Kinetic AR: a Framework for Robotic Motion Systems in Spatial Computing

Anna Fuste
PTC Reality Lab
Boston, MA, USA
afuste@ptc.com

Ben Reynolds
PTC Reality Lab
Boston, MA, USA
breynolds@ptc.com

James Hobin
PTC Reality Lab
Boston, MA, USA
jhobin@ptc.com

Valentin Heun
PTC Reality Lab
Boston, MA, USA
vheun@ptc.com

CHI '20 Extended Abstracts, April 25–30, 2020, Honolulu, HI, USA.

ACM ISBN 978-1-4503-6819-3/20/04.
http://dx.doi.org/10.1145/3334480.3382814

Abstract
We present *Kinetic AR*, a holistic user experience framework for visual programming of robotic motion systems in Augmented Reality. The *Kinetic AR* framework facilitates human-robot collaboration in a co-located environment. Our goal is to present a deployable guide for the creation and visualization of manifold robotic interfaces while maintaining a low entry barrier to complex spatial hardware programming. A two phase validation process has been conducted to assess our work. As an initial phase, we have performed a set of interviews with robotics experts. Based on these interviews we have established three main areas that our framework tackles in different time domains. In a second phase, we have developed a set of prototypes using mobile Augmented Reality that apply the principles of *Kinetic AR* to multiple hardware actors including an AGV, a robotic arm, and a prosthetic system. Additional feedback from experts indicate the potential of the *Kinetic AR* framework.

Author Keywords
Augmented Reality; Spatial Computing; robotics; HCI; UX.

CCS Concepts
•**Human-centered computing** → **Mixed / augmented reality;** User interface toolkits; User interface design; Interface design prototyping; •**Software and its engineering** → *Integrated and visual development environments;*

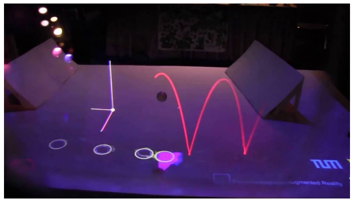

Figure 1: Laplacian Vision platform [14]

Introduction

Robotic motion systems have always been complex engineering challenges that require robotics experts. Experts need to train the ability to overcome a disconnect between a spatially moving robot and its abstract representation on a desktop computer. This problem becomes exponentially more difficult when attempting to synchronize multiple robots using 2D interfaces. Moreover, current robotics systems typically offer a limited interface to control the hardware motion, often by visualizing a digital twin on a teach pendant. In order to move the robot, the user has to compute the positions in the environment and input them through these limiting systems. Robotic motion is a spatial challenge and can be solved with a spatial solution; for this reason, Augmented Reality (AR) is a suitable technology to design usable systems to control and visualize motion in the environment. With Augmented Reality, we can now design improved user interfaces that lower the barrier of entry to interacting with these systems. To this end we present an Augmented Reality User Experience (UX) framework for visual programming of robotic motion systems.

Related work

Motion visualization of non-actuated objects

AR user interfaces have been explored in a vast range of research regarding motion representation and manipulation. Several projects explore motion tracking and visualization for non-actuated objects to assist in human tasks. Leigh et al. [17] present a system to visualize consequences of actions through AR. In Laplacian Vision [Figure 1] [14], Itoh et al. present a vision augmentation system which assists the human ability to predict future trajectory information.

Robotic motion in consumer level applications

Other fields of research have explored the concept of robotic motion tracking and interaction with actuated objects for simple generalized robotic and consumer level applications. In exTouch, Kasahara et al. [15] explore a robotic motion interface for actuated objects with a single control point using one and six degrees of freedom mediated by mobile AR. As an example, they show a movable furniture interface. The objects are actuated by the vector difference between a visual marker located on the actuated object and the phone tracking the marker. Using mobile AR with ground plane tracking, Fuste et al. perform path planning for a small robot in a playful interface in Invisible Highway [9].

Human-robot interaction for industrial spaces

While all this research tackles the challenge of motion control and representation of small actuated objects and consumer level robots, other research projects have focused on large scale co-bots and designing interfaces for industrial environments. Reardon et al. [20] use a Hololens head-mounted display (HMD) to present a visualization of the path and position of an Automated Guided Vehicle (AGV).

