EXPLORING NOVEL HUMAN SMART-THING INTERACTION THROUGH AUGMENT REALITY FRAMEWORK DESIGN

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Yuanzhi Cao

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

Dec 2020

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL

STATEMENT OF DISSERTATION APPROVAL

Dr. Karthik Ramani, Chair

School of Mechanical Engineering

Dr. David J. Cappelleri

School of Mechanical Engineering

Dr. Alexander J. Quinn

School of Electrical and Computer Engineering

Dr. Shreyas Sen

School of Electrical and Computer Engineering

Approved by:

Dr. Nicole L. Key

Head of the School Graduate Program

For my *mother*

ACKNOWLEDGMENTS

I would like to first express my gratitude to my advisor Professor Karthik Ramani for constantly inspiring and challenging me to identify the right research question that is worth exploring for the direction of our lab and also suitable for my own interest. Professor Ramani has always demonstrated full support when my personal life has encountered crisis events, and for that, I am forever grateful. Further, I would like to extend my gratitude to Professor David Cappelleri, Professor Alex Quinn, and Professor Shreyas Sen for serving as my doctoral committee members and the inspiring discussions on my research.

I would like to thank all the members of C Design Lab, past and present, for keeping me accompanied during this long and arduous journey. I'd like to express some special thanks to Ke, Yunbo, and Sang, who are the light towers that guided my Ph.D. research directions; Zhuangying Xu, Tianyi Wang, and Xun Qian, who fought trustworthily by my side; Luis, Ana, Terrell, whose optimism inspires positive life attitude from me; and Professor Min Liu, Pawan, Siddharth, with whom I have the privilege to work together as the head TA of ME444.

Finally, I want to thank my family for their unconditional love and support while I pursue my Ph.D. degree far far away from home. I am dedicating this thesis to my mother, who is a brave brave soldier that fought the inevitable to the last second, and never gave up. I promise to you that I will live my life to the fullest and make you proud in heaven.

TABLE OF CONTENTS

Page

Page

Page

Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

ABSTRACT

Cao, Yuanzhi Ph.D., Purdue University, Dec 2020. Exploring Novel Human Smart-thing Interaction through Augment Reality Framework Design. Major Professor: Karthik Ramani Professor, School of Mechanical Engineering.

We have never felt so connected with the surrounding social and physical environment, thanks to the increasingly populating mobile computing devices and rapidly developing high-speed network. These technologies transform the everyday objects into *smart-things* and make us accessible to a large amount of digital information and intelligence relating closely to the physical reality. To bridge the gap between the digital interface and physical smart-thing, Augmented Reality (AR) has become a promising media that allows users to visually link the digital content to its physical target, with spatial and contextual awareness. Thanks to the vast improvement to the personal computing devices, AR technologies are emerging to popular realistic scenarios empowered by commercially available software development kits (SDKs) and hardware platforms, which makes it easier for human users to interact with the surrounding smart-things.

Due to the scope of this thesis, we are interested in exploring for the smart-things that have physical interaction capabilities with the reality world, such as Machines, Robots, and IoTs. Our overarching goal is to create ***better*** experience for users to interact with these smart-things, that is ***visual***, ***spatial***, ***contextual***, and ***embodied***, and we try to achieve this goal through novel augmented reality system workflow/framework design.

This thesis is based on our four published conference papers [1–4], which are described in chapters 3-6 respectively. On a broader level, our works in this thesis focus on exploring spatially situated visual programming techniques for human smart-thing interaction. In

particular, we leverage contextual awareness in the AR environment with the interactivity of physical smart-things. We explore (1) spatial and visual input techniques and modalities for users to intuitively interact with the physical smart-things through interaction and interface design, and (2) the ecology of human smart-thing through system workflow design corresponding to the contextual awareness powered by the AR interface. In this thesis, we mainly study the following spatial aware AR interactions with our completed work: (i) *Ani-Bot* demonstrates Mixed-Reality (MR) interaction for tangible modular robotics through a Head-Mounted Device (HMD) with mid-air gestures, (ii) *V.Ra* describes spatially situated visual programming for Robot-IoT task planning, (iii) *GhostAR* has presented a time-space editor for Human-Robot Collaborative (HRC) task authoring. (iv) while *AvaTutAR-study* has presented an exploratory study that provided valuable design guidance for future AR avatar-based tutoring systems.

We further develop the enabling techniques including a modular robotics kit with assembly awareness and the corresponding MR features for the major phases of its lifecycle; a lightweight and coherent ecosystem design that enables spatial and visual programming as well as IoT interactive and navigatory task execution with a single AR-SLAM mobile device; and a novel HRC task authoring workflow using robot programming by human demonstration method within AR scene with avatar reference and motion mapping with dynamic time warping (DTW). Primarily, we design system workflows and develop applications for increasing the flexibility of AR content manipulation, creation, authoring, and intuitively interacting with the smart environment visually and pervasively.

Based on our completed projects, we conclude this thesis by summarizing the overall contributions of my Ph.D. works, and briefly providing my humble vision for the future of AR.

1. INTRODUCTION

We are entering an era where we will soon be surrounded by all kinds of smart-things in our daily lives. These smart-things will enhance the living experience through digital communication and physical interaction. We humans will be connected seamlessly to the surrounding smart-thing network and form an ecosystem that is constantly absorbing and generating new information with the environment. This human-centered ecology has the potential to greatly increase the sensing capability and executive force of the human users, and therefore augmenting their environmental awareness and influence. The development and the technological advances in mobile personal computing devices and the high-speed networks are demonstrating a path to the above-mentioned future, by increasingly providing access to individual users of digital information and intelligence. This virtual information is usually hard to be detected or generated directly by a human being's sense and mind.

With the fast advancement pace of robotic technology, we can expect robots to come out from the factory to the household environment in no time. As these robots are becoming lower in cost, higher in flexibility and adaptability, and most importantly, smarter than ever, they are ready to fit into the user's home and work coherently with the rest of the stationary smart-things, also known as Internet-of-Things (IoT). To this end, a new ecosystem will soon be established where human users are augmented and enhanced by stationary IoTs and mobile robots. This ecosystem is capable of performing network digital communication (IoT) as well as physical interaction with the surrounding environment (Robot) and is therefore comprehensive enough to exercise real-life tasks.

Under this broader context, my research is driven by the following questions: (1) How to make the human user more intuitively and effectively expressive his/her will to communicate

with this Human-Robot-IoT ecology, and (2) How to design the ecosystem so that each party in the system works collaboratively and enhance each others' capability to achieve comprehensive real-life task through mutual endeavors. Our works in this thesis aim to bring out the strength of AR by designing both hardware and software workflow/framework/ecosystem around it, to create novel user experiences for human smart-thing interaction. Overall, our work is specifically designed to achieve the following features: ***Visual***, ***Spatial***, ***Contextual***, and ***Embodied***. These can be considered as the high-level design goals of my Ph.D. thesis, which are reflected throughout all my research projects.

1.1 Visual Interactions with Physical Smart-things

We have progressed a long way in controlling and programming physical devices via a digital interface. The traditional command-line User Interface (UI) (i.e. scripting) is unapproachable to novice users because it requires them to have a reasonable understanding of electromechanical systems and programming [5]. As a result, researchers have developed new approaches that require less cognitive load and pre-requisite knowledge. These new approaches generally fall into two categories of control interfaces: the Graphical User Interface (GUI), and the Tangible User Interface (TUI) [6–8]. The GUI is a graphical substitute for a similar command-line UI. Instead of using computer languages, users can drag and drop editable function blocks and link them in particular ways to program physical devices. This approach is much easier than scripting, and it also maintains a high level of control capability. Although GUIs tend to be more user-friendly than scripting, each GUI has a different interface that a novice user must still devote time and effort to understand and learn. Furthermore, like traditional scripting, the control interface of a GUI is in the virtual environment (a computer, phone, tablet, etc.), which detaches the users from the physical target they are attempting to control. Contrary to the GUI approach, many other construction kits utilize TUIs for programming, where the physical devices are programmed by hands-on

manipulation and the pattern of assembly. The TUIs tremendously reduces the gap between the control interface and its target because everything happens in the physical environment. Therefore, robotics kits that use TUIs are usually easy to use and, as a result, children with little-to-no training can create animations with their DIY robots. Although easier to use than the GUI approach, the TUI has its limitations in terms of control capability. TUI requires hands-on manipulation in almost all its interactions, which makes teleoperation hard to realize. Moreover, due to the nature of human hands-on manipulation and lack of digital interfaces, precise movement can also be hard to achieve in a TUI-powered robots [9]. To summarize, GUIs provide high control capability, but the gap between its control interface and the target could make it hard to be intuitive and expressive. The TUIs are easy to use but are limited in the control effectiveness.

To bridge the gap between the control interface and its target (TUI), while maintaining high control capability (GUI), we seek to combine both their merits and propose to use Augmented Reality (AR) or Mixed Reality (MR) technology as the control interface for interacting with the physical devices. As augmented reality enables virtual imagery to be seamlessly combined with the real world, it becomes a promising surrogate to bridge the physical and the digital world. In an AR scene, digital information and intelligence are usually represented in the form of graphical augmentations. The virtual images and the real images are combined through a video see-through or an optical see-through display. Further, the virtual imagery is registered with the real world in three-dimensional (3D) space and remains interactive in real-time [10]. In short, the newly emerging AR and MR technology enables the embedding of a versatile and malleable digital interface without impeding the inherent tangibility. We seek a spatial and visual programming interface for controlling the physical smart-things and therefore AR technology is serving as a major media in this thesis. The use of AR interface to control physical IoTs and robots has been widely explored by researchers [10, 11]. Contrary to the previous works, the works presented in this thesis aim

to further explore the different modality of user interaction in the AR context, in terms of controlling stationary IoT and mobile robot device. Moreover, we developed novel systems that exploit the benefit of AR, which is visually active with spatial and contextual awareness.

1.2 Framework Design of Approaching AR Interaction

The essence of Augmented Reality is to enhance the physical reality with superimposed virtual content. So that the embedded semantic information of the interesting physical object can be directly and visually accessible by the user. Simply looking at the target object, and its virtual information is floating above. Therefore the first key point of AR is visual, that users do not need to rely on any external scripting or interfaces to access the digital content and link with the target physical object, it's already aligned with the physical object. This characteristic allows users to naturally achieve point-to-point interaction without needing to worry about if they have chosen the correct target. Another significant feature of AR is the spatial and contextual awareness of the user. Because the digital interface is placed into a physical reality, environmental information can be easily taken into consideration when users want to program the devices for physical interactive tasks. For example, using an AR interface with virtual representation, the user can easily program a robot arm to grab objects from the nearby surrounding environment, by simply manipulating the virtual arm and simulate the motion before execution. Because the location and the spatial information of each smart-thing can be directly reviewed from the AR scene, the user can take advantage of it and reference it with the context environment. For example, to create a 3D trajectory for a robot arm to grab an object while avoiding the obstacles, or to set a collision-free path for mobile robots to navigate around the room.

One of the core focus of this thesis is to explore the modality of user interaction in an AR setup, which is determined by the framework design of the AR system. To have an AR system, several key elements need to exist and the parameter regarding these elements

defines the uniqueness of the AR framework. First of all, the virtual content that enhances the corresponding physical target. How is the virtual content created? Is it automatically generated based on the on-the-fly input data or is it pre-defined? Secondly, how is the virtual content aligned with its corresponding physical target? Does it rely on a camera vision-based image marker? Or Automatically perform the self-localization and registration? Thirdly, what is the human's role in the AR workflow? Is it only the data reader and command sender through the AR interface? Or can it exploit its intuition and reduce the system workload by participating in the AR content generation and registration/localization process.

1.3 Towards Collaborative Intelligence Through Human Smart-thing Interaction

There are three key elements in the ecosystem proposed in this thesis: Human, stationary IoTs and Machines, and mobile robots. Humans are the command issuer, they tell the system what do to based on the status of the system and the end goal. IoT devices and Machines are mostly stationary and they are good at the local job that is pre-defined, like turning the light on/off or start cooking food in the microwave. They can also actively sense environmental data and share them across the digital network. Mobile robot, on the other hand, is capable of physical object manipulation and room-level navigation. It serves as a flexible and adaptable media that links multiple stationary IoT devices together to form a physical network, that is capable of a more complex task than any of the IoT alone. In our proposed system, all these three elements are bound together by the AR framework to form an ecosystem. Traditionally, humans are only command givers, based on the available information including the readings from the IoT sensors. They review the status of the system remotely, press the button to issue a command, and observe the result then iterate the process. Traditional human user does not participate in the execution cycle and that means they have to give sequentially low-level command, to instruct the system to do one small thing at a time. In that sense, they are more like a system operator which requires them to

be highly skillful and familiar with the system. Otherwise, the system needs to be highly autonomous to dissect and execute human user's high-level command, this requires a series of intelligent processes, which include and are not limited to task distribution, resource scheduling, and so on. Compared to the previous approach, this is much harder in terms of the system's intelligence and is therefore not expected to exist anytime soon.

