virtual, a number of different experiments can be performed based on various constraints and environments.

We tighten Milgram and Kishino's broad definition of MR by specifying its properties and components. Following the definition of Augmented Reality by Azuma *et al.* [10], we define a *Mixed Reality system* as one that:

- combines physical objects in at least one physical environment and virtual objects in at least one virtual environment;
- runs interactively (often called real-time); and
- spatially maps physical and virtual objects to each other.

The relationship between objects can be shown schematically with the general form given in Fig. 1. We will use this schema to describe our applications. Lines symbolize links between objects in the different environments.

Note that the combination of objects can be in any of the two types of environments (physical or virtual) or both. Additionally, there is no restriction on the number of physical or virtual environments; for example, it is possible to bring together physical objects in distinct locations or to use several different virtual environments.

A. Physical Environment

The physical world in a MR space includes objects such as people, robots, sensors, and obstacles. Those objects are able to interact with each other and with elements from the virtual environment. The environments themselves can be protected, closed environments to accommodate physical elements that pose a safety hazard, or allow for the installation of special equipment (such as a motion capture system) necessary to complete the physical-to-virtual communication link. Because of the direct communication of data between the physical and virtual environments, no changes or adjustments to physical-world components (e.g. the robot's camera and sensors) need to be made.

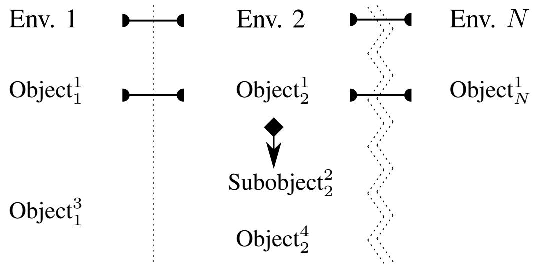


Fig. 1. Mixed Reality merges different environments (physical and virtual). The environments and their objects are aligned with each other. In this case, Object^1 is at least partially available in three environments, e.g. the physical world and two virtual worlds. Environment 2 enhances Object^1 by adding a subobject (Subobject$_2^2$, e.g. a sensor). Object^3 and Object^4 are visible in their respective environments only.

B. Virtual Environment

The virtual environment of a MR system can help overcome limitations of the physical environment and can be created using a variety of existing software such as robotic simulators or 3D game engines. Like the physical environment, the virtual environment can include robots and sensors as well as simulations of more complex objects. Because MR makes direct interaction between the virtual and physical worlds possible, there is much flexibility on which elements exist in the physical world and which can exist in the virtual world. Components can be chosen to be physical or virtual based on user needs and convenience.

C. Physical-Virtual Interaction

The defining feature of MR is its ability to allow for direct interaction between physical and virtual environments as well as multiple physical environments in different locations. Achieving full communication requires synchronization between environments.

1) Physical To Virtual: An isotropic mapping function can be used to map physical environments to the virtual ones. This might be any homeomorphism, such as scaling. This requires knowledge of the pose of all physical objects, thus external localization or self-localization is needed.

2) Virtual To Physical: Synchronizing the virtual environments to the physical ones is useful to show the state of the virtual world in the physical environment. Additionally, it can be used to map invisible virtual entities, such as a virtual goal location, to a physical environment. In practice, this is much easier to realize than physical-to-virtual mapping, because the full pose information is already known.

IV. BENEFITS OF MIXED REALITY

The interactive nature between different environments gives MR several advantages over AR or AV, including but not limited to the following.

Spatial flexibility. Interaction between physical and virtual environments in MR allows experiments with robots to be performed remotely. This can expand collaboration between groups, as they are no longer limited by geographic constraints and can meet in a centralized virtual environment.

(a) MR Components (b) Physical World (c) Virtual World (UNITY 3D)

Fig. 2. Two virtual humans simulated by SMARTBODY are followed by two quadcopters. The quadcopters are flying in a motion capture area and are synchronized with their virtual counterparts. This allows us to safely test algorithms while taking quadcopter dynamics into account.

Spatial flexibility also enables implementing scaled environments. A small physical environment can represent a large virtual one by using scaled coordinate transformation. Thus, smaller robots and motion capture systems can be used, reducing the overall cost of the experiment.

