2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)

# Exploring Augmented Reality Interaction for Everyday Multipurpose Wearable Robots

Jaryd Urbani, Mohammed Al-Sada, and Tatsuo Nakajima
Department of Computer Science and Engineering
Waseda University, Tokyo, Japan
{jaryd, alsada, tatsuo}@dcl.cs.waseda.ac.jp

Thomas Höglund
School of Technology and Innovations
University of Vaasa
Vaasa, Finland
thomas.hoglund@univaasa.fi

***Abstract*—Multipurpose wearable robots are an emerging class of devices, which offer intriguing interaction potential. The interoperability of multipurpose wearable robots and other types of wearable devices remains largely uninvestigated. We take the first steps to bridge this gap by presenting a framework for integrating augmented reality (AR) and multipurpose wearable robots. Our framework uses the publisher-subscriber model to expose different robot functionalities as services on a network. These can be invoked by the AR system. This model is advantageous in coping with different robot morphologies and various interaction methods. We implemented a prototype system using our framework by integrating an AR head-mounted display (HMD) and a wrist-worn robot, and demonstrate four experiences utilizing our prototype solution: 1) Robot status display, 2) shape-changing menus, 3) a media player and 4) a robot pose controller. To evaluate our approach, we performed a user study, which gauged user impressions and usability of developed experiences. Results indicate that our approach was well received, though participants highlighted a number of challenges in AR tracking when interacting within some of the experiences. Lastly, we discuss limitations and future research direction for our project.**

***Keywords-Wearable Robotics, Augmented Reality, Human-Robot Interaction***

## I. INTRODUCTION

Wearable robotics has long been researched in various communities, where it can be applicable to a wide variety of applications, ranging from haptic feedback to an exoskeleton for empowering a user's limbs. However, interaction with these devices is an ongoing challenge as most of them lack efficient input and feedback methods, or are comprised solely on interaction methods for a specific usage context [3], [4], [5].

Modern advancements in augmented reality (AR) have made it an easily accessible technology and a popular platform for a variety of applications. In addition, modern platforms used for AR, such as smartphones or head-mounted displays (HMDs), include several sensors and inputs, which enrich the entire AR experience.

The integration of AR and wearable robotics presents a highly intriguing potential interaction medium. Yet, cross-device interaction involving robotic wearables is largely unexplored, especially interaction that involves multiple wearable devices. Therefore, we present a framework for efficient development of integrated AR and robot experiences. Our integration framework offers the flexibility of both AR and robot control systems via a publisher-subscriber service model made available over the network. To verify and demonstrate our framework, we designed a multipurpose wrist-worn robot and an AR application, both integrated within our framework. We then developed a few experiences, which include AR robot pose control, robot status display, AR menu navigation with robot shape-change, and a robotic haptic interface for an AR media player. We conducted a user study to validate our experiences. Results indicate that most participants preferred the media player experience, and liked the robot status display least. We discuss the current limitations of our work and provide a possible direction for future work.

The contributions of this work include the following: 1) A flexible framework for developing applications involving wearable robots and AR. 2) A number of experiences to demonstrate the capabilities of our framework. 3) Results of a user evaluation of our developed experiences.

Figure 1. User operates the AR-integrated wearable robot.

## II. RELATED WORK

### A. Augmented Reality

Augmented reality allows for digital information, such as text, graphics, or 3-D models, to be rendered within real world contexts. As AR is essentially an output method, AR is supplemented with various interaction modalities [11], [12], such as voice commands or hand gestures. Researchers have investigated using AR for maintenance [7], [8], where visual and auditory instructions can easily be relayed to users in the field. Reality Editor [10] and Smarter Objects [9], show how AR can serve as an effective medium for controlling smart environments. For example, visualization of mappings among smart objects and making alterations to suit their control requirements. Their work

2325-1301/18/$31.00 ©2018 IEEE
DOI 10.1109/RTCSA.2018.00033

also showed how users can control smart objects with AR using a variety of interaction methods.