It has already been discovered by researchers that the proper collaboration between humans and machines is capable of higher productivity than both humans and machines alone. This might still hold even the level of machine intelligence further develop in the future. Because the intuition and originality of a human can never be replaced by any machine. In other words, even though machines are good at certain types of task, it is best to leave some other type of the job to human beings, who are more flexible, adaptable, and well, human. This is very similar to the collaboration between two people whose skill set and personality are very different, and yet they can be highly productive partners if working together and outperform each of them working individually. Therefore, the key concept we want to bring out in this thesis is human smart-thing collaboration through an AR framework. Why AR framework here? Because AR is capable to visually present the status of the system, which is highly intuitive and effective, and it also provides high spatial awareness for the user to make better contextual decisions. More importantly, with an AR framework, human users can physically place themselves inside the AR scene and perform as a special smart-thing device. Exploiting human being's intuition, it can help and dissect the otherwise very complex tasks for a fully autonomous system, for example, navigator path planning in a highly cluster scene. To this end, complex tasks can be executed through proper collaboration among human and smart-things. With the emergence of rapidly developing artificial intelligence, we believe this concept will play a crucial role in numerous future application scenarios like factory, office, and household environment.

1.4 Overview of Contributions

The purpose of this thesis is to explore system framework design for visual, spatial, contextual, and embodied interaction with smart-things. We seek novel user interaction in the AR setup via designing workflows. These system frameworks are later generalized and can be used by other researchers as guiding material towards the design for collaborative intelligence for human smart-thing interaction. We here summarize the contributions as follows:

- The design of system workflows that embeds Augmented Reality interaction with various enabling technology to achieve DIY robotic animation authoring, sequential task planning for mobile robots, Human-Robot Collaborative task authoring, and human-human skill transfer for machine task applications.
- Develop the corresponding system that involves exploration of the hardware design and the incorporated Augmented Reality features that promote novel interaction experience.
- Investigate the role of human user and place it in the center of the interaction loop, and aiming towards intelligent collaboration via human smart-thing interaction.
- Evaluate the technical performance and overall usability of proposed AR systems and interactions.

We will discuss the above summary in greater details in the rest of this thesis. Chapter 2 will describe the state-of-the-art related work compared to each of our work. In Chapter 3, we will discuss a novel system design that Mixed-Reality Interaction for DIY modular robotic. We will talk about the system workflow that achieves assembly awareness for the proper virtual model to be generated corresponding to the physical assembled DIY robot. We will discuss the hardware modular design as well as the incorporated mixed-reality

feature that promotes novel DIY robotics experiences. In Chapter 4, we will introduce a human centered approach for authoring sequential tasks for the Robot-IoT ecology. We will describe the lightweight workflow that utilizes one single AR-SLAM device to perform task authoring as well as robot execution. In Chapter 5, we will present a human-robot collaborative task authoring system. We will describe the system workflow featuring on embodied interaction (Programming-by-Demonstration) for HRC task authoring, using the recorded AR ghost as time-space authoring reference and editing agents, and motion mapping using DTW for acting out the HRC task. Chapter 6 will present an exploratory study on AR presence for machine task tutoring. We conducted a series of user study to compare the tutoring experience of the following AR presence: video, AR-only, AR with half-body avatar, and AR with full-body avatar. By analyzing the quantitative and qualitative results, we have extracted valuable design guidelines for future AR-avatar based tutoring systems. Chapter 7 summarizes the overall contributions of this Ph.D. thesis work. In Chapter 8, I will conclude this thesis by giving my prospective for the future of AR.

2. RELATED WORK

The research works presented in this thesis are in the area of Human-Computer Interaction (HCI) and Human-Robot Interaction (HRI). Specifically, we have intensively investigated novel interaction metaphors regarding Augmented Reality for visual and spatial programming. Our related work also covers construction robotics kit, mobile robot programming and task planning, and skill transfer via embodied demonstration. We will discuss the related work selectively for positioning our work in a broader background, and highlighting our contributions with respect to existing works. The following content is primarily based on the RELATED WORK section in our featured publications [1–4], with slight modification.

2.1 Modular Construction Robotic System with Embedded Mixed-Reality Interaction (*Ani-Bot*)

2.1.1 Interacting with DIY Robots

Due to the inherent tangibility in the DIY robotics process, previous works have developed several TUI approaches. Topobo and VEX robotics adopted the Programming-by-Demonstration method with kinetic memory to play back user-defined motions and animate the robot [12, 13]. Since only actuation modules can be programmed, this approach has a limited level of controllability. On the other hand, the Programming-by-Assembly approach requires no further programming once the robot is constructed, thus encouraging users to try different assembly configurations [14]. Therefore, this highly engaging TUI approach has been widely used in the area of childhood robotics education by Cubelets, LittleBits, MakerWear, etc. [15–19]. However, because each module is pre-programmed

for a specific function, complex robots require a large number of modules, which increases both the physical size and the difficulty of assembling. Furthermore, since the TUI robots are assembled and programmed for designated tasks, changing a task usually results in re-assembling a new robot architecture, which lowers the versatility and malleability.

Due to the limitations in TUI's controllability, many commercial robotics kits have adopted an additional GUI to control the robot, such as Lego Mindstorms, Tinkerbots, VEX Robotics, etc [13, 20–23]. However, most of these GUIs have been separated from the physical robot targets, thereby creating a gap which results in an inconsistent user experience. Alternatively, researchers have been exploring the merging of other interaction methods with DIY robotics. KIWI used scannable image-target-covered cubes to tangibly program the robot [24]. Handimate [25] and PuppetX [26] used hand and body gestures to control the crafted DIY robots, respectively.But although these control modalities show good interactivity, the lack of a fine level of controllability still remains an open issue. Mirror Puppeteering achieved user-defined playback animation with hands-on manipulation [27], yet it still required an external camera to track markers on the articulating parts. On the other hand, Ani-Bot's MRUI is superimposed onto the physical target that bridges the gap by directly interacting with the target object. More importantly, a coherent MRUI design preserves the tangibility of DIY robotics and the consistency of the user experience. By exploiting the advantage of a digital user interface, our system Ani-Bot is capable of achieving informative visualization and complex programming.

2.1.2 Assembly-Aware Construction

To effectively control modular robots, a virtual controller needs to be mapped with the physical target. Both GUI designs and controller-enabled interactions require manual correspondence for the mapping, which can be a tedious process for users. For example, when using Handimate [25], users need to manually set the gesture-actuator mapping

configurations on a mobile application before controlling the robot. This problem can be solved if the modular kit can be made aware of its own assembly configuration and can therefore accomplish the mapping automatically. Such an assembly-aware concept already exists in many TUI construction kits, including Cubelets and MakerWear [15, 19]. However, they address only electronic communication logics without geometric information of the physical assembly. In our case, geometric assembly-awareness is essential for deploying a mixed-reality user interface. Prior works have explored the subject via hardware connection [28–30] and computer vision approaches [31]. But most of these works only applied assembly-awareness to passive building blocks that involved no motions. In comparison, the Ani-Bot system provides a virtual geometric model of robotic modules that update and coincide with the physical assembly, thus allowing responsive interactions and active visual feedback.

2.1.3 Assembly and Design Guidance with MRI

Utilizing MRI, different modalities of virtual guidance have been explored to assist users in the assembly process. Henderson et al. overlaid instructions from the view of users' Augmented Reality (AR) headset [32], while Makris et al. displayed the corresponding virtual CAD model [33]. In terms of interaction media, some researchers used a virtual interactive tool [34], while others chose to directly manipulate the virtual model with bare hands for assembly guidance [35] and design simulation [36]. These works focused on providing assembly guidance for robotics/machines with pre-defined designs. Furthermore, without an external monitoring system on the assembly procedure, the guidance remained non-interactive. By embedding an RFID tag in each of the construction modules, Zhang et al. achieved real-time tracking and monitoring for non-mobile passive blocks that enabled interactive guidance [37]. Moreover, researchers have been coupling MRI with interactive design processes, including participatory design [38], decision making [39], and design

evaluation [40]. Driven by the needs from the creation and tweaking phases, we focused on incorporating suggestive design guidance for functional robot design. By achieving assembly-awareness, Ani-Bot provides users with interactive mixed-reality assembly guidance for re-configurable modular robotics construction.

2.1.4 Robot Operation with MRI

MRI has been investigated for interacting with robots. TouchMe and exTouch have demonstrated the process with mobile robots [41, 42]. Utilizing MRI with robotics, researchers have achieved human-robot collaboration for object manipulation [43], object delivery [44], and household sequential task instruction [45]. Besides being viewed through an AR tablet device [46–48], the control interface can also be projected directly onto [49] or near the physical target [50, 51] for ease of mixed-reality interaction. Furthermore, MRI is applied in industrial robotics for path planning [52, 53], spatial programming [54], and trajectory planning [55]. However, the above work utilized mixed-reality primarily for programming movements for robots with determined designs and configurations. Instead, we aim at investigating an MRUI with higher malleability for DIYing a re-configurable robot with both output and input modules. To the best of our knowledge, no prior work has attempted to or explored embedding MRI with DIY robotics; this, is our primary contribution of this paper.

2.2 In-Situ Visual Authoring System for Robot-IoT Task Planning with Augmented Reality (*V.Ra*)

2.2.1 Workflow of Human-Robot System

Due to the limited on-board perception capabilities and underdeveloped artificial intelligence (AI), the ad-hoc tasks in our daily environment which we take for granted are

still challenging for robots [56]. Thus, before it comes to an era of full autonomy and high level AI, a sophisticated human-robot interface is the key to author the domestic robots to accomplish any useful tasks. Within the authoring interface, users need to be spatially aware of the physical environment and the mobile robots. Previous works introduced an external vision system to track the robots and fed the live camera view to the interface [41,44,57–59]. However, this approach limits the authoring scene to the perspective of the camera only, which is usually fixed. In contrast, *Magic Cards* proposed an implicit command authoring workflow with human manually and spatially placing the task-representing paper tags [60]. Still, tracking from an overhanging infrastructured camera is prone to occlusion, especially in a cluttered scene such as a household environment. Further, recent researches employed mobile AR interfaces and associated the robots within the AR scene, e.g., with hand-held [61,62] or head-mounted [1] devices. Although the mobility allows users to move around and author distributed tasks from different perspectives, the limited field-of-view constrains the robots' navigation range.

Other works separated the authoring interface and navigation by equipping robots with on-board SLAM capabilities. This way, user referred to a scanned map of the real scene as authoring context and the robot conducted tasks using the same map [63–65]. However, the pre-scanned SLAM map, once created, remains static and cannot adapt to the changes in the environment. In fact, for an ever-changing scenario such as user's home, the system will be hampered with outdated SLAM maps. Informed by these previous works, we propose a mobile AR authoring interface with which users can spatially author the tasks by either explicitly defining navigation paths or implicitly visiting the IoTs by just walking to each of them. Moreover, we emphasize a transparent knowledge transfer between human and the robots by allowing robots to use the same AR device as 'eyes' and 'brain' directly. We further increase the adaptability of the robots against environment changes as we rely only on on-the-fly updated SLAM maps.

2.2.2 Robot-IoT Ecology within AR

An AR interface is spatially and physically aware of the environment by its nature [10]. Previous works have explored accessing and controlling IoTs through the digital representations superimposed in the AR scenes [66, 67]. But in these works, the augmentation relies on keeping the IoTs in the AR camera view, thus only allow for local interactions in a limited volume. Further, leveraging the SLAM embedded in mobile AR devices [68, 69], researchers also investigated spatially registered IoTs in the SLAM map to support embodied interactions in a larger space [70]. In addition, AR has been used to author and edit IoT programs in-situ [47]. Moreover, recent works further emphasized on multiple IoTs in the same environment, e.g., visualization of the data flow among sensors, logic programming between devices [71], and visual analytics of the fetched data [72].

For stationary industrial robot arm programming, AR motion planning allows users to preview the generated trajectories and examine potential discrepancies and collisions [52, 53, 55]. In a robot-IoT context, the mobility of the robots is a critical complementary element. We focus on authoring room scale navigatory tasks for visiting distributed IoTs and assume that the local manipulation are handled by robot itself. While simple graphical augmentation can be superimposed onto the video streamed from the external camera system [41,44,57–59] or projected to the physical environment [50, 51], we follow a mobile AR approach because a handheld [42, 61] or head-mounted AR device [1] allow users to freely move in the environment and inspect the augmentations from multiple perspectives. Besides authoring tasks for robots, researchers further explored using AR to debug robot behaviors [73, 74], passing knowledge to robots through demonstrations [54, 75], and interacting with the embedded AI decisions [76]. Although we do not develop these specific applications, our workflow shows potential to create robust test-beds for a variety of human-robot-IoT studies.