Elimination of safety risks. With MR, safety issues typically associated with human-robot interaction can be resolved by separating them into distinct physical or virtual environments. In MR, physical robots can interact with virtual humans or physical humans in a different physical environment, eliminating concerns for human safety when errors occur. This can also be applied to experiments involving potentially dangerous interactions between robots. Because it allows interaction between multiple physical and virtual environments, MR creates a safer, lower-risk experiment space.

Simplification of debugging. The shared physical and virtual MR space reduces the gap between simulation and implementation. MR's virtual aspect allows for visualization of various states of the robots (e.g. field of view and planned path) so errors can be caught earlier. MR creates an enriched environment where all physical and virtual data interact directly in real-time; no further computation or calculation is necessary. This expedites and simplifies debugging.

Unconstrained additions to robots. A MR environment allows adding or changing virtual features of robots that may be too costly, time-consuming, or impossible in reality. For instance, one can add a virtual camera to a robot that is too small to carry one.

Scaling up swarms. Finally, MR simplifies experiments on robot swarms. Full interaction between the physical and virtual environments means most of the swarm can be simulated, as it may be sufficient for experiments to be performed on only a handful of real robots. This allows practical testing of swarm algorithms even if money or space constraints limit the number of physical robots.

V. DEMONSTRATIONS

In the following we present several representative examples of using Mixed Reality in robotics research. In our demonstrations we use the Crazyflie 2.0, an open-source nano quadcopter [1], weighing 27 g and measuring 92 mm motor to motor. Its software and hardware specification are available online[1]. The Crazyflie is controlled externally using a computer (with USB dongle) or a tablet/smartphone (Bluetooth); onboard sensors (9 axis IMU and barometer) are used to stabilize flight.

Small size and low price (180 USD) make the Crazyflie an attractive option for research on multi-robot coordination. However, it has several limitations. Payload is limited to 15 g, which restricts sensors that can be added. The size also causes the Crazyflie to be more sensitive to external forces such as wind. Onboard processing abilities are also limited compared to larger research grade quadcopters such as the AscTec Hummingbird. These limitations make the Crazyflie a good candidate for algorithm prototyping in Mixed Reality.

Each of the following subsections will describe a practical use-case of Mixed Reality along with technical details on how to reproduce similar results[2]. A video of the experiments is provided as supplemental material. To demonstrate the versatility of the approach, we use three different virtual environments: UNITY 3D, V-REP, and GAZEBO.

A. Human-Following UAVs

In this demonstration, we aim to develop a team of UAVs capable of following humans. Since human motion is complex and unpredictable, and UAVs are highly dynamic systems, the typical coarse-detail simulation is not expected to be very accurate. While pure simulation is a valuable tool as a proof of concept, new issues are often encountered when moving from simulation to reality. These issues may pose safety risks for humans sharing the environment with the robots. Mixed Reality can add an intermediate step between simulation and practice. Note that while this demonstration uses virtual humans, it can also be applied to humans acting in a separate motion capture space.

1) Technical Details: Typical robot simulators either do not support simulating humans (e.g. GAZEBO 2+), or use a very simplified model (e.g. V-REP). We use the game engine UNITY 3D (4.6.2 Free Edition) [11], combined with the Virtual Human Toolkit [12] to simulate humans. This toolkit includes SMARTBODY [13], a character animation platform able to accurately simulate humans. The Behavioral Markup Language (BML) can be used to describe complicated be-

---

[1] See `http://github.com/bitcraze`

[2] Source code developed for a swarm of Crazyflies with a motion capture system using ROS will be published at `http://github.com/whoenig/crazyflie_ros`

(a) MR Components (b) Physical World (c) Virtual World (V-REP)