Several research projects have focused on conveying intent in a co-located environment by using projection or an LED interface. Chadalavada et al. [5] use projected AR in front of the AGV to convey motion intent. In [21], Sasai et al. present a robot that can guide people while projecting valuable information in its surroundings. Fernandez et al. [8] use LEDs along the chassis of an AGV to give people co-located feedback of the robot's immediate motion in the environment.

The above projects focus on visualization purposes. However, other research has tried to go further by allowing the user to control the AGV. In [19], Muhammad et al. build on top of Reardon's work [20] by presenting an AR framework that allows for the visualization of the AGV motion information as well as adjustment of its behavior by asking key questions to the user. Using an AR marker and a laser pro-

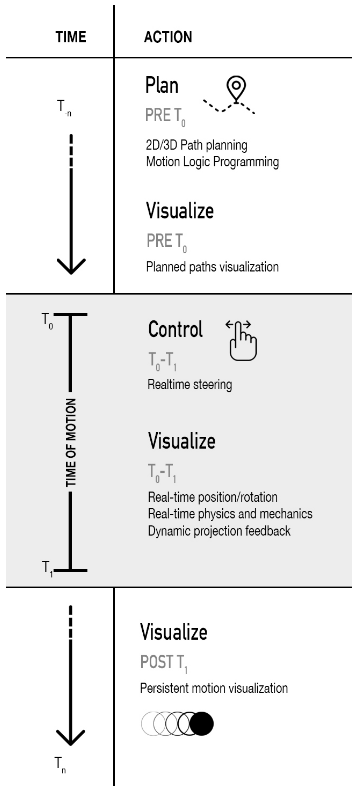

Figure 2: *Kinetic AR* motion framework time scopes and user actions.

jection, Huy et al. [13] present a framework to send real-time commands to an AGV using a custom made hand controller. Lakshantha et al. [16], on the other hand, have implemented their own SLAM algorithm and allow for a basic path planning interface for their AGV.

In the industrial arena, plenty of different automated robotic systems coexist. A body of research has focused on designing AR interfaces for robotic arms. Early in 1993, Milgram et al. [18] presented a collaborative system for a robotic arm using cameras for tracking and a monitor for visualization. Likewise, Akan et al. [1] or Hashimoto et al. [10] present similar interfaces displayed on a monitor, while Chong et al. [6] present an AR interface for a HMD to plan collision-free paths. A different approach is used in [24], steering the robot's arm in real-time with the user's gestures. Bischoff et al. [2] allow the visualization of workflows for a Kuka robotic arm to help inexperienced users. Cao et al. [3] present a visual authoring system for robot-IoT task planning. Fang et al. [7] present a system for planning the path and the orientation of the end-effector for an industrial robot. Additionally, a body of work has focused on AR interfaces for drones [11, 22, 23].

All the aforementioned explorations set the foundation for a field of research that will vastly advance the industrial space. The driving force for advancement in this field is a need for an untrained user to perform tasks otherwise reserved for well trained experts. Each of the presented related works tackle a different challenge separately. The majority of the work is directed towards visualizing and giving real-time feedback. All of the described interfaces are limited to one type of robotic system. A holistic framework applicable to all robotic motion in a modern factory is missing. Moreover, interface designs for the user to interact with the robots tend to be screen-oriented 2D representations and non-spatial, such as the interfaces to change behavior from Muhammad et al. [19].

Methodology and objectives

As seen in the related work, an extensive body of research has been dedicated to explain novel systems to control robotic motion using AR. Nevertheless, there is a lack of research in trying to present a holistic visual programming UX framework that tackles the different challenges of robotic motion and is inclusive of different robotic hardware technologies. Our aim is to define and present this UX framework. We have developed a foundation for the control, planning, and visualization of motion across different hardware systems. We have selected a sample of hardware systems that are representative of the different types of motion that can be found in manufacturing environments or environments that provide similar conditions. These sample hardware systems include: Automated Guided Vehicles (AGVs), robotic arms, collaborative robots as well as human robotic prostheses [4] and exoskeletons. This set of robotic systems was chosen to encompass a range of hardware capabilities; a variety of control points and applications in sufficiently complex settings to explore a meaningful range of interface challenges. The different scopes for these systems can be used as starting points for other types of robotic systems not included in this sample list.