2.3 Human-Robot Collaborative Task Planning by Embodied Authoring with AR (*GhostAR*)

2.3.1 Human-Robot Collaboration Model

Many cognitive frameworks and computational architectures have been proposed for enabling and supporting teamwork between humans and robots [77]. One of the keywords in human-robot collaboration (HRC) is *adaption*: a robot interacting with people needs to reason over its uncertainty over the human internal state, as well as over how this state may change, as humans adapt to the robot [78]. While some previous work took the approach of human adapting to robot [79], and human-robot mutual adaption [80], the largest body of current HRC works have been focusing on a *lead-assist* collaboration type and empowering the robot to be an assistant and to adapt to human actions. Researchers have presented different mathematical models and formulations focusing on task allocation and communication via goal-oriented controller [81], on improving human-robot coordination through cross-training [82], and on efficient learning with human inference with joint-action demonstrations [83]. Other researchers emphasized on robot learning methods and frameworks and proposed *interactive primitive*. Along this thread, a series of studies demonstrated cooperative task learning with single [84] and multiple [85] primitives. Further, *probabilistic movement* model has been introduced to improve human-robot coordination [86] and action recognition [87]. These work primarily targeted at general mathematical solutions and learning methods for specific collaborative scenarios. However, it is still challenging to achieve applicable human-robot collaboration in real-world setups. In fact, most of the HRC tasks were pre-defined and simplified versions of intended scenarios [77]. Also many of these work require offline training with pre-capture data, which is not desired for on-site HRC.

On the other hand, our system complements the previous works by focusing on providing an in-situ HRC task authoring tool. We exploit the initiative of human users and enhance their capabilities with embodied interactions and AR interfaces. To better support a smooth workflow and rapid iteration of task plans, we adopt a real-time process for task authoring and collaboration acting without offline training. Taking advantages of AR interface, we also provide active visual feedback with spatial and contextual reference so that human and robot are always aware of each other during the collaboration.

2.3.2 Robot Programming by Demonstration

Robot programming by demonstration (PbD), also referred to as imitation learning, has become a popular method for programming and training robots. PbD reduces search space complexity for learning, supports natural means of embodied user interaction, thus enables flexible and user-friendly robot programming and training [88]. Extensive body of works have been done in developing methods and algorithms for learning individual motions [89–91] and compound motions [92, 93], as well as incremental teaching methods [94, 95]. So far, PbD has shown great successes in training individual robots to do specific tasks with offline data captures. When applying PbD into collaborative scenarios, additional reference is needed since robot is no longer operating in isolation. Instead, robots need to coordinate with the human partner, whose uncertainty depends on human's internal states upon actions. To achieve PbD for HRC tasks, previous works primarily relied on two people demonstrating the tasks where one of them plays the robot's role. The human demonstration is captured with motion tracking system offline and fed a computational model to generate robot policy at runtime [84–86, 96, 97]. The above approach is intuitive to practice and has been used in HRC task authoring including object handover and joint manipulation. However, this PbD approach is limited to simple and pre-determined task authoring due to the lack of visual interface for sophisticated editing. Moreover, as the offline demonstrations

usually happened in a controlled lab environment, the collaboration volume was constrained, e.g., most of the presented collaboration tasks were executed using a stationary robot arm.

GhostAR, on the other hand, exploits a visual interface and displays the captured human motion as ghost images in the AR scene. Using the AR ghost as time-space references, users can author the HRC tasks by manipulating a virtual avatar of the real robot collaborators. We emphasize instantiating PbD by supporting embodied authoring in our workflow. Our system allows for collaboration authoring of robots with various types of configurations. Further, when users perform the collaborations with robots, we allow users to use the same self-contained AR interface for motion inference.

2.3.3 Human-Robot Interaction through Augmented Reality

An AR interface is spatially and contextually aware of the surrounding environment by its nature [10]. Thus, it serves as an ideal media to bridge the digital interface and physical reality. For example, it has been used for visual and spatial interactions with robots [1,41,42] and smart devices [47,70]. AR for human-robot interaction has been widely explored across industrial motion planning [52, 53, 55], mobile teleoperation [41, 42, 98, 99], sequential task planning [44, 57–59, 100], and multi-robot controlling [101], analyzing [102], and debugging [103]. Previous works primarily treated AR as an control interface for robots operating in isolation. While AR was explored to display robot's intent for user visualization to achieve better collaboration [104–109], it has not been proposed to empower the entire life-cycle of HRC, from task authoring to collaboration acting. To the best of our knowledge, *GhostAR* is the first system that achieves incorporation of AR within a full HRC workflow, enabling natural embodied authoring with context-aware visual programming.

The key of HRC task authoring is to provide reference of the collaboration partner in a spatial and temporal manner during the authoring process, which in turn ensures correct time-space coordination when the HRC task is in action. By further exploring into human-

human scenarios, we have found several interesting AR works that achieve augmented collaboration through interactively reconstructing the surrounding environment [110], spatially visualizing the collaboration partners [111], and demonstratively externalizing user's body [112]. Informed and inspired by these recent works, we introduce a novel ghost visualization serving in a human-robot scenario for collaboration reference, authoring and editing, as well as simulation and preview of authored joint action plans.

2.4 An Exploratory Study of Augmented Reality Presence for Tutoring Machine Tasks (*AvaTutAR-study*)

2.4.1 AR Tutorials for Machine-related Operation

AR naturally supports spatially and contextually aware instructions for interacting with the physical environment. Researchers have explored various designs for AR-based text instructions [113–116], including numerical values [117, 118] for precise operational descriptions with quantitative real-time feedback. Symbolic visual guidance, such as arrows [119, 120], pointers [121], circles [122], and boxes [123], are commonly used for visualizing motion intent and guiding a user's attention. Besides text and symbols, prior works have also explored virtual 3D models of the interactive tools and machine components for a more comprehensive and intuitive visual representation, in use cases such as object manipulations and geometric orienting operations [124–127].

These means for creating AR instructions have been useful for tutoring physical tasks. AR-based training systems have been thoroughly explored and applied to complex real-world scenarios, such as vehicle maintenance training [114, 122, 128], facility monitoring [113, 115, 125], machine tool operations [117, 118, 124], and mechanical parts assembly [116, 120, 123, 126]. However, most of these AR-based training systems are focused on *local* interactions that involve very little spatial navigation and bodily movement as a part of the human-

machine task itself. To incorporate human motion into the task instruction, we propose virtual avatars for externalizing the human tutor. We acknowledge the necessity of the AR instructions in the existing work and additionally propose an avatar as a supplementary tutoring presence, mainly for the *spatial* and *body-coordinated* interactions in the machine task scenarios. We are interested in finding out if the added avatar visualization would improve the users' machine task tutoring experience and provide additional benefits that will inspire the future designs of intelligent tutoring systems.

2.4.2 Virtual Humanoid Avatar in AR/VR Training Systems

A virtual humanoid avatar is an animated human-like 3D model that embodies the human user's body movements, gestures, and voice information in VR and AR environments. It has been adopted as an expressive visualization media for human motion training. Chua et al. [129] built a Tai Chi training platform with a virtual instructor performing pre-recorded movements, where the students follow and learn asynchronously. YouMove [130] utilized an AR mirror to achieve full body gesture comparison with a projected tutor avatar. In terms of providing a better comparison with the virtual instructor, previous works [131–133] superimposed the virtual instructor together with the user's perspective in the AR view, enabling the user to align his/her body spatially with the virtual avatar. Moreover, OutsideME [134] adopted virtual avatars to externalize the users themselves as a real-time reference so that they can see their own bodies from a third-person view while dancing. While differentiating from regular-sized avatars, Piumsomboon et al. [135, 136] exploited a miniature avatar to empower collaboration between a local AR user and a remote VR user. Most recently, Loki [137] has created a bi-directional mixed-reality telepresence system for teaching physical tasks by facilitating both live and recorded remote instructions via avatars and RGBD point cloud.

These previous works reveal the virtual avatar's advantages in enhancing bodily-expressive human-human communication, for applications such as asynchronous learning, self-observing and training, teleconference, and MR remote collaboration. Nevertheless, the usability of the avatar as a tutor presence for training in physically interactive tasks has not been systematically explored. This paper proposes to use avatars for representing the human tutor's spatial and bodily movements in the machine task training scenario. A machine task is a compound mixture of multiple types of interaction, and existing tutorial visualizations do have their own advantages. Therefore, it is paramount for us to study *when* and *how* to use avatars in order to apply it effectively in machine task tutoring.

2.4.3 Authoring by Embodied Demonstration

An embodied demonstration enables a user to use the shape, positioning, and kinematics of one's body as spatial reference for digital content creation. Researchers have achieved complex hand-related 3D sketching [138], design of personalized furniture [139], and creative 3D modeling [140]. Additionally, the motion data of the demonstrations can be extracted from videos to produce step-by-step training tutorials for human body action [130, 141], first-aid procedure [142], and parts assembly [143]. Similarly, by mapping extracted body motion to virtual characters, users can act out stories and generate animations directly [144–147]. The embodied demonstration has also been applied in the area of human-robot interaction. Vogt et al. [148] and Amor et al. [149] used motion data captured from human-human demonstrations for programming human-robot collaboration (HRC) tasks. Recently, GhostAR presented a workflow of authoring HRC tasks by externalizing the human demonstration and using that as a time-space reference to program the robot collaborators [3]. Further, Porfirio et al. [150] applied the method of human demonstration for human-robot social interactions.

To summarize, an embodied demonstration empowers rapid creation of complex and dynamic content through intuitive and straightforward bodily interactions. It is, therefore, suitable for machine task tutorial authoring especially in a fast-changing working environment. We envision the embodied demonstration to become the predominant method for creating machine task tutorials in future factory scenarios. While we apply this method for generating the tutor contents, we also emphasize the design space of the tutor presence in AR.

3. MODULAR ROBOTICS SYSTEM WITH MIXED REALITY INTERACTION (ANI-BOT)

This chapter is a slightly modified version of "***Ani-Bot: A Modular Robotics System Supporting Creation, Tweaking, and Usage with Mixed-Reality Interactions***" [1] published in *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction* and has been reproduced here with the permission of the copyright holder.

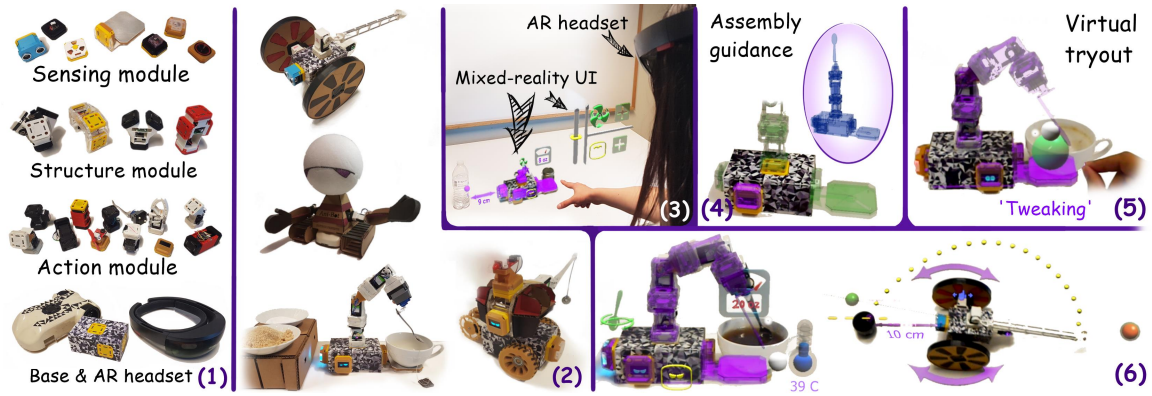


Fig. 3.1. Ani-Bot system overview: Ani-Bot provides users with (1) a modular kit that allows them to (2) assemble and construct robots with crafted DIY objects, and (3) use mixed-reality interaction to perform direct manipulation, sensor driven programming, and animation authoring. (4) The system can assist users in the assembly process, and (5) help them tweak ineffective designs through virtual tryout. (6) Taking advantage of mixed-reality, users can easily program their robots to perform environmentally interactive tasks, such as adding sugar to a teacup or shooting objects into a bowl.

Ani-Bot is a modular robotics system that allows users to control their DIY robots using Mixed-Reality Interaction (MRI). This system takes advantage of MRI to enable users to visually program the robot through the augmented view of a Head-Mounted Display

(HMD). In this paper, we first explain the design of the Mixed-Reality (MR) ready modular robotics system, which allows users to instantly perform MRI once they finish assembling the robot. Then, we elaborate the augmentations provided by the MR system in the three primary phases of a construction kit's lifecycle: Creation, Tweaking, and Usage. Finally, we demonstrate Ani-Bot with four application examples and evaluate the system with a two-session user study. The results of our evaluation indicate that Ani-Bot does successfully embed MRI into the lifecycle (Creation, Tweaking, Usage) of DIY robotics and that it does show strong potential for delivering an enhanced user experience.