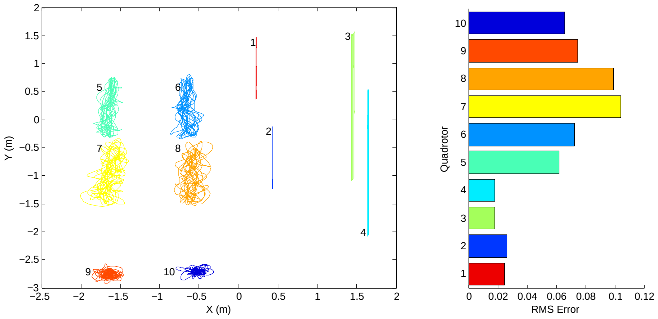
Fig. 3. A swarm of robots attempts to cover an area (outlined by the USC ACT letters). The six robots on the left are synchronized to a physical Crazyflie 2.0 swarm using a motion capture system. The four robots on the right are simulated in V-REP. All downward facing cameras are virtual.

haviors, such as locomotion (including steering), lip-syncing, gazing, and object manipulation. SmartBody transforms those high-level descriptions into animations in real-time.

The scene in Unity 3D contains two virtual humans and two quadcopters with forward-facing virtual cameras in an outdoor setting. Humans are controlled by SmartBody to follow a predefined path while obeying natural movement rules. To include more accurate quadcopter dynamics, we use two Crazyflies which fly in a space equipped with a 12-camera Vicon MX motion capture system [14]. The robots' positions are tracked at $100\,\mathrm{Hz}$, and a Unity 3D script updates the position of the virtual quadcopters using the VRPN protocol [15]. To demonstrate the versatility of the approach, the virtual space is two times larger than the physical space ($5\,\mathrm{m} \times 6\,\mathrm{m}$), allowing the virtual humans to walk farther. A simple controller takes the known positions of the quadcopter and the human to compute a goal position which keeps the human in the field of view. The Mixed Reality schema, virtual environment, and setup in the physical world are shown in Fig. 2.

2) Discussion: When humans and robots share environments, reducing the gap between simulation and practice can be crucial to ensure safety. For instance, the AscTec Hummingbird quadcopter can reach speeds up to $15\,\mathrm{m/s}$. A crash at such high speeds can cause severe accidents, especially if the UAV operates close to humans as in the discussed surveillance application. Even if an algorithm works well in simulation, many added uncertainties in the physical world could put a human participant at risk. The MR approach reduces those risks by adding intermediate steps between simulation and realization. Additionally, MR allows the limited space of indoor motion capture systems to be overcome by using a smaller robotics platform with added virtual sensors and by introducing scaling between the virtual and physical environments. For example, when outdoor UAV flight is restricted, outdoor components such as rocks, trees, or buildings can be fused with scaled indoor motion capture data in a virtual environment. Thus, MR enables refining an algorithm in a safe and less restricted environment.

Similarly, it is possible to use actual humans in a separate physical environment rather than using virtual humans, combining the physical environments in a virtual one using a simulator. Such a multi-physical world approach provides safety and better approximation to reality compared to virtual simulations. This method can be used for different robots as well, improving collaboration between researchers. Furthermore, the possibility of scaling between different physical environments allows interaction between entities of different sizes.



(a) Trajectory plot (b) RMS Error

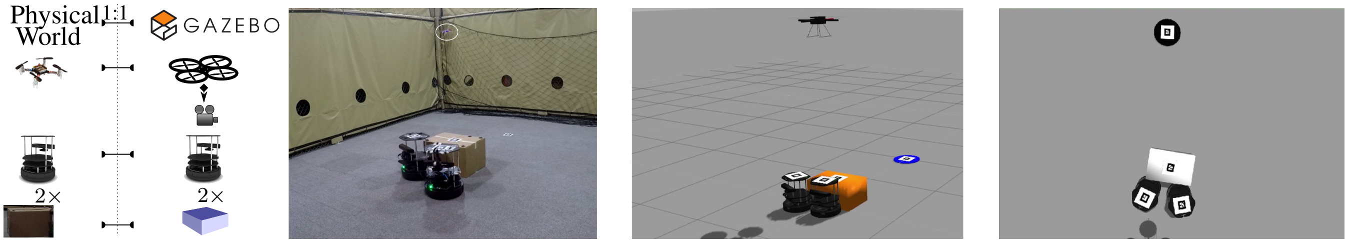
Fig. 4. Tracking performance of physical vs. virtual quadcopters. The six trajectories on the left in (a) were created by Crazyflies in the physical world, while the four on the right are simulation results. The root mean square (RMS) error between planned and actual position is much smaller for simulated quadcopters (b).