Our line of research aims to solve challenges posed by current robotic systems. To ensure we considered all challenges that our framework had to address, we conducted a set of interviews with robotics experts that handle a multitude of applications as well as different perspectives. For example, we conducted these interviews with robotic systems integrators, manufacturers, and users. Their expertise ranged from academic and industrial research to day-to-day factory deployment for large and small scale manufac-

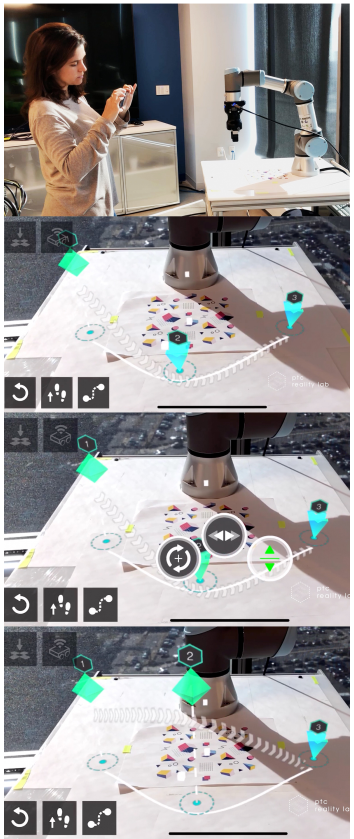

Figure 3: UR3E robotic arm interface. (a) User holding mobile device in front of a UR3E robotic arm. (b) Path planned with one elevated checkpoint and 2 surface checkpoints. User selects second checkpoint. (c) Checkpoint menu appears with motion parameters. (d) After selecting height parameter, user has made the second checkpoint elevated.

turing. Overall, we conducted interviews with 12 different industry leading experts. From these conversations, we established three major goals our research had to accomplish. First, provide intuitive and deployable path planning systems consistent across robots with different degrees of freedom. Second, improve flexibility in programming robotic systems in order to lower the need of acquiring more hardware for diverse tasks, reduce costs, and reduce training times. Finally, offer enhanced visualization tools for robotic motion intent in real-time and for past motion analysis.

Framework Description

To solve these challenges we have focused the *Kinetic AR* framework on a set of actions that the user can perform:

Path planning

Path planning is one of the most relevant applications when defining motion in a robotic system. Path planning is performed before the motion happens as seen in Figure 2. It can be performed by specifying locations in space for the robotic system to reach along a path. Using a mobile device, the user can move around in the environment and anchor a series of points on the ground or surface for a robotic system to follow. These points contain several parameters that can be tweaked, and a visual programming interface that allows the robotic motion to easily trigger or be triggered by external hardware or software systems. When the user taps on the screen, a checkpoint will appear at the location on the floor or surface that they are pointing at. This checkpoint consists of an inverted pyramid and a top label with the checkpoint index number in the overall path. The checkpoints can be directly dragged along the surface where the robot is in order to change their position. As the user creates more checkpoints, a path will be drawn between them. This path contains directionality with a set of arrows that point towards the direction of the motion.

Our motion interface has been designed to work either in a 2D motion setting (i.e. an AGV robot) or a 3D motion setting (i.e. a robotic arm or drone). The interface always allows for the creation of a checkpoint in a 2D plane corresponding to the floor or surface where the robot is. The user may decide to move this point in a third axis and create a 3D path as seen in Figure 3. To perform this action, the user [Figure 3a] can select the checkpoint [Figure 3b] and a menu will appear, with different motion parameters [Figure 3c]. In order to define a proper interface for path planning, we take into account the basic components of motion that can be tweaked in each checkpoint: landing position in the 2D plane, landing rotation, speed and height. By selecting the height parameter and dragging the checkpoint upwards [Figure 3d], the user gives a third dimension to this checkpoint, always relative to a physical surface in the environment. The checkpoint changes from being an inverted pyramid to being a 3D rhombus. The 3D rhombus signals the fact that the checkpoint is not in contact with the surface. The user sees a dashed line from the 2D surface to the checkpoint location. This dashed line allows the user to have a reference to the projection of the position in the physical environment. It allows for a more accurate localization of checkpoints.

If the user selects the rotation parameter, a representation of the rotation for that specific robot appears [Figure 4b]. The user is able to rotate the digital representation to set the desired rotation at that checkpoint. The user is able to adjust speed by dragging the finger up and down. The path gets wider (slower) or thinner (faster) to represent the speed at each point in the path.