3.1 Introduction

DIY modular robotics has a strong appeal to makers and designers since it can be used in quickly designing, building, and animating their own creation which opens the thrilling possibility of bringing imagination to life. The physical modular units inherently serve as tangible interactive interfaces within a DIY robotics process. Thus, developing a robotics kit with embedded Tangible User Interface (TUI) shows the potential to allow intuitive interaction in the DIY process [12, 15]. However, the versatility and malleability of such TUI's are limited when it comes to programming complex tasks involving a fine level of control [14]. To provide comprehensive controllability for the robotics kit, a Graphical User Interface (GUI) design has been adopted by commercial products such as Lego Mindstorms [20]. This separate digital interface, however, breaks the bridge between the physicality and the virtuality which are built through the TUIs [151]. To prevent inconsistent and fractured user experiences in the DIY robotics process, we seek for a seamlessly integrated workflow in which the intuitive tangible interactions are enhanced by a coherent spatially situated and contextually relevant digital interface.

The newly emerging Mixed-Reality (MR) technology enables the embedding of a versatile and malleable digital interface in the DIY robotics process without impeding

the inherent tangibility. Previous researches have attempted mainly in either assisting the assembling of passive building blocks [33, 34, 37] or controlling a pre-defined robot/machine [41, 42, 47, 48, 53]. Although we are inspired and motivated by these efforts, we focus on extending mixed-reality interaction to the whole lifecycle of modular robotics, namely **Creation**, **Tweaking**, and **Usage** [152]. Therefore, we propose Ani-Bot, a modular robotics system embedded with MRI. As demonstrated in Figure 3.1, while users are building their robots, the corresponding virtual model is automatically generated and superimposed with the robot from the view of HMD. Users can then visually control the physical robot by interacting with the virtual representation. With the Ani-Bot system, users can: (1) **Create** robot constructions with virtual guidance; (2) **Tweak** ineffective designs and perform virtual tryout; and (3) **Utilize** mixed-reality to make their DIY robots interact with the surrounding environment. To summarize, the main contributions of this paper are:

1. *System workflow*, which embeds MRI with modular robotics.
2. *Design of the Ani-Bot system*, including the mixed-reality ready 'plug-and-play' hardware and the incorporated MR features that promote a novel interaction experience.
3. *Evaluation results*, including the constructive feedback summary from our user studies that guides future endeavors.

3.2 Design Process and Goals

To design and fabricate the Ani-Bot system, we followed a user centered design process. We first developed a preliminary system with a few basic modules and MRI features. Then, by conducting a participatory design study with the preliminary system, we elicited critical design principles for our mixed-reality modular robotics system.

3.2.1 Participatory Design Activity

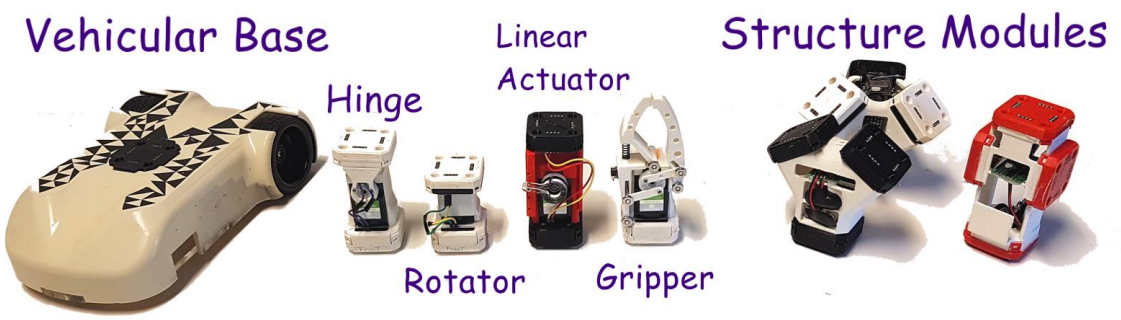


Fig. 3.2. Preliminary modular kit for the Ani-Bot system.

Our preliminary system is demonstrated in Figure 3.2 with basic 'on-target' direct manipulating UI. We recruited 5 participants (3 male) who had substantial experience in DIY robotics and asked them to use the system. We encouraged the participants to think out loud. Also, a semi-structured post-study interview was conducted. We focused on investigating the design of a mixed-reality ready modular robotics kit, a coherent user interface, and appropriate interactions. After the study, we found that participants unanimously requested more modules with various structures and functionalities in order to fully support the element of DIY. In terms of the design of UI and the interaction methods, users suggested that we fully exploit the advantage of the digital interface by displaying more informative and visually dynamic user interfaces with appropriate operations to interact with them.

3.2.2 System Design Goals

Based on the feedback about our participatory design activity as well as our own experience in designing the preliminary prototype, we have synthesized the following key design goals:

- **Plug and play.** The system should be mixed-reality ready for users as they play with the modules, with no configuration/preparation time so as to ensure fluid user experiences.

- **Low floors and high ceilings.** The MR system should be intuitive and easy to start, but provide high a ceiling for the level of control capability.

- **Visually intuitive.** The system's MRUI should be informative, provide active feedback, and be self-explanatory. Moreover, it should not be distractive and obstructive between users and their robots.

- **Support creative exploration** As a DIY platform, the system should support users' creative interactive exploration via both hardware and software designs.

3.3 The Ani-Bot System Design

3.3.1 System Workflow

Ani-Bot embeds MRI with DIY modular robotics; the workflow is illustrated in Figure 3.3. All modules in the system have processing power and can be physically connected with each other to establish network communication. By organizing the configuration data from each device, the robot is aware of its own assembly configuration and sends the data to the AR headset (Microsoft HoloLens [153]) to generate the corresponding virtual model. By detecting and tracking an image marker (Vuforia [154]) on the Base Module, the virtual model is superimposed onto its physical target for the mixed-reality interaction. The kinematics data of the virtual model are constantly transmitted to drive the physical robot. In this way, users interact with the physical robot by manipulating the virtual representation from the view of the AR headset.

3.3.2 Module Design

As shown in Figure 3.4, we expanded our preliminary module library based on feedback from the participatory design activity. **Base Modules** are the starting point of users' DIY construction, and they have three purposes: 1) realize tracking and detection of the virtual model via the image marker; 2) organize and transfer data between devices and the AR headset as a communication hub; and 3) provide a power supply for the connected devices. **Action Modules** provide various types of actions for users to interact with the real world. **Structure Modules** increase the structural diversity of the modular robot's configuration. They help the Ani-Bot system to better support users' DIY creation process. **Sensing Modules** read the environmental data, which are visualized in the MRI and used to program

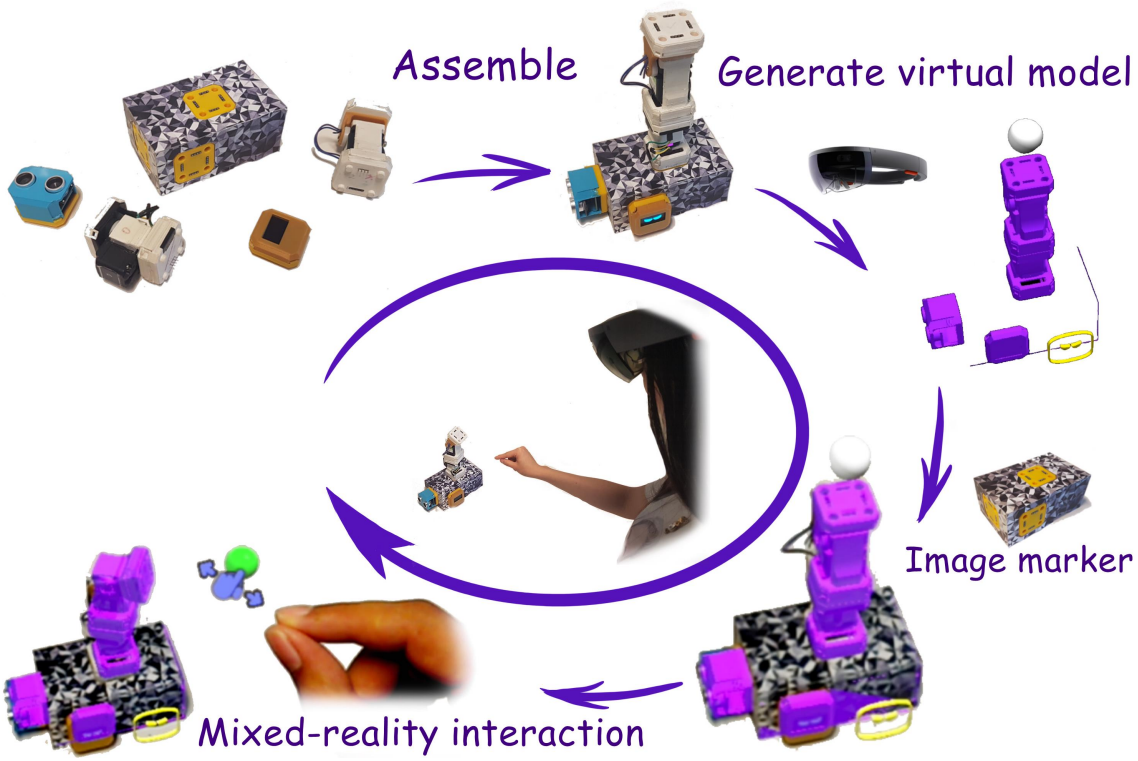


Fig. 3.3. Ani-Bot system workflow.

Fig. 3.4. Module library of the Ani-Bot system.

the robot behaviors. Together, all these modules compose the hardware modular kit that works coherently with the corresponding software interface to constitute Ani-Bot's mixed-reality modular robotics system.

3.3.3 Hardware Implementations

The modular design is illustrated in Figure 3.5 (2), using the Hinge Module as an example. The physical connection of the Ani-Bot's module is a Male-Female surface connection setup (42mm * 42mm), which is positioned by four cylindrical pins and secured by two embedded magnets (K&J Magnetics: DC1-N52). Most of the modules in the system have one pair of Male-Female connection surfaces to pass the power supply as well as an electric signal via four pins. All modules in Ani-Bot have only one Male surface, containing a customized PCB and a Bluetooth Microcontroller (RFduino). By reading from the *Sequence pin* and *Orientation pin*, the MCU knows its current position and orientation in the whole robot's assembly. The Base Module, as shown in Figure 3.5 (1), contains a Bluetooth MCU that receives the configuration data from all the connected devices. By integrating the data from all the devices, the Base is aware of the whole robot's assembly configuration instantly. The assembly configuration data are then transmitted to the HMD

via an on-board WIFI MCU (ESP8266) to generate the corresponding virtual representation of the physical robot. When receiving action data from the HMD, the Base Module organizes the incoming data and feeds them to the corresponding receiver device for action.

3.3.4 Interface and Interaction Design

The Ani-Bot system utilizes gesture-based interactions for most of its control and programming. These interactions are supported by the HMD (Microsoft HoloLens) and they require an *air tap* with one finger for clicking and selecting, and a *drag and drop* with two fingers for continuous manipulation. The MRUI in the Ani-Bot system is superimposed or floating nearby the physical robot for a seamless interaction experience. Our UI is designed

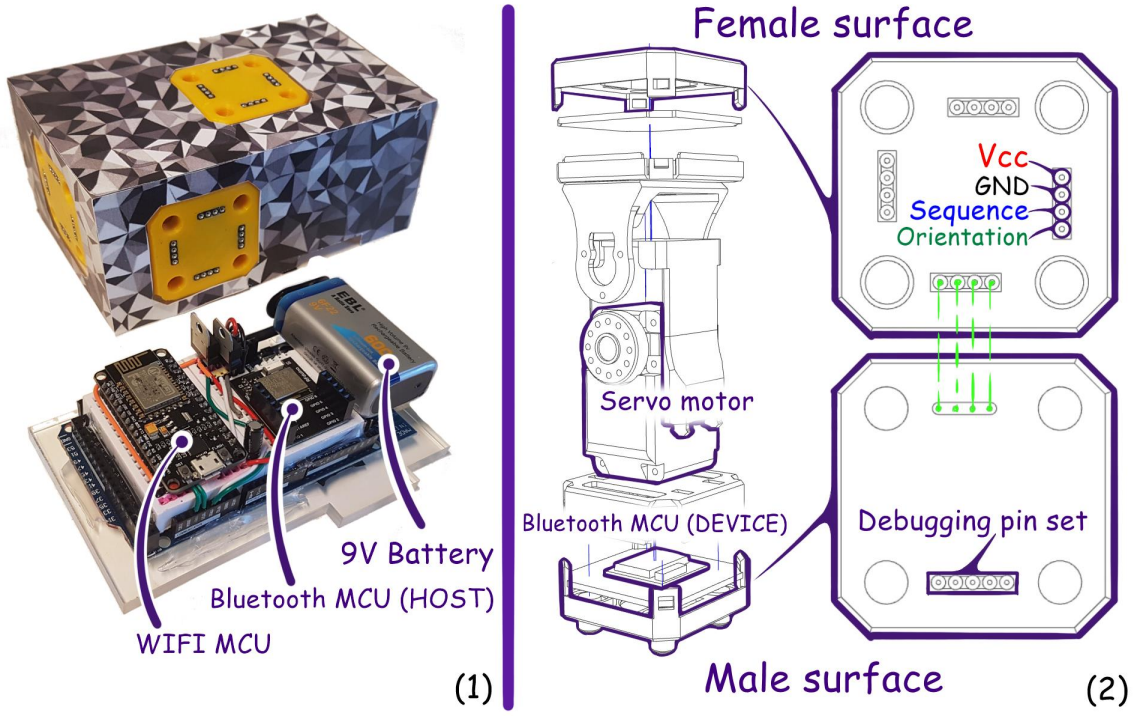


Fig. 3.5. Hardware design of Ani-Bot's module. (1) Cuboid Base module design setup. (2) Exploded view of the Hinge Module.