B. Area Coverage using UAVs

For this demonstration, we are interested in testing an algorithm for a large swarm of robots when enough robots are not available or space does not allow so many to be used. We use area coverage with UAVs as the task with a simple centralized approach, where the goal position of each quadrotor is directly computed from the known position of the target and a fixed translation. This demonstration also conveys that the behavior of a physical robot swarm differs substantially from that of a simulated swarm.

1) Technical Details: We fly six Crazyflies in our motion capture arena; their pose is sent to V-REP (3.2.0) [16] over ROS. V-REP simulates four additional quadcopters using the included `Quadricopter` model and also simulates the desired area to cover. The area changes over time, thus the updated target pose for each quadcopter is computed by a child-script in V-REP based on the current area configuration. The target pose is visualized by red (physical robots) and green (simulated robots) spheres for each quadcopter in the virtual environment. The motion capture space is mapped directly to the virtual environment (no scaling); the overall virtual space, however, is larger to accommodate the additional quadcopters. Screenshots from the demonstration and the MR schema are shown in Fig. 3.

2) Discussion: This approach showcases splitting a swarm of robots into virtual and physical components, which allows side-by-side comparison of the accuracy of simulation with real implementation.



(a) MR Components (b) Physical World (c) Virtual World (Gazebo) (d) Virtual Camera

Fig. 5. Two TurtleBots collaboratively move a box using the guidance of a physical UAV with a virtual downward facing camera simulated with Gazebo (a). The pose of the robots and the box are estimated using AR tag detection.

For the given example, the physical quadcopters do not follow the target trajectory as smoothly as the simulated ones shown in Fig. 4, due to the added uncertainties of a real-world implementation and external forces such as air currents. Furthermore, connectivity issues and as a consequence increased latency in the control algorithm cause additional instabilities which are not part of a simulation. As a result, the physical quadcopters are unable to monitor their areas with the same accuracy as the virtual quadcopters.

Since quadcopters are highly dynamic systems which are difficult to accurately simulate, especially as a team, this approach allows for motion comparisons between the quadcopters in the physical and virtual environments. Data collected from such an experiment can be used to design a model for a simulated quadcopter that is specific to the unique test environment the quadcopter will face. For instance, in order to simulate a swarm of UAVs under strong wind conditions, a wind model would be required. Tuning such parameters in an MR setting is simpler as the difference between model and reality is clearly visible. Once the simulation matches the physical robots, a larger swarm can be simulated by using both the physical robots and virtual robots using the newly tuned model, which can save money and time. This also allows for swarm algorithms to be tested in a simulation that more closely depicts the actual physical motions of a robot, prior to implementing the algorithms on their real counterparts. A similar approach could be used to identify limits of the simulation, for example by analyzing the behavior of UAVs flying in close proximity to each other.

C. *Object Moving with Limited Localization*

In this demonstration, two robots without onboard vision must push a large box with the help of the overhead quadcopter. In environments that are GPS-denied and not equipped with motion capture infrastructure, a camera-equipped UAV can be used to identify and track robots using special markers (so-called AR tags) and vision processing (see [17]). Based on the locations of the robots and the box, which are computed with vision processing from the overhead camera view, the robots drive toward the box to move it. While this is a simple scenario, it demonstrates the capabilities of the MR approach for testing vision-based approaches without having the required set of hardware. It also shows an example of a step-by-step transition from pure simulation to MR.

1) Technical Details: We implemented the scenario in simulation using Gazebo 2.2.3 [18] and ROS Indigo by incorporating `turtlebot`, `hector_quadrotor`, and `ar_track_alvar` packages. After successful simulation, we began Mixed Reality testing in stages.

In the first stage, we replace the simulated quadcopter by a Crazyflie in the physical environment, while keeping the camera simulated by Gazebo. Note that the 6 degree-of-freedom pose of the quadcopter captured by a motion-capture system is used in positioning the virtual camera and thus the camera angles are accurate. This allows analyzing the behavior of the algorithm while including more realistic dynamics and external influences such as wind.