### B. *Augmented Reality in Robotics*

Previous work in this domain investigated the use of AR to compliment interaction with robots. Several works utilized AR for displaying *robot intention*, whereby the expected motion trajectory or future state of a robot is represented digitally before execution [23], [24], [25], and for *robot control*, where robots can be operated or teleoperated in real-time [10], [26], [27], and for *environmental tracking*, in which AR monitors robots and their environments to facilitate movement and interaction with their surroundings [1], [31], [32]. Some research into digital augmentation of robot form provided robots with previously unavailable functionality such as robot gestures or deixis [28], [29], [30]. Anthropomorphic aesthetics such as faces or hands can be added to robots to allow for more human-like interaction.

We differ from previous research by focusing on the previously uninvestigated integration of personal use wearable robots and AR. Another challenge our framework addresses is the integration of various types of interaction modalities of AR with multipurpose shape-shifting robots.

### C. *Multipurpose Robots and Actuated Wearables*

Some wearables utilize actuation for a variety of purposes. Leigh and Maes [3] demonstrated a wrist-worn shape-shifting robot capable of serving as a supernumerary robotic limb to hold objects, or as a haptic input device for interacting with a PC. The robot is controlled by a 'Myo' electromyography arm band monitoring hand gestures [6]. They also presented a wrist-worn robot [2] with interchangeable modules. By juxtaposing different modules, the hardware can deliver different experiences, such as notifications, haptic or shape-changing capabilities, or PC controls through a knob type input module. Rovables [4], [5] are swarm type wearable robots that offer a variety of interactions. These robots can reposition themselves, which allows them to deliver haptic feedback in different locations on the user's body. When equipped with LEDs, they can deliver visual notifications or serve as fashion displays.

Within the surveyed works, cross-contextual control and feedback remains a challenge. Demonstrated scenarios function on a singular interaction method and lack the ability to switch between different applications. For example, switching from functioning as a haptic feedback device to a device for self-expression. In contrast to previous works, our framework enables investigation of cross-contextual AR based interaction with multipurpose wearable robots, which we believe was largely unexplored.

## III. Approach and Implementation

### A. *Design*

The objective of our framework is to enable proficient development of integrated AR and wearable robot experiences for coherent user interaction. Delivering such a unified experience requires integrating control and feedback methods from both the AR HMD and the robot. The use of AR, and related interaction modalities, enables the robot to function independently of the user's physical limbs, and allows for the entire system to be used in a mobile context. We define our approach for each component, including hardware, software and integration implementations below.

### B. *Multipurpose Wearable Robot*

Software: Our robot control software was developed in C# and utilizes the Robotis SDK [19]. This software abstracts all robot communication and control functionality, and provides a unified servo class interface through which servos can be easily accessed. We designed a graphical user interface (GUI), which allows us to interact with the abstracted information (Fig. 2). Through this GUI we are able to monitor and control the robot and each individual servomotor. All servomotor attributes and movement controls are made available over the network.

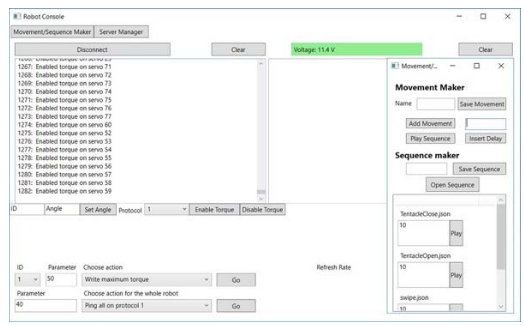


Figure 2. The control software GUI. Motors can be controlled manually within the interface. The smaller window to the right, allows movements to be captured and saved. These movements can be executed singularly or as sequences.

Hardware: Our robot consists of six interlinked Robotis Dynamixel AX-12A servomotors (stall torque 1.5 Nm) [19]. The servomotors are fastened together using plastic brackets. We have chosen this robot configuration to maximize the number of useful robot poses, making it flexible enough to be used in a variety of applications. The robot is mounted on the user's wrist using a plastic bracelet and a Velcro strap to provide a stable base. The total weight of the robot is 500g and the length at full reach is 300 mm.

Power: The robot is powered by a lithium polymer battery, which allows for approximately 25 minutes of robot operation.

Control Unit: Our robot control software is deployed on a GPD WIN portable mini-computer [18] (Fig. 3), which connects to the robot via USB cable interface.