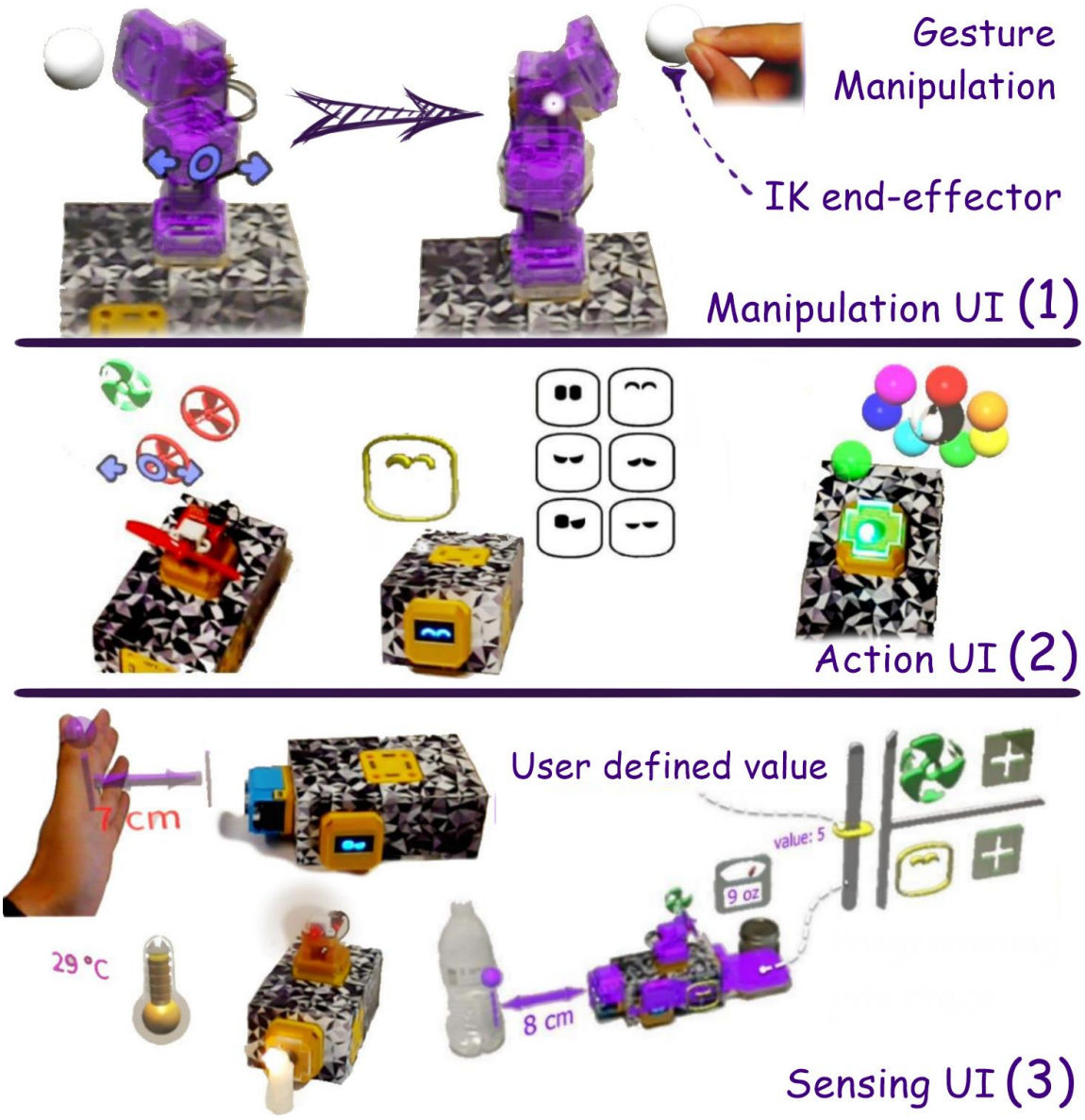


Fig. 3.6. MRUI in the Ani-Bot system consists of (1) Manipulation UI for actuators, (2) Action UI for the other action modules, and (3) Sensing UI for visualizing and programming the sensing modules.

for three categories of interaction (Manipulation, Action, and Sensing) according to the property of the physical module. For actuators such as Hinge, Rotator, Linear Actuator,

and Gripper, we superimpose the corresponding semi-transparent virtual model directly onto the physical module, as shown in Figure 3.6 (1). Users control these modules by manipulating the virtual models. They can achieve one DOF motion by manipulating each individual module or achieve multi-DOF motions by manipulating the auto-generated Inverse-Kinematics (IK) end-effector on top of the assembly tree. For the other action modules with discrete mode switching, a list-like UI is designed individually according to the module's function. As shown in Figure 3.6 (2), users can access these UIs to switch on/off the Fan module, change the facial expression on the Face module, and change the light color on the LED module, etc. In terms of the sensing modules, each environmental sensing value (distance, temperature, weight, etc.) is dynamically displayed as shown in Figure 3.6 (3). Moreover, users can program logic events by setting a user-defined threshold value and accessing the currently connected action modules. An example is illustrated in Figure 3.6 (3 right), where the user just programs the Fan module to turn ON when the weight sensing value is above 5 and the Face module to display 'happy' when the value is below 5.

Besides gesture-based interactions, the Ani-Bot system also exploits the HMD's multi-media capabilities to create immersive user experiences with active feedback. To avoid being overwhelming and distracting, we utilize voice commands and audio feedback for functions which have no explicit need to visualize, such as mode transition and menu navigation.

3.4 Modular Robotics With MRI

In this section, we demonstrate and discuss the augmentation offered by MRI in the Ani-Bot system. Specifically, we illustrate the system's designated features for the three phases of a construction kit's lifecycle: **Creation**, **Tweaking**, and **Usage**, respectively. We showcase how these embedded MR features can enhance the user experience for DIY robotics.

3.4.1 Creation

The process of playing with a modular robotics kit begins with the assembly. Ani-Bot encourages users to freely explore different assembly configurations by providing a rich module library. In addition, the system can also fully or partially assist users in the assembly process. Utilizing MRI, Ani-Bot provides users with *mixed-reality assembly guidance* for existing designs (Figure 3.7 (1)). The virtual guidance is interactive and gives real-time assembly feedback. For example, the color of the virtual model will change when the corresponding physical module is correctly assembled. Besides the full assembly manual, the system can also provide partial functional structure suggestions according to the key input module. For example, in Figure 3.7 (2), upon detecting the 'Distance Sensor,' users can activate the *functional suggestion guidance* by voice command, and the system will display a 2-DOF thrower setup with default adjustable structure parameters.

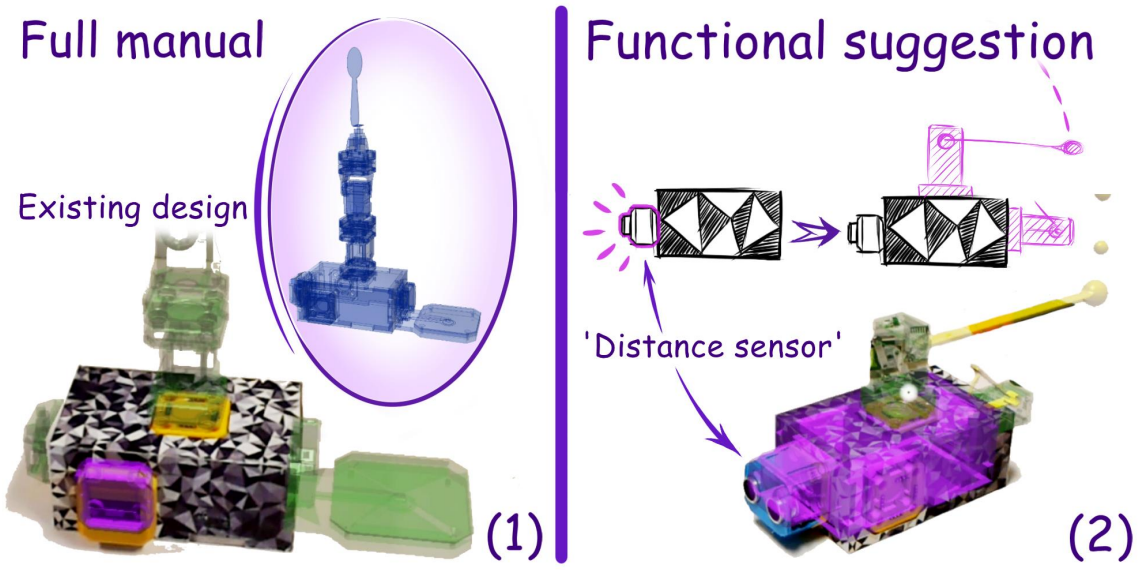


Fig. 3.7. Creation with MRI: mixed-reality assembly guidance. (1) Full MR assembly manual for existing design. (2) Suggestive guidance based on key input device.

3.4.2 Tweaking

When encountering ineffective designs, users will start the iterative process in order to explore and find a working solution, namely Tweaking. Instead of physical tweaking, which requires effort for iterations with real robots, Ani-Bot provides a virtual tryout feature for users to tweak the ineffective designs into a working configuration. As demonstrated in Figure 3.8, upon removing the end-effector, users can activate the '*Tweaking Mode*' through a voice command. The system will then display a series of suggested derivative configurations based on the current physical setup. Users can try different virtual assemblies and compare their performance in the mixed-reality simulation to find better solutions.

Fig. 3.8. Tweaking with MRI: virtual tryout for functional improvement. Tweaking a robot manipulator setup so that the spoon tip can reach inside the bowl.

3.4.3 Usage

One of the main advantages of mixed-reality is its ability to the merge of a virtual interface with its corresponding physical target. By exploiting this property, Ani-Bot allows users to easily control their robots to effectively interact with the surrounding environment.

For instance, Ani-Bot's sensing modules expressively visualize the input data (temperature, distance, force, etc.) from the surrounding environment (Figure 3.1 (3,6)). In addition, each sensing module offers the ability to program sensor-driven logic events with the programming UI (Figure 3.1 (3)). In this case, the user just programs the 'Fan Module' to turn on when the weight exceeds the set value, otherwise the 'Face Module' displays a smiling expression.

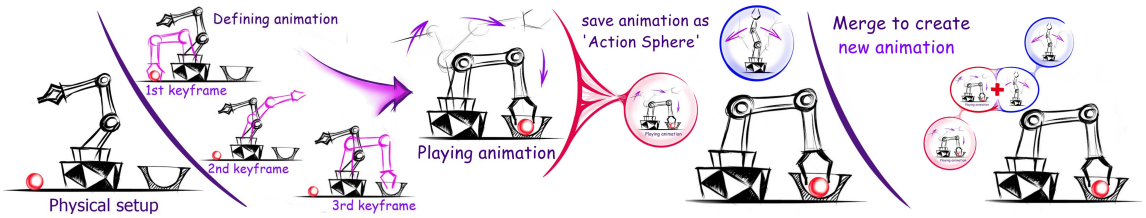


Fig. 3.9. Mixed-reality animation authoring and management.

Besides the sensing programming, Ani-Bot allows users to create and manage keyframe animations that enable their robots to execute automatic actions. As illustrated in Figure 3.9, after activating the '*Animation Mode*,' users can manipulate the virtual model to set the keyframes (the physical robot will not move in '*Animation Mode*'). Upon playing the animation, the robot will automatically transit through the defined keyframe positions and complete the action. Each animation can then be saved as an interactive '*Action Sphere*,' which floats near the physical robot and plays back the animation when tapped. Users can create multiple animations and intuitively manage them. By dragging an '*Action Sphere*' into another one, users can merge them together and create a new '*Action Sphere*' which has the combined animation. In this way, users can easily achieve complex animation authoring to create environmentally interactive and storytelling like animations.

3.5 Example Applications

Figure 3.10 demonstrates the four use cases we have created to showcase the diversity and controllability of our system, including two service robots (1,3) assisting environmentally interactive daily practice and two storytelling robots (2,4) with expressive emotions and stylish actions. The **Robot Thrower (1)** is able to display the predicted shooting projectile to guide users to manually hit the targets with pinpoint accuracy. The thrower can also utilize the distance sensor to automatically adjust the shooting angle based on the distance reading from the target. The **Emotional Fire Fighter (2)** is a fully equipped vehicular robot with a front temperature sensor for detecting candle fire. He is never too shy to show his emotions via the face modules and he shows no hesitation in using his head-mounted fan and hanging hammer to put out a fire. The **Tea Maker (3)** is a smart service robot with an arm. By visualizing the temperature and weight of the teacup, users can customize their favorite beverage by programming the Tea Maker to automatically add sugar and keep stirring until the tea is ready to serve, which is detected by the temperature sensor and indicated by the blinking LED and the smiling face. The **Dancing Robot (4)** is a DIY character with a big head and gloomy expression. Users can program him to make numerous amazing dance moves.