The user can adjust the position of the checkpoint in two different ways. First, the position can be quickly set by dragging the checkpoint along the surface plane. After that, if

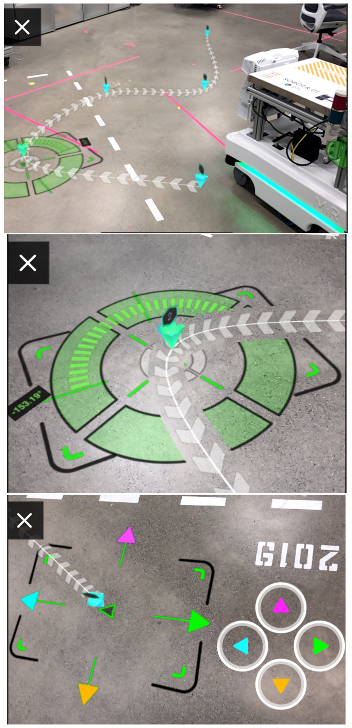

Figure 4: AR Interface for motion programming of an MIR100. (a) Several checkpoints form a path. One of the checkpoints has the rotation activated. (b) Rotation interface for the MIR100. The user can rotate the graphics to set the angle. (c) Position interface to adjust the accurate position of the robot at this checkpoint.

the user wishes to adjust the position in order to make it more accurate, the position option in the checkpoint menu shows a sub-menu with arrows [Figure 4c]. These colored arrows match the drawing on the checkpoint representation and allow the user to accurately adjust the position of that checkpoint using discrete values set with buttons.

Motion programming

Besides being able to tweak motion parameters for each one of the checkpoints, the checkpoints can be visually linked to any other piece of hardware or software in order to trigger actions. For example, an AGV could trigger a feeding station once a specified checkpoint is reached in front of the feeder and therefore goods can be moved to the AGV. By doing this, we allow the user to create their own motion logic and use it to orchestrate a bigger system or environment. For example, multiple robotic systems and AGVs can be orchestrated to collaborate. We use the Reality Editor platform [12] and have integrated the motion interfaces with it. As an example, we can program an AGV robot to follow a certain path. We can then link this defined motion to a robotic arm. When the AGV reaches a certain checkpoint, this will trigger a movement in the robotic arm, such as moving on a defined path to pick up a box the AGV is transporting. We express this entire workflow in two motion paths and one logic connection. The *Kinetic AR* framework simplifies a procedure that, as interviewed experts pointed out, normally takes hours of expert programming work into a couple minutes of simple interactions with clear affordances.

Real-time steering

Often, it is easier to control robotic systems by having them directly follow the motion of a human. Using *Kinetic AR*, we synchronize spatial environments for the visual tracking system and the robot coordinate system. This allows the possibility of having the robot follow a user as they move holding the AR device. For example, an AGV can follow a user as the user walks around, or a robotic arm can move in synchronization with a human arm.

Visualizing

This spatial synchronization between the robot and visual tracking system gives all the necessary information to the display device to provide any kind of related motion visualization. The motion can be visualized in real-time in mobile AR or in a projection in front of the robot [Figure 5]. The system allows for a continuous visualization of the past motion path performed by the robot as well as the current and future direction and speed values for the robot.

Another example is the visualization of robotic motion values for a prosthetic limb. After gaining insights from a prostheses expert we have designed the framework to accommodate for their challenges. He highlighted the fact that it is hard to adjust values for a prosthetic limb while a user is trying it. He follows the user with his laptop in his hands which becomes a rather cumbersome task. In contrast to his current workflow, if he were to use an interface designed with the *Kinetic AR* framework, the robotics expert could visualize values for the different components of the prosthetic device [Figure 6] and visualize the movement along a path. This also expands the possibilities to allow for control and adjustment of the prosthesis by its own user, making the whole interface more accessible to non-technical people.

Application Prototypes

We have designed and developed a set of prototypes to showcase the different scopes of the *Kinetic AR* framework using different hardware actors: an AGV, a robotic arm and a lower-extremity powered prosthesis. We use the Reality Editor [12] as our AR programming platform.