### C. *Augmented Reality and Head-Mount Display*

Software: We developed an AR application using Unity3D [16] and Vuforia [17]. The user interface is controlled with gaze, allowing the user to simply look at objects to interact. Here, gaze is configured for the user to focus on an object for two seconds in order to perform an operation. For spatial registration, we utilize fiducial markers (Fig. 3). Fiducial markers are easily detectable images recognized by the AR device and provide a fixed point of reference, including scale and position, upon which AR content can be placed in the real world. They are easy to use, robust, and programs for fiducial marker tracking are common and easily deployable on most modern devices.

In the head-up display (HUD) mode, our application also displays AR content but without spatial registration. Here, pieces of AR content are fixed within the user's field of view, as shown in Fig. 5 and Fig. 6.

By providing both fiducial markers and a HUD, our system can accommodate a variety of interaction experiences. These modes make the system context adaptive in situations where users are not able to maintain constant visual contact with fiducial markers.


Figure 3. Fundamental components (a) AR enclosure, (b) GPD WIN, (c) wearable robot with attached fiducial markers, and (d) Android smartphone.

Hardware: The HMD is a generic 3-D AR/VR smartphone enclosure fitted with a Google Nexus 6P Android Smartphone (Fig. 3) running our AR application.

## D. Integration Framework

Our integration framework relies on WebSockets [14] to implement a publish-subscribe messaging pattern [13] between the robot and AR system. Systems can subscribe to services based on required functionalities. In our system, the robot controller computer is the publisher, and the AR system is the subscriber. Two types of WebSocket services are exposed, these enable the AR system to issue commands for the robot and receive feedback about the robot. We categorize the services as follows:

1) Control services, which allow clients to invoke servomotor controls, such as turning to specific angles, altering speeds, or setting maximum torque limits. Movements and movement sequences are also invoked via control service.

2) Feedback services, which broadcast robot information at specific rates and on events. For instance, applications can receive regular updates of servomotor attributes, such as angles, temperatures, maximum torque limits, and loads.

Messages exchanged with services are created using JavaScript object notation (JSON) [15]. We chose this notation as it is an easy and robust method of encapsulating data. Each control message comprises message type, intended command, servomotor IDs and other parameters. Similarly, each feedback message comprises feedback type and related data. For example, to rotate servomotor 11 to angle 90 degrees, at a speed of 50%, we send:

*{"Movement":[{"ID":"11","protocolVersion":"1", "ServoType":"AX 12A","angle":"90","speed":"50"}]}*

Message structures can easily be defined by developers to match specific application needs. For example, messages can also be created to address multiple servomotors, to request specific feedback, or to broadcast commands to the robot.

The publish-subscribe messaging pattern provides numerous advantages. Several services could be created for various control or feedback requirements, such as controlling specific servomotors or reading temperature