3.6 System Evaluation

To evaluate the Ani-Bot system, we invited 20 users to participate in our two-session user study (10 for each).

3.6.1 Session 1: System Usability Evaluation

We designed four tasks for the first study session featuring the key functions of the system. We invited 10 users (7 male), 7 of them in the 20-25 age range and 3 in the 25-30

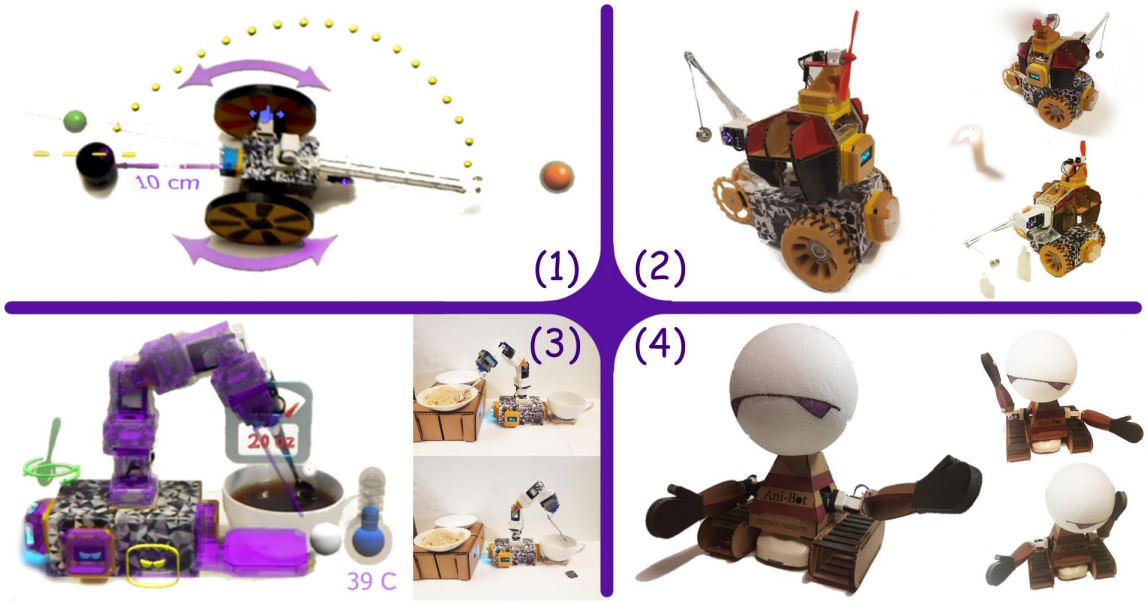


Fig. 3.10. Use cases demonstration of the Ani-Bot system. (1) The Robot Thrower. (2) The Emotional Fire Fighter. (3) The Smart Tea Maker. (4) The Dancing Robot.

age range, with varies backgrounds. The goal of this study session was to evaluate the usability of the Ani-Bot system and explore the user experience of DIY robotics with MRI.

Procedure. Session 1 took about 1.5 hours for each user, including a tutorial to introduce the HMD device and the Ani-Bot system (20 mins). We adopted one of our use cases ('Tea Maker') as the evaluation prototype due to the comprehensiveness of its functionality and the complexity of its physical structure. We dissected the prototype into four manageable tasks focusing on the three phases: **Creation**, **Tweaking**, and **Usage**. Users were given a questionnaire with Likert-type items and subjective questions after each task. Some of the more representative results are intuitively displayed as a colored scale bar for each task. Each Likert-type item is graded by users from 1 to 5, where 1 means strongly disagree and is colored in red, while 5 means strongly agree and is colored in green. The scale bars are

aligned with positive answers (yellow, yellowgreen, green) on the right and negative answers on the left (red, orange). (N = number, U = user)

Task 1: Assembly Guidance, Paper Manual vs MR Manual

For a design configuration with seven modules (Figure 3.11), we asked users to complete the assembly using both a paper manual and an MR manual as a guidance (random order).

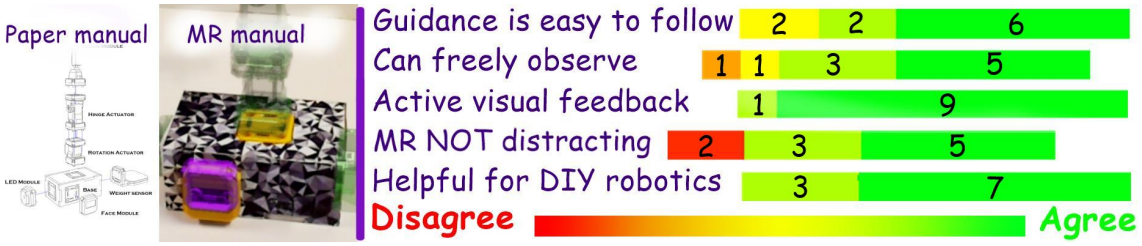


Fig. 3.11. Task 1: Assembly guidance. Paper manual vs MR manual.

Feedback and Discussion. Due to the simplicity of this design, users were able to complete the assembly almost equally rapidly (less than 30 s) and accurately with both manuals. It is noted that the point of this task was not to systematically study the time efficiency and accuracy between the two approaches. Rather, we tried to focus on exploring the user experience of MR assembly guidance and compare it with the most commonly used paper manual method. From the post-study survey, most users (N=8) preferred the MR manual over the paper manual for assembly guidance. The two users who disagreed felt that the overlaying virtual model was distracting and they suggested a switch function to toggle the MR interface. *"I am having some trouble differentiating the virtual model from the real one (U2)."* However, they still admitted that the MR guidance for the DIY robot was useful (reporting 5 and 4). The assembly process included identifying the right module and putting it in the right location. Ani-Bot's MR system allows users to freely observe

the guidance virtual model from different perspectives for module identification. One user disagreed with this because of the limited field-of-view of the HMD device. *"It is hard for me to see the whole guidance model without moving my head (U7)."* We found that the virtual feedback for confirming the assembly correctness was particularly appreciated by the users. *"The color change feedback really assures me about my assembly and makes me confident."* Overall, we found that the expressive visual feature as well as the interactivity of the MR guidance provided an engaging and entertaining assembly experience. *"I really like it, it's fun and makes me want to try more. (U5)"*

Task 2: Design Tweaking, Physical vs Virtual Tryout

In this task, users were asked to improve the performance of the initially ineffective robot design by changing its assembly configuration. The goal was for the robot arm's end-effector (spoon) to get inside the bowl (Figure 3.12). Users were asked to perform tweaking in both ways with a randomized sequence.

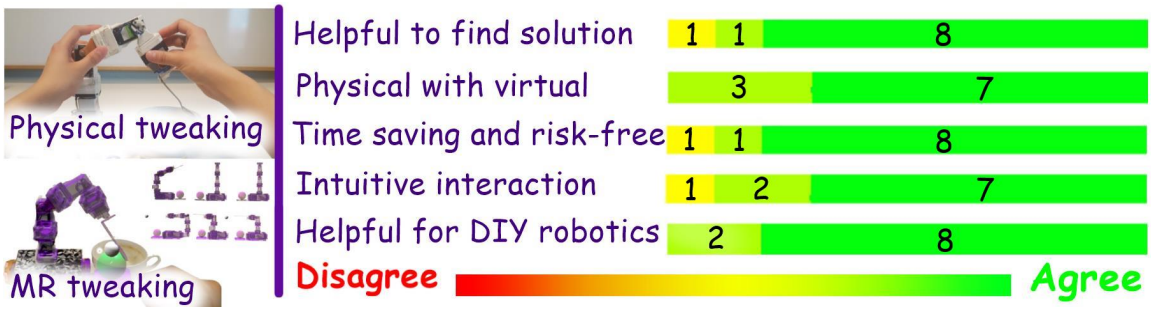


Fig. 3.12. Task 2: Hands-on tweaking vs virtual tryout.

Feedback and Discussion. After this task, most users reported that they preferred MR tweaking over the physical tweaking (N=8). They particularly liked the grafting of the virtual model on the physical modules, while both moved together corresponding to

users' simulating manipulation. *"I think the virtual/reality simulation really helps me to understand the dynamics of the robot (U7)."*. According to the survey and our observation, users generally enjoyed the MR tweaking due to its time-saving effectiveness and risk-free characteristics. *"MR is fast and easy to try. I enjoy testing different options (U4)."*. *"The MR tweaking reduces the cost and risk for revising physical robots (U1)."*. As for the two users who preferred hands-on tweaking, they believed the added complexity of the MR tweaking was unnecessary, but they still appreciated the visualization and simulation ability offered by the MR tweaking. To summarize, the system's virtual tryout feature effectively helped users to identify the configuration for performance improvement. *"With MR, I know it works and I do not need to finish assembling something to test (U8)."*

Task 3: Programming Sensor Driven Events

During this task (Figure 3.13), users were given a simple setup with two action modules (Fan, Face) and two sensing modules (distance, weight). They were asked to interact with the sensing modules and define the logic events triggered by the environmental data to drive the action modules. Users were first given a brief demonstration and then asked to freely explore the feature and define the logic events by themselves.

Fig. 3.13. Task 3: Programming sensor driven events.

Feedback and Discussion. All the users found it easy to program a logic event except for U4, who answered 1 in this question. She initially struggled to understand the working mechanism and suggested adding more text or audio instructions to guide users. Despite this, she still agreed that the feature was useful for DIY robotics (score=5). Users generally enjoyed the sensing visualization, which presented the environmental information in a tangible and interactive way. *"This feature makes reading from a sensor so intuitive and entertaining (U3)!"* From the survey results as well as the subjective comments, we found users particularly appreciated the system's fast and easy approach to programming fairly complex events (N=9), which could even elicit and promote interest in DIY robotics. (*"I like this instant programming. It is easy and can make people more interested in DIY robots (U8)."*)

Task 4: Creating Environmentally Interactive Animations

In this task, users evaluated the animation authoring feature in the system. They were asked to define the keyframe animations that facilitated the 3-DOF robot arm with a spoon end-effector to automatically add sugar to the teacup (Figure 3.14). The average time cost for this task was about 15 min.

Fig. 3.14. Task 4: Creating mixed-reality keyframe animations.

Feedback and Discussion. Considering the difficulty of task 4, we were surprised to find that most users (7/10) successfully accomplished the task with just one shot. We observed great excitement from the users when they have achieved this complex animation with just a few operations. *"I have never programmed a robotic arm so quickly and easily (U10)!"*. Based on the survey results, we found that users were highly satisfied with the MR animation authoring feature in the system. They appreciated the system's ability to create automatic actions *"Animation is useful to do repetitive work (U1)"* that enabled them to quickly explore their ideas with physical robotic movements. *"I like to set several actions at one time and make it keep doing what I want it to do (U9)."* Furthermore, they found the system to be very helpful for programming environmentally interactive tasks due to the active visual feedback from the mixed-reality view. *"I can use the surrounding objects as references when I define the animations; this makes it so easy for me to program my robot around them (U1)."*

Summary. After this session, users generally agreed that the interaction of the system was intuitive and effective (avg=4.6) with well integrated functions that help to provide elevated user experiences in DIY modular robotics (avg=4.56). The System Usability Scale (SUS) survey was also deployed after the study session to evaluate the system with an average score of 83.25 and a standard deviation of 6.8, which indicated high usability of the proposed system.

3.6.2 Session 2: Creating and Animating DIY Robots

The element of DIY is significant to Ani-Bot as the system is designed to support add-on DIY creativity by providing a modular platform and an effective method for controlling and programming. To evaluate this, we invited 10 users from diverse backgrounds for the second session of the study (7 male), with 8 of which in the 20-25 age range and 2 in the 25-30 age range.

Process. Session 2 lasted about 80 min including a 20 min system tutorial. Users had full access to the system's modular kit, as well as various DIY crafting tools and materials to create their own robot. During the session, they were asked to design, craft, assemble, and animate their own DIY robot.

Fig. 3.15. Results from the open creation study session showcasing users' DIY robot. (1) Mr. Destroyer (2) Box Porter (3) Peru Totem (4) 3-head Nezha (5) Robot Bandit (6) The Whomping Willow (7) Sun-eye Monster (8) Cheerleader (9) Robo-Cop (10) The Hulk

Results. Figure 3.15 showcases all the DIY robots created by the users during the open creation session. We observed a large variety from the end results, ranging from humanoid characters (4,8,10), to mechanical characters (1,2,5,9), to object-based characters (3,6,7). Each user's DIY robot consisted of 7-9 modules, which indicated high complexity involving multiple degree-of-freedom movements. All animations were created by the users uniquely for their characters which truly brought the robots to life. For example, Mr. Destroyer (1) does not hesitate for one bit to shred anything he sees (detected by the distance sensor) with his blade, claw, and drill bit. The Box Porter (2) is a diligent fellow that specializes in moving anything delivered to him (activated by the weight sensor) to the designated area. The Cheerleader (8) is a lovely girl waving 'De-Fence' for her team, while the Whomping Willow (6) furiously bashes the 'flying car' trapped on its trunk.