In the second stage, we replace the virtual TurtleBots with physical ones (TurtleBot 2) while leaving the box in the virtual environment. The TurtleBots are localized using the external motion capture, however the control algorithm is based solely on the virtual camera image. This introduces new uncertainties caused by the differences in the PID controller used, uneven floor, and other effects which are not considered in simulation. Similar to the previous demonstration, it is possible to have one physical and one simulated TurtleBot to compare the different behaviors side-by-side.

Finally, we introduce a physical box to include the additional forces created by the box movement. Screenshots and MR schema of this particular experiment are shown in Fig. 5.

2) Discussion: This experiment shows how Mixed Reality can help reduce the gap between simulation and implementation. The approach can be used to break the implementation into more controllable stages, where it may be easier to analyze and isolate failures. For example, a staged approach can be used to debug a particular issue by moving objects into a reproducible virtual environment and thus isolate a component that causes a failure in the physical environment.

In software engineering, such virtualized objects, known as mock objects, play a crucial role in development and testing [19]. In particular, moving physical objects to the virtual environment solves the same problems as mock objects do in software engineering, such as introducing deterministic behavior, improving execution time, or simplifying testing to isolate bugs.

VI. DISCUSSION

Through discussion of its novel advantages and three compelling use-cases for the approach, we have shown that Mixed Reality can provide many advantages for research and development in robotics. Furthermore, as one specific example of MR, we provide a testbed on how to use small UAVs such as the Bitcraze Crazyflie 2.0 instead of bigger and more expensive quadcopters, with practical tips for how to use different robotics simulation tools with Mixed Reality.

Our demonstrations show that it is possible to use Mixed Reality with typical, well-known robotics simulators as well as less familiar ones. More traditional robotics simulators such as GAZEBO or V-REP can be used to first simulate algorithms and then, using the same platform, test intermediate steps using Mixed Reality. UNITY 3D, combined with SMARTBODY, can be used to accurately simulate humans, creating a safer intermediate step for testing algorithms for robots in environments shared with humans without the typical risks. Furthermore, the Mixed Reality approach allows for early discovery, isolation, and correction of potential implementation issues, such as wireless communication dropouts. In addition, Mixed Reality enables experimenting with more robots than available or physically capable, not only as pertaining to virtual swarms as we have demonstrated, but also with physical robots in different locations. This expands the possibility of collaboration between different labs with different robotics hardware. Finally, Mixed Reality enables relying on cheaper, or in the case of robot swarms, fewer robots for initial experiments by simulating additional hardware, or sensors that cannot be easily installed on existing robot hardware.

Mixed Reality also permits a large amount of flexibility for testing environments. The choice of which objects are physical and which are virtual can be made by the user based on the experiment's needs. Additionally, it is easy to move physical objects to virtual ones and, given available hardware (e.g., the correct robots), vice versa. The different simulation platforms for virtual environments have different capabilities, which can be combined using multiple virtual environments.

While the Mixed Reality approach clearly has many advantages, it does have some limitations. First, the linking of virtual and physical objects requires localization. This can be a self-localization method such as SLAM, which requires in many cases a lot of sensors and onboard computation, or external localization, such as a motion capture system. Second, a partially simulated environment does not have the same properties of an entirely physical environment; for example, virtual cameras may not show distortion or be affected by lighting conditions, and physical robots that interact with a virtual objects do not receive force feedback from the object. Therefore, as in pure simulation, a successful Mixed Reality experiment does not guarantee a working system in the physical world. The goal, however, is to add intermediate steps between pure simulation and full implementation.

In our future work we plan to address issues related to scaling between virtual and physical environments. In our demonstrations we used nano quadcopters to simulate bigger ones. Although we showed that this is more realistic compared to pure simulation, the behavior of the bigger drone would be different. For example, reaction to wind is highly dependent on the size of the UAV, which is amplified if scaling up between virtual and physical environments: While the scale allows flight in a smaller space, it also amplifies the noise. Furthermore, certain control algorithms such as trajectory following might require non-trivial changes (e.g., velocity adjustments) to accommodate the change in scaling.