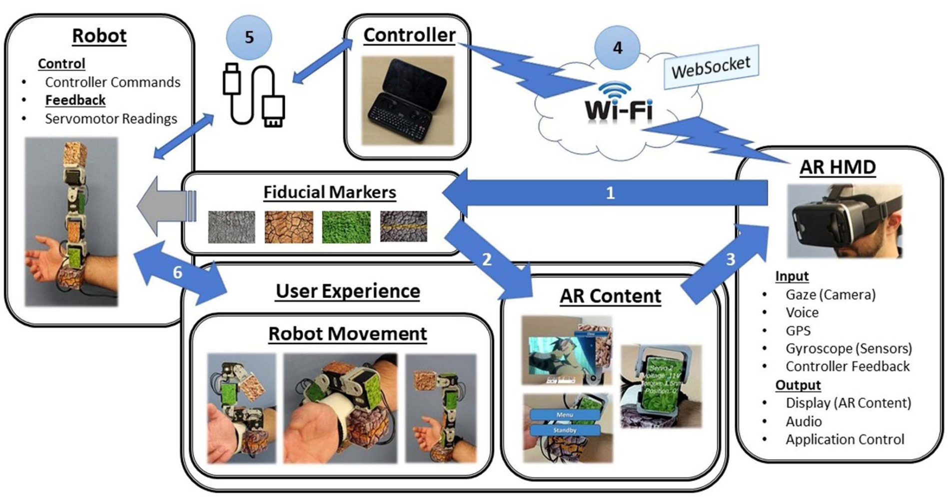


Figure 4. This diagram illustrates an overview of our framework. The workflow runs in numeric order from (1) to (6) or reverse numeric order from (6) to (3). Example, 1) AR HMD tracks the fiducial markers fixed to the robot, 2) Relevant AR content is displayed on the markers, 3) gaze interaction with the AR content is performed, 4) Application instructions are sent to the controller service via WiFi, 5) Service messages are translated to robot movements, 6) Robot movements are performed.

feedback. Such segregation can provide scalability and reliability, enabling different services to run as separate processes or threads [13]. Moreover, this structure can accommodate changes to the selected hardware, such as changing the robot morphology or AR HMD. The segregation of manufacturer specific robot controls by means of invoked services allows the robot control software to be altered or upgraded without affecting the network interface. Accordingly, this enables developers to experiment with different AR interaction and spatial registration methods without affecting the robot software. This model also allows multiple devices to subscribe and interface with multiple robots, which enables a flexible platform for experimenting with various device or robot configurations.

Our framework addresses the challenges of integrating AR and wearable robots, especially for creating interactive user experiences. Previous work focused on demonstrating potential AR use cases, without emphasizing the underlying implementation infrastructures [1], [9], [27], [30], [31].

Other similar frameworks focused on different application domains and robot types, such as industrial robots and applications [33].

ROS [35] is middleware for robotics and is mainly concerned with the integration and control of different robotic components, but differs in scope from our framework. Our framework targets the creation of experiences for AR and daily worn robots. For example, our framework can accommodate different interaction modalities, AR tracking methods, and robot morphologies. These aspects drive research within relevant fields, such as human robot interaction and user experience design.

## IV. Applications

We developed a few applications to demonstrate and evaluate our proposed framework. A feedback service and controller service, labeled "/feedback" and "/control" respectively, were created and exposed for the following AR applications to connect to.

### A. Robot Status Display

AR can provide visualization and spatial registration capabilities for different kinds of robot information. We developed an application that displays real-time robot information, including servomotor angles, applied torque and voltage. Such information is displayed in two modes:

1) The information is spatially registered to each corresponding robot servomotor (Fig. 5). This method allows the user to inspect the status of each servomotor independently. Each servomotor has a unique fiducial marker which, when brought into focus, displays the corresponding information of the servo.

2) Information is rendered in a head-up display (HUD) and includes all available status readings for every servomotor. Information is positioned, without spatial registration, discretely within the user's view (Fig. 5). This is especially useful when the user cannot maintain visual contact with a specific servomotor, such as when it is wrapped around their wrist. The HUD mode is triggered immediately upon losing track of fiducial markers and enables the user to seamlessly continue monitoring the robot's information.


Figure 5. Examples of robot status display. Readings are shown for (a) servomotor 1 and (b) servomotor 2 each spatially registered to their respective fiducial marker, and (c) all servomotor readings are displayed within the HUD.

Implementation: Upon launching the "Status" application from the AR menu, the AR system connects to the feedback service through WebSockets. Servomotor information is provided at a rate of 2 Hz. With every message received, the values within the information on display is updated.

### B. Media Player

Wearable robots can provide numerous haptic interaction capabilities, especially when combined with AR. Thus, we developed a media player application to demonstrate how digital applications may be embedded within our framework and how we interact with them in this new context. Similar to the robot status application, the media player may be displayed in two modes:

1) Content is displayed with spatial registration to the endmost servomotor as shown in Fig. 6. In this mode, the user can interact with AR buttons and view additional information such as the progress bar.

2) Within a HUD (Fig. 6), without access to AR control buttons. The HUD mode is triggered immediately upon losing track of the fiducial marker, and vice versa.


Figure 6. Examples of Media Player display types, where (a) shows spatially registered content and the corresponding AR buttons and progress bar only available in this view, and (b) shows HUD style content where the media is fixed to the top right of the user's view, for a less obscured real-world view.