Feedback and Discussion. From the post-study survey, we found that most users appreciated the system's coherent work-flow to create and animate the DIY robots. *"It was especially fun to make the cheerleader and then see her actually move and do things."* From our observation, we found that the idea of combining DIY with robotics really promotes users' interest in exploring more features and functionalities of the system. *"Just being able to add skin for a better appearance on my robots also encourages me to explore more designs and shapes."* During the ideation process, many users liked to test-play with multiple modules and put on the HMD to quickly test the animation performance.*"It responded well to what I wanted it to do."* The plug-and-play seamless user experience as well as the real-time responsive mechanism was highly appreciated by the users.*"You don't need to worry how you are going to articulate your model for creating the control code. You just plug and play."*

3.7 Limitation and Discussion

During our user study, the most common complaints we received were about the HMD device, specifically, its form factor and interaction modality. *"It is too heavy and makes me feel dizzy after some time." "The display view is too small that I have to move my head to see the whole scene." "I don't like the mid-air gesture interaction; it feels awkward and is not very accurate."* Because our system was built on the HMD device (Microsoft Hololens), the limitation of the device became the limitation of our system. Furthermore, the device confined the mixed-reality experience exclusively to the headset wearers, which inevitably impeded the distribution and social impact of the system. This suggests that we need a better MR platform with a user-friendly form factor, intuitive interaction, and most importantly, public viewing and/or accessibility.

Another limitation of the system is caused by the MR tracking mechanism. The Ani-Bot system is currently implemented with an image marker on the Base Module. This requires an initializing detection and tracking process each time users start a new assembly. Moreover, the tracking results can be corrupted by occlusion from other connected modules. *"The virtual model is sometimes mis-aligned with the robot."* The tracking mechanism is the key reason for requiring the Base Module, which constrains the physical structure design of the modular system. This limitation can potentially be addressed by incorporating markerless tracking in the future.

It is interesting to note from the evaluation that users were always asking for more feedback (audio, visual, tactile) and natural control methods (gesture, voice). Future endeavors should therefore focus more on the modality of the interaction approaches to achieve comprehensive control with minimum cognitive load. Furthermore, the system should better understand its users and execute low level operations automatically. With the rapid development of AI technology, a new balance can be established and constantly adjusted between robot intelligence and user-involved controllability.

3.8 Conclusion

In this paper, we present a novel mixed-reality modular robotics system, called Ani-Bot. We explore and investigate embedding a coherent MRI for DIYing a modular robot. Our use cases as well as the system usability study have evaluated and verified the augmentations for modular robotics by embedding the mixed-reality interaction. The results from the open creation study have demonstrated the Ani-Bot system's capability to support both creating and animating DIY robotics. To this end, our system has shown a strong potential for delivering in-situ and novel user experiences for DIY robotics.

4. SPATIALLY AND VISUAL PROGRAMMING FOR ROBOT TASK PLANNING (V.RA)

This chapter is a slightly modified version of "***V. Ra: An In-Situ Visual Authoring System for Robot-IoT Task Planning with Augmented Reality***" [2] published in *Proceedings of the 2019 on Designing Interactive Systems Conference* and has been reproduced here with the permission of the copyright holder.

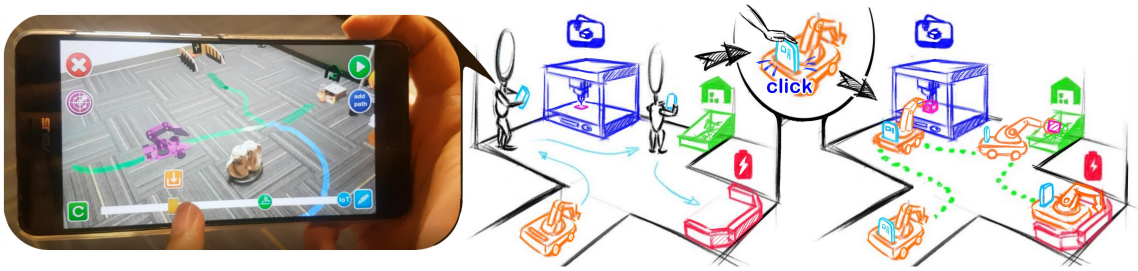


Fig. 4.1. V.Ra system workflow. Using an AR-SLAM mobile device, the user first spatially plan the task in the AR interface, then place the device onto the mobile robot for execution. The room-level navigation of the robot is guided by the SLAM feature on mobile device.

V.Ra - Virtual Robotic assistant, is a visual and spatial programming system for robot-IoT task authoring. In V.Ra, programmable mobile robots serve as binding agent to link the stationary IoTs and perform collaborative tasks. We establish an ecosystem that coherently connects the three key elements of robot task planning (human-robot-IoT) with one single smartphone device. Users can perform visual task authoring in an analogous manner to the real tasks that they would like the robot to perform with the Augmented Reality (AR) interface. Then placing the device onto the mobile robot performs the same tasks the users did in a what-you-do-is-what-robot-does (WYDWRD) manner. The mobile device mediates the interaction between the user, robot and IoT oriented tasks, guiding the path

planning execution with Simultaneous Localization and Mapping (SLAM). Our use cases and evaluation results have demonstrated V.Ra's capability of enabling robust room-scale navigatory and interactive task authoring.

4.1 Introduction

The vision of *ubiquitous computing* has been emerging rapidly as the Internet of Things (IoT) based electronics are getting smaller, lower in cost, proliferating and being embedded in our everyday environment. Typically, human-IoT interactions take the form of transforming IoT data into informative knowledge, augmenting human sensory capabilities, and assisting humans to make correct and efficient decisions [155]. However, the IoT devices are mostly stationary and have limited physical interactions particularly with each other. In conjunction, the concept of Internet of Robotic Things (IoRT) has not been widely explored in practice across the IoT and robotics communities [156], and an authoring system for such robot-IoT interactive task planning is underdeveloped [157]. We envision the emergence of programmable mobile robots in a near future to serve as key medium to conduct coordinated and collaborative tasks with surrounding IoTs. In this vision, the mobile robots are combined with the embedded multiple stationary IoTs to create new types of workflows and in addition also extend humans' motor capabilities.

Current user interfaces are often designated to either IoT or robots only, without considering the robot-IoT ecology. Contemporary IoT devices allow access and control through offloaded mobile interfaces. With additional web-based services such as IFTTT [158], users can also coordinate multiple devices working with other productivity tools or social medias via active human-IoT communication [155, 159]. Even in these coordinated works, the IoT tasks are rather spatially independent. In these cases, conventional graphical user interfaces (GUI) mostly suffice the IoT-only interactions which are insensitive to their spatial distributions. In contrast, to command mobile robots to complete distributed tasks, the

significance of spatial-awareness for authoring interfaces varies depending on the level of the robots' autonomy. For highly autonomous robots driven by embedded intelligence, users simply need to assign tasks using high level instructions requiring less spatial information, e.g., instruct a Roomba [160] to clean the room. However, besides the simple specific tasks, the robots' intelligence remain underdeveloped for a majority of the ac-hoc tasks in less controlled environments including our daily household environment [161]. Therefore, we develop interfaces and workflows to program robots that bridge the mediation between IoT embeddings and overcome these complexities by exploiting users' innate capabilities. From this perspective, the contextual visualization and spatial awareness of the environment are essential and utilized by us to ensure the efficiency of the authoring UI [10].

In the context of robots-IoT ecology [156], we design, prototype, and demonstrate a coherent authoring interface specializing at robot-IoT interactions with human-in-the-loop through: (i) the pervasive sensing capabilities and the knowledge embedded within the IoTs that facilitate the robots to complete tasks at a semantic level; (ii) IoT devices serve as spatial landmarks to navigate the robots around, and (iii) in addition the robots manipulate the IoT devices or interact with the machines and objects physically. These newly introduced aspects have not been developed, to the best of our knowledge, in the existing human-IoT or human-robots programming UIs.

The emerging augmented reality (AR) shows promise towards augmenting and interfacing with the physical world. In fact, AR interfaces have been introduced for IoT and robots respectively. For example, Reality Editor allows users to visually program the stationary IoT devices which are affixed with fiducial markers [47]. In a similar manner, robots have been attached with tags and tracked through the users' AR camera view [1, 42, 61]. However, the robots and the IoTs remain locally registered in the AR only, e.g., to resolve the spatial relationship between a robot and an IoT, a user has to keep both of them in the same AR camera view. To register multiple agents globally and coordinate them spatially, some alter-

natives including external tracking systems (e.g., infrastructured cameras [41,44,58,60]) and pre-scanned and manually tagged environment maps [63–65] have been proposed. But these approaches further constrain deploying robots to ad-hoc tasks in our daily environment.

On the other hand our approach leverages the advancing SLAM techniques to globally associate the user, IoTs, and robots together. Users first freely examine and explore the IoT environment within a mobile AR. Then within the same AR scene, users seamlessly transfer their insight about the tasks regarding the environmental factors such as the path planning, as well as the semantic knowledge such as the situational awareness from IoTs to the robots. Further, SLAM also enables a novel embodied programming modality, namely, users demonstrate a sequential chaining of distributed tasks to the robots by physically visiting the IoTs. In addition, since both AR and the robots' navigation share large commonalities in terms of spatial awareness of the environment, we support a smooth exchange of human knowledge between the AR device and the navigation module of the robots. The robot now has perceptive knowledge of the physical environment, the interactive knowledge for the IoTs, and is ready to execute the planned task from the user. To this end, we present V.Ra, an in-situ authoring interface for robot-IoT task planning using a mobile AR-SLAM device. The key contributions of this paper are as follows:

1. **V.Ra workflow** that uses one AR-SLAM mobile device for robot-IoT task authoring and execution, so that the human-robot-IoT interaction is bound together synergistically.

2. **Authoring interface design** that enables path planning, logic driven event scheduling, task chaining, and knowledge transfer to the robots, as well as spatial awareness and contextual feedback.

3. **Use cases and evaluations** demonstrating and verifying that V.Ra supports robust room-scale household navigatory and interactive task authoring within our prototyped robot-IoT ecosystem.

4.2 Design Goal

We followed a user-centered design approach to derive the design goals of our system. We conducted conversational style informative interviews for our study. We first explained the context of a household robot-IoT ecology to the interviewees, then we asked them to think about a scenario where users author multiple tasks to the robots and reveal their considerations and requirements for the system. We interviewed 42 people totally, including students, staff, and professors in the university with various backgrounds. Each interview took 6-10 minutes with the conversation recorded in audio. After analyzing the interview records, we identified the requirements and preferences from the participants. Combining the interview analysis with our vision of robot-IoT ecology, we propose the following four Design Goals (DG).

DG1: Easy and Instant Deployment. Less dependencies on the environment is preferred so that the system can be used instantly even for a new environment. Especially for the tasks handling chores, if the preparation takes even longer than finishing the chores by users themselves, the acceptability of the robot would be severely decreased. Thus, our system should be developed in a *self-contained* and *plug-and-play* manner to avoid the environmental dependencies and allow for *in-situ* authoring.

DG2: Physical and Spatial Awareness. We aim towards leveraging users' innate knowledge of the environment to instruct the robots to accomplish tasks in a household environment which are unstructured and ever changing. A physical and spatial aware authoring interface would allow users conveniently and accurately express their intents and transfer them to the robots.

DG3: Iterative Process with Feedback. Many participants unanimously required the system to keep them informed about its operating status during the entire process with active feedback. Further, our system should support users to visually preview and iterate the authored actions so that the efficiency of a sequence of distributed tasks can be improved.

DG4: Low learning curve. Participants suggested to develop the system based on easy-to-access devices and tools so that the basic interaction modalities remain familiar to novice users. Compared to abstracted task planning tools for professionals, the system should emphasize low cognitive load by closely associating planning interactions with actions of the robots in the physical world.

4.3 V.Ra Ecosystem Workflow

4.3.1 Choice of Approach

We want to develop an ecology where robots and IoTs are complementary to each other's role. As illustrated in Figure 4.2, Our workflow supports users to coordinate robots and IoTs temporally and spatially to accomplish multiple tasks synergistically in our daily surroundings. We deploy our AR authoring interface to a SLAM capable mobile device which is easy to access thanks to commercial AR SDKs such as ARCore [68] and ARKit [69]. Within a mobile AR scene, users simply register IoTs with the SLAM map. By referring to the spatial distribution of the IoTs and the geometry of the environment, users then plan, preview, and iterate the robot-IoT interactions in-situ. Further, the same AR device can be employed as the the 'eye' and 'brain' of the robot to execute the authored task. Such interchangeability between an authoring interface and robot navigation module promotes an transparent knowledge transfer from the users to the robots. As the SLAM map is constructed on-the-fly, our workflow does not rely on external tracking systems or an

existing spatial map a priori, our system is therefore easy-to-install in a new environment and ready-to-use instantly.