In summary, Mixed Reality testing is one step closer to accurate implementation of a robotic system in the physical world, and can be an extremely valuable tool for research and development in robotics, particularly for multi-robot systems or robots which share environments with humans.

REFERENCES

[1] Crazyflie nano quadcopter. [Online]. Available: http://www.bitcraze.se

[2] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE Trans. Information Systems*, vol. E77-D, no. 12, pp. 1321–1329, Dec. 1994.

[3] F. Ghiringhelli, J. Guzzi, G. A. D. Caro, V. Caglioti, L. M. Gambardella, and A. Giusti, "Interactive augmented reality for understanding and analyzing multi-robot systems," in *IEEE/RSJ Intl Conf. Intel. Robots and Systems*, Chicago, Sept 2014, pp. 1195–1201.

[4] F. Leutert, C. Herrmann, and K. Schilling, "A spatial augmented reality system for intuitive display of robotic data," in *ACM/IEEE Intl Conf Human-robot Interaction*, 2013, pp. 179–180.

[5] A. Garcia, G. Fernandez, B. Torres, and F. López-Peña, "Mixed reality educational environment for robotics," in *IEEE Intl Conf Virtual Environments, Human-Computer Interfaces and Measurement Systems*, Sept 2011.

[6] I. Y. Chen, B. A. MacDonald, and B. Wünsche, "Mixed reality simulation for mobile robots," in *IEEE Intl Conf Robotics and Automation*, Kobe, Japan, May 2009, pp. 232–237.

[7] E. Freund and J. Rossmann, "Projective virtual reality: bridging the gap between virtual reality and robotics," *IEEE Trans. Robotics and Automation*, vol. 15, no. 3, pp. 411–422, 1999.

[8] M. T. Bolas and S. S. Fisher, "Head-coupled remote stereoscopic camera system for telepresence applications," *Proc. SPIE*, vol. 1256, pp. 113–123, 1990.

[9] Z. Yang, K. Nahrstedt, Y. Cui, B. Yu, J. Liang, S. Jung, and R. Bajcsy, "TEEVE: the next generation architecture for tele-immersive environment," in *IEEE Intl Symposium on Multimedia*, Irvine, CA, Dec 2005, pp. 112–119.

[10] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *IEEE Comput. Graph. Appl.*, vol. 21, no. 6, pp. 34–47, Nov. 2001.

[11] Unity 3d. [Online]. Available: http://unity3d.com/

[12] A. Hartholt, D. Traum, S. C. Marsella, A. Shapiro, G. Stratou, A. Leuski, L.-P. Morency, and J. Gratch, "All together now: Introducing the virtual human toolkit," in *Intelligent Virtual Agents*, Edinburgh, UK, Aug. 2013.

[13] A. W. Feng, Y. Xu, and A. Shapiro, "An example-based motion synthesis technique for locomotion and object manipulation," in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ser. I3D '12. New York, NY, USA: ACM, 2012, pp. 95–102.

[14] Vicon motion systems ltd. [Online]. Available: http://www.vicon.com

[15] R. M. Taylor, II, T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helser, "VRPN: A device-independent, network-transparent VR peripheral system," in *Proc ACM Symposium on Virtual Reality Software and Technology*, Banff, Alberta, Canada, 2001, pp. 55–61.

[16] M. F. E. Rohmer, S. P. N. Singh, "V-REP: a versatile and scalable robot simulation framework," in *IEEE/RSJ Intl Conf Intel. Robots and Systems*, Tokyo, Nov 2013, pp. 1321–1326.

[17] K. Hausman, J. Müller, A. Hariharan, N. Ayanian, and G. Sukhatme, "Cooperative control for target tracking with onboard sensing," in *International Symposium of Robotics Research*, Morocco, Jun 2014.

[18] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, Sep 2004, pp. 2149–2154.

[19] D. Thomas and A. Hunt, "Mock objects," *IEEE Software*, vol. 19, no. 3, pp. 22–24, May 2002.