Implementation: The media library is accessed directly from the SD Card of the smartphone. A haptic controller was developed which, upon launching the “Media Player” application from the AR menu, connects to the exposed feedback service through WebSockets. This allows the user to control the AR media by manually rotating a servomotor. A message with the corresponding command is sent each time a designated servomotor angle is reached.

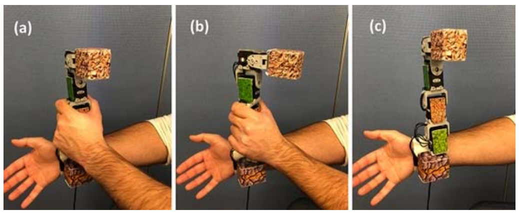

Figure 7. Media Player haptic “next” command (a) the user rotates the robot at servomotor 1, (b) at 40 degrees the feedback service broadcasts the instruction, the media library is iterated and (c) the robot returns to the original pose.

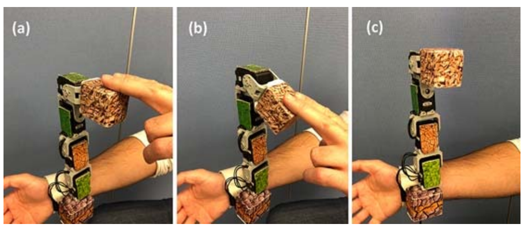

Figure 8. Media Player haptic “play/pause” command (a) the user depresses servomotor 5, (b) at 10 degrees the feedback service broadcasts instruction, media either holds or resumes and (c) the robot returns to the original pose.

### C. Shape-Changing Menus

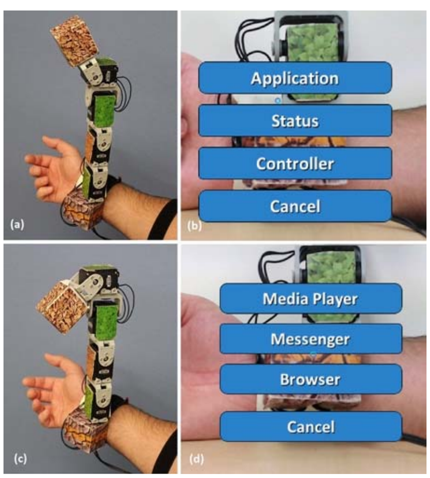

Figure 9. Examples of Shape-Changing Menu Representations where (a) and (c) are the physical representations of menus (b) and (d).

Robot movements and shape attributes can be utilized to convey information or supplement AR experiences. We demonstrate this concept through an application that manipulates the robot pose according to which AR system menu or embedded AR application is currently running. This is a physical representation of the AR menu and enables the user to recognize the AR state by observing the robot’s pose. This is useful if tracking is lost or if the display of AR content becomes obscured.

Implementation: The AR system connects to the controller service through WebSockets. Upon making an AR menu selection, a message is sent to the control service to trigger a preprogrammed robot pose, which matches that menu selection.

### D. Robot Pose Controller

User control methods of robots is an ongoing challenge [34], especially with regard to wearable robots [2], [3]. Users need to convey intended movements or poses to the robot using a suitable interaction method. HMD based AR can provide a rich interaction medium to address mentioned challenges. We demonstrate an application that allows a user to trigger a desired robot pose from within the AR menus (Fig. 10). This allows the robot to accommodate different situations or contexts. In standby mode, for example, the robot is wrapped around the user’s wrist. This unobtrusive pose allows the user to perform other physical tasks unimpeded by the robot’s usual extended form.

Implementation: The AR system connects to the controller service through WebSockets. Upon selecting a pose in the AR menu, a message is sent to the control service to trigger the robot pose which matches the user’s selection.

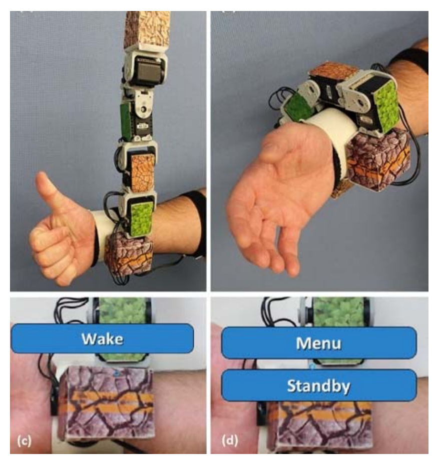

Figure 10. Robot pose control showing (a) operational and (b) standby modes, which are triggered by selecting the (c) “Wake” or (d) “Standby” menu items, respectively.