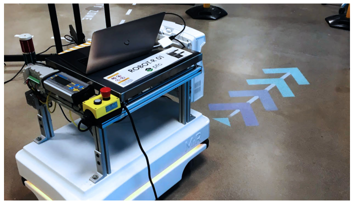

Figure 5: Co-located projection feedback. Additional users can visualize the intent of the AGV in real-time without the need of holding a device. This case requires the use of a projector.

Figure 6: Lower-extremity powered prosthesis AR interface. (a) Joint rotation and torque with variables graph for each element. (b) Robotic prosthesis components.

The Reality Editor is a platform built in JavaScript that allows for the creation of standard HTML applications that can be deployed to physical objects via AR. As such, this platform allows for spatial computing via web technologies. Anyone can create custom hardware interfaces that allow the communication with any other hardware.

AGV

The AGV robot is an MIR100 robot. It has two laser scanners that map the space and a REST API that allows for the programming of missions. By synchronizing the AGV mapping with the phone mapping we can program the robot to move to any position in the environment. Using the *Kinetic AR* framework, we have programmed the robot to perform different tasks. The robot can access the position of the phone at any time and follow the user.

Robotic Arm Collaborative Robot

The robotic arm we have prototyped with is a UR3E robot. The UR3E is accessible through a basic TCP connection via socket messaging. Using the same approach as the AGV, the user can utilize the *Kinetic AR* framework to plan and program motion for the robot. Despite having an additional dimension of motion compared to the AGV, the framework applies to the problems of a robotic arm.

Robotic Prostheses

We have also designed the interfaces that will be used for a lower-extremity powered prosthesis. Using the *Kinetic AR* framework, the user will be able to visualize joint position, angle or torque force. Other mechanical parameters are visible in AR such as information on the control unit, the motor encoder, the force sensor or the joint encoder. The interface will allow for the visualization of past motion in space as the person moves around. In this case, only a mock-up of the interface design has been developed as seen in Figure 6.

These prototypes were presented to the robotics experts in a validation session where they gave qualitative feedback. Although a quantitative evaluation is still needed for validation, all of the experts agreed that the technology and system had great potential for their industrial use cases.

Conclusions and Future Work

We have designed and implemented a visual programming User Experience framework for robotic motion in Augmented Reality called *Kinetic AR*. We conducted a set of interviews with experts in robotics that validated our initial directions and guided the continued development of our framework.

We have developed prototypes for three different hardware systems in order to validate the usability of the framework. Additional feedback from the aforementioned set of experts has demonstrated the potential of the platform in industrial environments. Because of this identified potential, we are moving forward with efforts to open source our framework and make it widely available to the research community.

As future work, we will improve the framework implementation and conduct a twofold evaluation. We will first test our application prototypes with a quantitative evaluation, comparing our framework to traditional programming interfaces such as teach pendants or browser interfaces. We will later proceed with a qualitative evaluation analyzing how different users design different interfaces using *Kinetic AR*.

Acknowledgements

This work has been possible thanks to the support of James Heppelmann, the PTC Office of the CTO, specially Steve Dertien and Stephen Prideaux-Ghee, and the PTC Academic program. We also want to thank Matt Carney for his support and work on lower-extremity powered prostheses.

REFERENCES

[1] B Akan, B Çürüklü, and others. 2010. Augmented reality meets industry: Interactive robot programming. In *Proceedings of SIGRAD 2010: Content aggregation and visualization; November 25–26; 2010; Västerås; Sweden*. Linköping University Electronic Press, 55–58.

[2] Rainer Bischoff and Arif Kazi. 2004. Perspectives on augmented reality based human-robot interaction with industrial robots. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, Vol. 4. IEEE, 3226–3231.

[3] Yuanzhi Cao, Zhuangying Xu, Fan Li, Wentao Zhong, Ke Huo, and Karthik Ramani. 2019. V. Ra: An In-Situ Visual Authoring System for Robot-IoT Task Planning with Augmented Reality. In *Proceedings of the 2019 on Designing Interactive Systems Conference*. 1059–1070.

[4] Matt Carney, Tony Shu, Roman Stolyarov, Jean-Francois Duval, and Hugh Herr. 2019. Design and Preliminary Results of a Reaction Force Series Elastic Actuator for Bionic Ankle Prostheses. engrXiv. DOI:http://dx.doi.org/10.31224/osf.io/3wt5j

[5] Ravi Teja Chadalavada, Henrik Andreasson, Robert Krug, and Achim J Lilienthal. 2015. That's on my mind! robot to human intention communication through on-board projection on shared floor space. In *2015 European Conference on Mobile Robots (ECMR)*. IEEE, 1–6.