## V. Evaluation

### A. Obectives and Participants

We investigated user impression and usability of our developed applications and implementation platform. 7 students between the ages of 21 and 29 were recruited to participate in our study. All participants indicated that they were familiar with AR through common smartphone applications, while most knew of wearable robots through science fiction media or other research.

### B. Flow

The evaluation began with a brief introduction of AR and our robot. The participants were walked through the four applications, they then had the opportunity to experience each for themselves. After each application, users were asked to complete a short survey on that experience. Upon completion, the participants answered a questionnaire including a semi-structured interview (5-point Likert scale, 5 represents strongly agree). The user study lasted 60 minutes per participant and the configuration of our study is shown in Fig. 11.

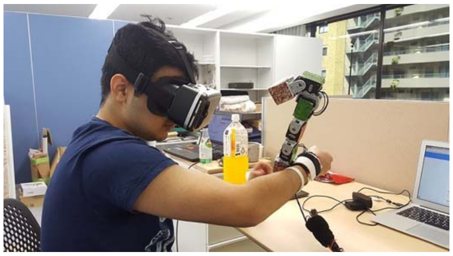

Figure 11. A participant trying shape-changing menus within our user study.

### C. Results and Analysis

Robot Status: Participants feedback was generally positive about this application ($m$=4.14, $sd$=1.06). They suggested including further information that could be useful to them, such as temperature readings or failure notifications. A few participants preferred using the HUD display due to the difficulty experienced when tracking between fiducial markers. We further discuss this shortcoming within the next section.

Shape-Changing Menus: Participants had mixed feedback for this application. Most participants were able to accurately correlate robot pose and menu state when the AR menus became hidden ($m$=3.14, $sd$=1.57). Participants indicated that the movement when changing robot pose was too subtle when navigating the AR menu. Participants recommend that movement should be more noticeable. We conclude that more distinct poses and faster pose transitions could contribute to such shortcomings.

Media Player: Participants found this application the most enjoyable, admiring the dual control methods of the content on display. They also approved of the HUD capable content for use in conjunction with other tasks ($m$=4.43, $sd$=0.53).

Robot Pose Controller: Participants found this application useful and the AR interaction easy to use ($m$=3.57, $sd$=0.53).

Robot and AR HMD: Participants stated that the robot and HMD were quite comfortable ($m$=3.0, $sd$=0.81) but mentioned that available tasks could benefit from a smaller form factor. AR navigation was easy and intuitive ($m$=4.14, $sd$=0.69), stating approval of “hands-free” control.

Overall, participants were satisfied with our approach ($m$=4.71, $sd$=0.49). Participants added that including further interaction modalities, like voice commands or eye-tracking, could provide intriguing multimodal interaction capabilities. We believe the results were generally encouraging to pursue further works.

## VI. Limitations and Future Work

The implemented applications have shown a number of shortcomings in our approach. First, despite its ease of use, our chosen hardware for AR has minor issues. The fiducial markers tracking varied with differences in lighting and camera focus. These limitations affect the interaction with the robot, by forcing the user to compensate by periodically moving into close proximity of the markers. Moreover, in addition to the obtrusiveness of the VR enclosure, prolonged use of the resource-intensive application resulted in poor smartphone performance, including severe battery drain and overheating, which challenges our envisioned usage scenario of an always worn and available wearable. Lastly, WiFi is vulnerable to communication delays, which may occur depending on network traffic and degrade the user experience.

To overcome tracking limitations, we intend to extend our evaluation to smart eyewear, such as HoloLens [20], which have optimized battery life, tracking-performance, connectivity and multiple interaction modalities. Advancements in tracking technologies, such as model tracking [21] or simultaneous localization and mapping [22] , are promising and could yield better tracking than our current implementation. We will also explore further applications for our robot. For example, different types of haptic feedback can be delivered, such as tapping or scratching the user with varied intensities and or directions. Lastly, we intend to use an ad-hoc wireless connection technology, such as Bluetooth, to overcome the limitations experienced with infrastructure WiFi.

We would also like to extend our framework to further accommodate robot enhancements, including intelligent motion planning systems and sensors. Mentioned enhancements would enable introducing advanced capabilities, such as collision avoidance with user’s hands or surroundings. Such advancements require improvements to the offered services within our future framework.

## VII. Conclusion

In the future, daily worn robots will play a larger role in our lives. These wearables have inherent control and feedback difficulties, especially as they lack embedded input methods that are suitable across a variety of contexts. Therefore, we bridge this gap with a framework that enables developing experiences that combine AR and wearable robots. After describing the details of our framework, we developed several applications using it. We proceeded with an evaluation of these applications. Results indicated that users liked haptic interaction using the robot, but shape-changing experiences require faster or larger movements to be noticeable during interaction. The results overall encourage us to pursue further work, specifically addressing technical challenges in tracking and networking, as well as exploring the multimodal interaction potential of our robot in different usage scenarios.

## ACKNOWLEDGMENT

The presented work was supported in part through the ABE Initiative by JICA and the Program for Leading Graduate Schools, “Graduate Program for Embodiment Informatics” by the Ministry of Education, Culture, Sports, Science and Technology. We acknowledge the financial support from KAUTE Foundation for the research reported in this paper.


## REFERENCES


[1] J. Frank, S. Krishnamoorthy, and V. Kapila, “Toward Mobile Mixed-Reality Interaction with Multi-Robot Systems,” IEEE Robotics and Automation Letters, vol. 2, no. 4, pp. 1901-1908, 2017.

[2] S. Leigh, T. Denton, K. Parekh, W. Peebles, M. Johnson, and P. Maes, “Morphology Extension Kit: A Modular Robotic Platform for Physically Reconfigurable Wearables,” Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction (TEI ’18), pp. 11-18, 2018.

[3] S. Leigh and P. Maes, “Body Integrated Programmable Joints Interface,” Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16), pp. 6053-6057, 2016.

[4] A. Dementyev, et al, “Rovables: Miniature On-Body Robots as Mobile Wearables,” Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16), pp. 111-120, 2016.

[5] H. Kao, et al, “Exploring Interactions and Perceptions of Kinetic Wearables,” Proceedings of the 2017 Conference on Designing Interactive Systems (DIS '17), pp. 391-396, 2017.

[6] Myo.com, “Myo Gesture Control Armband,” [Online] https://www.myo.com/ [Accessed: 15 April 2018].

[7] S. Henderson and S. Feiner, “Exploring the Benefits of Augmented Reality Documentation for Maintenance and Repair,” IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 10, pp. 1355-1368, 2011.

[8] B. Schwald and B. De Laval, “An augmented reality system for training and assistance to maintenance in the industrial context,” Journal of WSCG, vol. 11, no. 3, pp. 425-432, 2003.

[9] V. Heun, S. Kasahara, and P. Maes, “Smarter Objects: Using AR Technology to Program Physical Objects and their Interactions,”. CHI 2013 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13), pp. 961-966, 2013.

[10] V. Heun, J. Hobin, and P. Maes, “Reality Editor: Programming Smarter Objects,” Proceedings of the 2013 ACM conference on Pervasive and Ubiquitous Computing adjunct publication (UbiComp '13 Adjunct), pp. 307-310, 2013.

[11] M. Kolsch, R. Bane, T. Hollerer and M. Turk, “Multimodal interaction with a wearable augmented reality system,” IEEE Computer Graphics and Applications, vol. 26, no. 3, pp. 62-71, 2006.

[12] M. Lee, M. Billinghurst, W. Baek, R. Green, and W. Woo, “A usability study of multimodal input in an augmented reality environment,” Virtual Reality, vol. 17, no. 4, pp. 293-305, 2013.

[13] J. Kaiser and M. Mock, “Implementing the real-time publisher/subscriber model on the controller area network (CAN),” Proceedings of 1999 IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC '99), pp. 172-181. 1999.