[6] Jonathan Wun Shiung Chong, SKc Ong, Andrew YC Nee, and KB Youcef-Youmi. 2009. Robot programming using augmented reality: An interactive method for planning collision-free paths. *Robotics and Computer-Integrated Manufacturing* 25, 3 (2009), 689–701.

[7] HC Fang, SK Ong, and AYC Nee. 2012. Robot path and end-effector orientation planning using augmented reality. *Procedia CIRP* 3 (2012), 191–196.

[8] Rolando Fernandez, Nathan John, Sean Kirmani, Justin Hart, Jivko Sinapov, and Peter Stone. 2018. Passive Demonstrations of Light-Based Robot Signals for Improved Human Interpretability. In *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 234–239.

[9] Anna Fuste, Judith Amores, Jam3, and Google Creative Lab. 2017. Invisible Highway. (2017). https://experiments.withgoogle.com/invisible-highway

[10] Sunao Hashimoto, Akihiko Ishida, Masahiko Inami, and Takeo Igarashi. 2011. Touchme: An augmented reality based remote robot manipulation. In *The 21st International Conference on Artificial Reality and Telexistence, Proceedings of ICAT2011*, Vol. 2. Citeseer.

[11] Hooman Hedayati, Michael Walker, and Daniel Szafir. 2018. Improving collocated robot teleoperation with augmented reality. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 78–86.

[12] Valentin Heun, James Hobin, and Pattie Maes. 2013. Reality editor: programming smarter objects. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 307–310.

[13] Dinh Quang Huy, I Vietcheslav, and Gerald Seet Gim Lee. 2017. See-through and spatial augmented reality-a novel framework for human-robot interaction. In *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*. IEEE, 719–726.

[14] Yuta Itoh, Jason Orlosky, Kiyoshi Kiyokawa, and Gudrun Klinker. 2016. Laplacian vision: Augmenting motion prediction via optical see-through head-mounted displays. In *Proceedings of the 7th Augmented Human International Conference 2016*. ACM, 16.

[15] Shunichi Kasahara, Ryuma Niiyama, Valentin Heun, and Hiroshi Ishii. 2013. exTouch: spatially-aware embodied manipulation of actuated objects mediated by augmented reality. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*. ACM, 223–228.

[16] Eranda Lakshantha and Simon Egerton. 2014. Human Robot Interaction and Control: Translating Diagrams into an Intuitive Augmented Reality Approach. In *2014 International Conference on Intelligent Environments*. IEEE, 111–116.

[17] Sang-won Leigh and Pattie Maes. 2015. Aftermath: Visualizing consequences of actions through augmented reality. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 941–946.

[18] Paul Milgram, Shumin Zhai, David Drascic, and Julius Grodski. 1993. Applications of augmented reality for human-robot communication. In *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, Vol. 3. IEEE, 1467–1472.

[19] Faizan Muhammad, Amel Hassan, Andre Cleaver, and Jivko Sinapov. 2019. Creating a Shared Reality with Robots. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 614–615.

[20] Christopher Reardon, Kevin Lee, and Jonathan Fink. 2018. Come see this! augmented reality to enable human-robot cooperative search. In *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 1–7.

[21] Takuya Sasai, Yo Takahashi, Mitsunori Kotani, and Akio Nakamura. 2011. Development of a guide robot interacting with the user using information projection—basic system. In *2011 IEEE International Conference on Mechatronics and Automation*. IEEE, 1297–1302.

[22] Daniel Szafir, Bilge Mutlu, and Terrence Fong. 2015. Communicating directionality in flying robots. In *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 19–26.

[23] Michael Walker, Hooman Hedayati, Jennifer Lee, and Daniel Szafir. 2018. Communicating robot motion intent with augmented reality. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 316–324.

[24] Tom Williams, Nhan Tran, Josh Rands, and Neil T Dantam. 2018. Augmented, mixed, and virtual reality enabling of robot deixis. In *International Conference on Virtual, Augmented and Mixed Reality*. Springer, 257–275.