[14] I. Fette, “The websocket protocol,” 2011, [Online] https://tools.ietf.org/html/rfc6455 [Accessed: 15 April 2018].

[15] JSON.org, “Introducing JSON,” [Online] https://www.json.org/ [Accessed: 15 April 2018].

[16] Unity3D.com, “Unity3D,” [Online] https://unity3d.com/ [Accessed: 28 March 2018].

[17] Vuforia.com, “Vuforia in Unity3D,” [Online] https://www.vuforia.com/ [Accessed: 15 April 2018].

[18] GDP.hk, “GPD WIN,” [Online] http://www.gpd.hk/gpdwin.asp [Accessed: 21 March 2018].

[19] Robotis.com, “ROBOTIS,” [Online] http://www.robotis.com/ [Accessed: 21 January 2018].

[20] Microsoft.com, “Detail of light reflection on lens of HoloLens,” [Online] https://www.microsoft.com/en-us/hololens [Accessed: 16 April 2018].

[21] Vuforia.com, “Model Targets – 2.0 API Overview,” [Online] https://library.vuforia.com/articles/Solution/model-targets-user-guide.html [Accessed: 16 April 2018].

[22] Wikitude.com, “SLAM (Simultaneous Localization & Mapping) Augmented Reality SDK That Enables Markerless and Marker-based AR Projects,” [Online] https://www.wikitude.com/wikitude-slam/ [Accessed: 15 April 2018].

[23] M. Walker, H. Hedayati, J. Lee, and D. Szafir, “Communicating Robot Motion Intent with Augmented Reality,” Proceedings of 2018 ACM/IEEE International Conference on Human-Robot Interaction (HRI ’18), pp. 316-324, 2018.

[24] E. Rosen, et al, “Communicating Robot Arm Motion Intent Through Mixed Reality Head-mounted Displays,” arXiv e-print, arXiv:1708.03655, 2017.

[25] R. Chadalavada, H. Andreasson, R. Krug, and A. Lilienthal, “That’s on my Mind! Robot to Human Intention Communication through on-board Projection on Shared Floor Space,” 2015 European Conference on Mobile Robots (ECMR), pp. 1-6, 2015.

[26] N. Tran, J. Rands, and T. Williams, “A Hands-Free Virtual-Reality Teleoperation Interface for Wizard-of-Oz Control,” Proceedings of the International Workshop on Virtual, Augmented, and Mixed Reality for HRI, 2018.

[27] S. Hashimoto, A. Ishida, M. Inami, and T. Igarashi, “TouchMe: An Augmented Reality Based Remote Robot Manipulation.” Proceedings of International Conference on Artificial Reality and Telexistence (ICAT 2011), 2011.

[28] T. Williams, N. Tran, J. Rands, and N. Dantam, “Augmented, Mixed, and Virtual Reality Enabling of Robot Deixis,” Proceedings of the International Conference on Virtual, Augmented, and Mixed Reality, 2018.

[29] T. Holz, M. Dragone, and G. O’Hare, “Where Robots and Virtual Agents Meet,” International Journal of Social Robotics, vol. 1, no. 1, pp. 83-93, 2009.

[30] T. Willaims, "A Framework for Robot-Generated Mixed-Reality Deixis," Proceedings of the International Workshop on Virtual, Augmented, and Mixed Reality for HRI, 2018.

[31] R. Grieder, J. Alonso-Mora, C. Bloechlinger, R. Siegwart, and P. Beardsley, "Multi-robot control and interaction with a hand-held tablet," In Workshop Proceedings of International Conference on Robotics and Automation (ICRA 2014), vol. 131, 2014.

[32] M. Fiala, "A Robot Control and Augmented Reality Interface for Multiple Robots," Proceedings of Canadian Conference on Computer and Robot Vision, pp. 31-36, 2009.

[33] N. Andersson, et al, "AR-Enhanced Human-Robot-Interaction - Methodologies, Algorithms, Tools". Procedia CIRP, vol. 44, pp. 193-198, 2016.

[34] M. Matteucci, "Publish/Subscribe Middleware for Robotics: Requirements and State of the Art," Technical Report, 2003.

[35] ROS.org, "ROS," [Online] http://www.ros.org/ [Accessed: 18 April 2018].