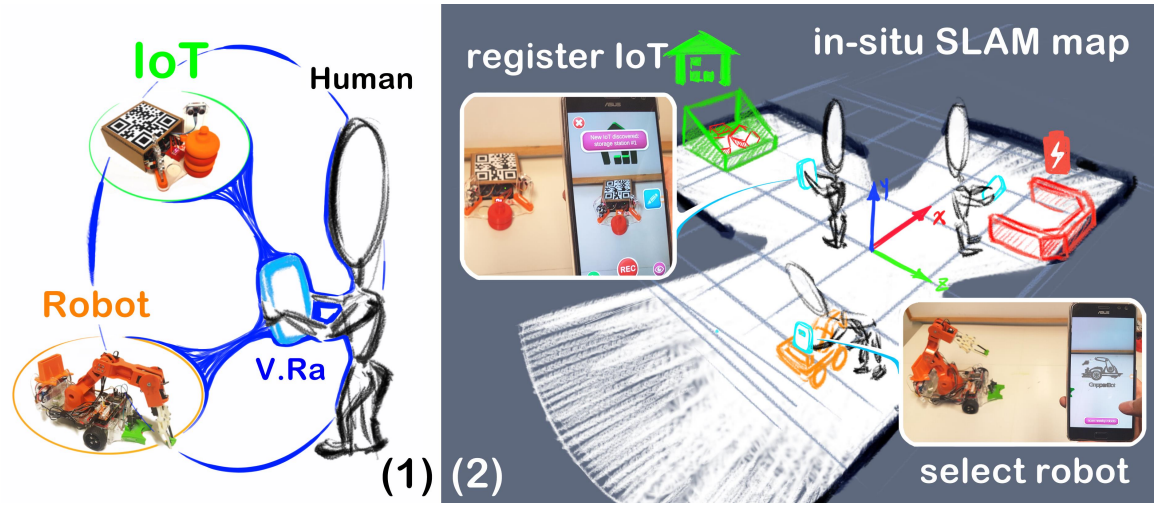


Fig. 4.2. V.Ra ecosystem design coherently connects the three key elements of robot-IoT task planning with one AR-SLAM mobile device (1), the spatial information for robot navigation and IoT interaction are stored in the on-the-fly generated SLAM map (2).

4.3.2 V.Ra System Walk-Through

As illustrated in Figure 4.1, we walk through our workflow with a typical use scenario. In a household environment, users first select a robot for the desired tasks from the available nearby ones. This allows an AR authoring interface to be specialized based on the capabilities of this particular robot. The spread IoTs can be registered into the SLAM map through a one-time QR code scanning. Users then access the embedded knowledge from the IoTs in AR view. Using our authoring interface, users formulate a group of navigation paths, IoT interactions, and other time and logic constructs to achieve the desired robot-IoT coordination. After the authoring is finished, users physically place the authoring device onto

the modular slot of the robot, and the system guides the robot to execute the tasks. Because of the transparency between the users' intents and robots' actions in the AR authoring phase, we achieve programming a robot in a WYDWRD fashion.

4.4 Authoring Interface Design

4.4.1 Task Planning Construct

To start designing the authoring interface for mobile robot task planning, we first extract the basic elements of the task. The nature of our work is robot planning for physical tasks that involves interaction with different devices at various locations. The planned task may take a long period of time to execute, and it involves logic conditions that handle unexpected situations dynamically. By referring to previous programming protocols for IoTs and robots [158, 162] and catering them to our system specifics, we develop the following ***Node***s to represent task elements and construct a task sequence.

Navigation Node ● : represents the path for the robot to travel through. It contains 3D coordinate information that can guide the robot's navigation during the Play mode.

Action Node ● : defines an action event that relates to the robot and/or the IoT device. The most common Action Node in our system is a robot-IoT interaction Node.

Time Node ● : contains information that allows the system to perform time based behaviours. For example, *keep doing this for some time,* or *wait until that happens,* etc.

Logic Node ◆ : contains a user defined check condition that allows the system to perform logic driven tasks such as *if Condition A then Action B*.

These *Nodes* are the basic abstractions that form any user authored task in V.Ra, namely, a construct array in our system, called **TaskSequence**. User can add new Nodes or manipulate the existing Nodes in the TaskSequence. When executing in the Play mode, the system guides the robot to run through each Node sequentially thus accomplish the authored task.

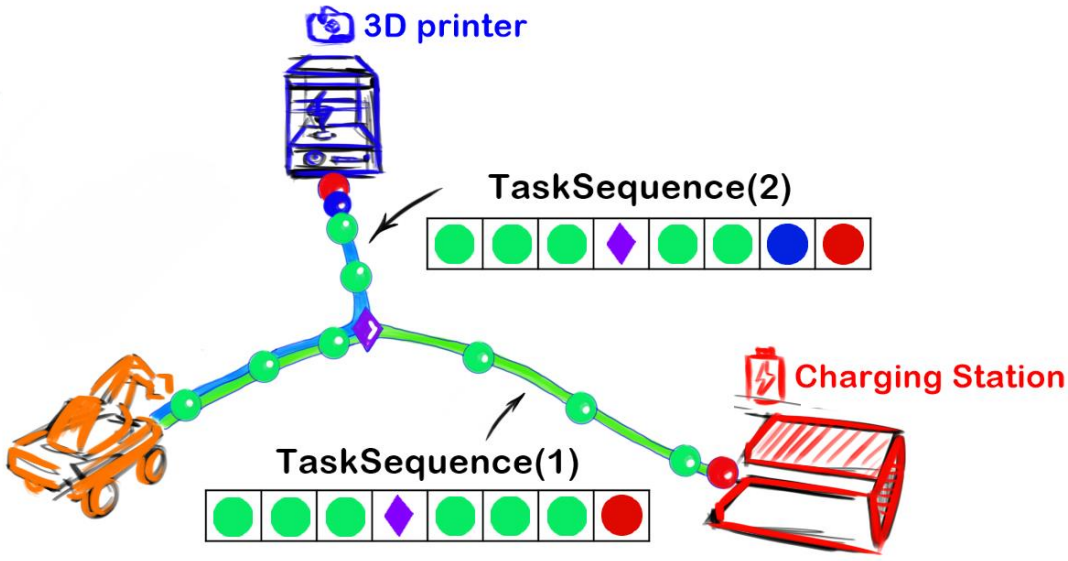


Fig. 4.3. User authored tasks are represented by TaskSequence in V.Ra system, and they are formed by four types of Nodes. Logic driven event is represented by multiple TaskSequences.

The logic driven events are realized by multiple TaskSequences with each one representing one task line. Figure 4.3 illustrates a logic event with its corresponding TaskSequences. The robot checks the condition at the Logic Node and decides which path to take. If the battery is low, it will continue on TaskSequence(1) and go to the Charging Station; otherwise it will proceed on TaskSequence(2) and go pick up the 3D printed part when it is finished. Note that the *wait...until* function is realized by the blue Time Node.

4.4.2 V.Ra Interface and Interaction

The interface design of V.Ra system is shown in Figure 4.4. We exercise simple and clean style for the UI design, while maintaining the accessibility of the primary features and functions. Users can create TaskSequence and preview it in the AR view and also in the ***EventLine***, which is an interactive abstraction of the task.

Fig. 4.4. Main interface design of V.Ra system (top). An icon reference list for interactive functions in the system (bottom).

To start a new task planning after selecting a robot, a user first defines the robot path with *AddPath* function by spatially walking around or hand-drawing on the AR view. When interacting with an IoT device, a user first uses the *IoTScan* function to scan its QR code and register it into the AR scene, then touches on its function list to add new robot-IoT interactions. User can preview the authored task by dragging the yellow handlebar on the EventLine, he can *Insert* new IoT function, time delay, or alternative TaskSequence at the

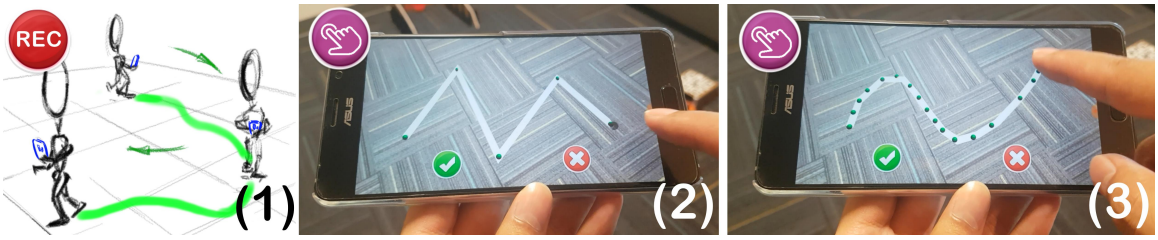


Fig. 4.5. Authoring Navigation Node with (1) spatial movement, (2) hand-drawn segment line, and (3) hand-drawn curve.

position of his choice. He can also partially loop, mirror, or delete the selected EventLine using the *Edit* function. The user has the option to create periodic robot tasks (i.e. repeat everyday) using the *Repeat* function. When the user is happy with the planned task and ready to execute, he can activate the *Play Mode* and place the mobile device onto the robot. The robot then starts the execution of the planned tasks by sequentially running all the Nodes in the TaskSequence.

4.4.3 Basic task generation

Add robot path. Navigation Nodes are the majority Nodes that form the TaskSequence in our system as it defines the path for the robot to navigate in the environment. There are two ways to add navigation nodes: 1) record spatial movement (REC), or 2) hand-draw the path on the screen, as illustrated in Figure 4.5. The hand-drawn method are suitable for path planning in a smaller area, while the REC is designed for conveniently creating large room-level navigation paths through embodied spatial movement. Each created path is broken into a series of Navigation Nodes and are added to the end of the TaskSequence. After a navigation node is added, a green path will be displayed in the AR scene giving the user active visual feedback.

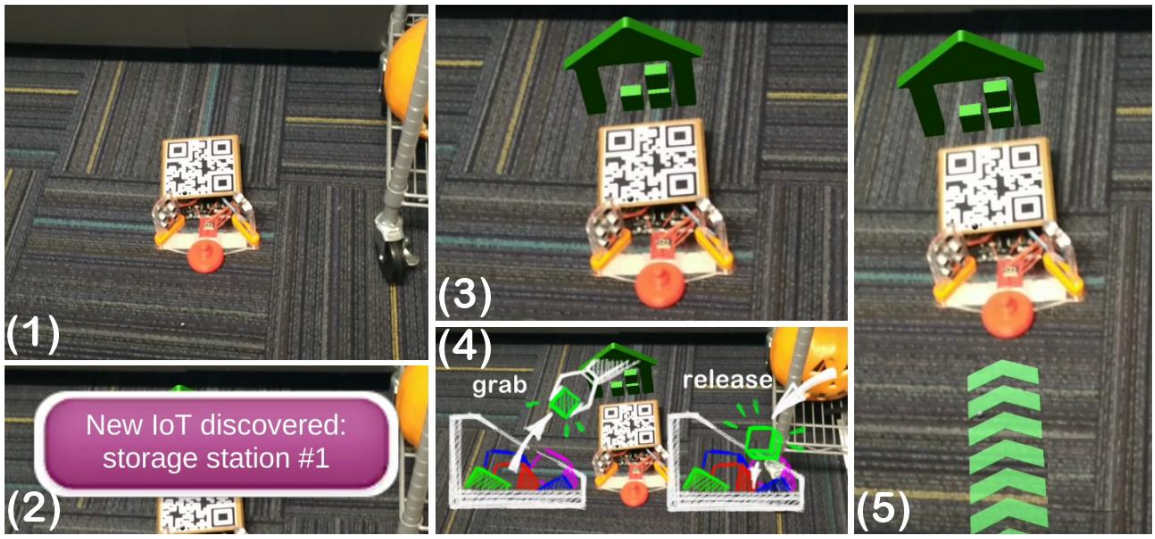


Fig. 4.6. The process to add IoT interaction Node. (1) First scan its QR code to (2) register it into the AR scene. (3) Then touch on its virtual icon (4) to access the function list. (5) When finished, a green arrow path will appear for visual confirmation.

Add IoT interaction. Robot-IoT interaction encompasses the majority of the Action Node in the system. Other Action Nodes include IoT-only and robot-only functions. To add a new robot-IoT interaction Node, the user first needs to register the IoT device into the AR scene, which is achieved through a one-time scan of the IoT's QR code (Figure 4.6 (1-2)). This not only brings an interactive 3D virtual model into the AR scene (Figure 4.6 (3)), but also imports semantic information into the system, like IP address and interaction protocol. After the IoT registration, user can press its virtual icon to access its function list and select to add an Action Node (Figure 4.6 (4)), to the end of the TaskSequence. When a robot-IoT interaction Action Node is added, a green arrow path appears, pointing towards the IoT device as a visual indicator (Figure 4.6 (5)). Other types of Action Nodes can be added using the *Insert* function, which will be described later.

EventLine task visualization. While the AR view is good for spatial task visualization, it is constrained by the view of the display, which makes it difficult for user to perform global

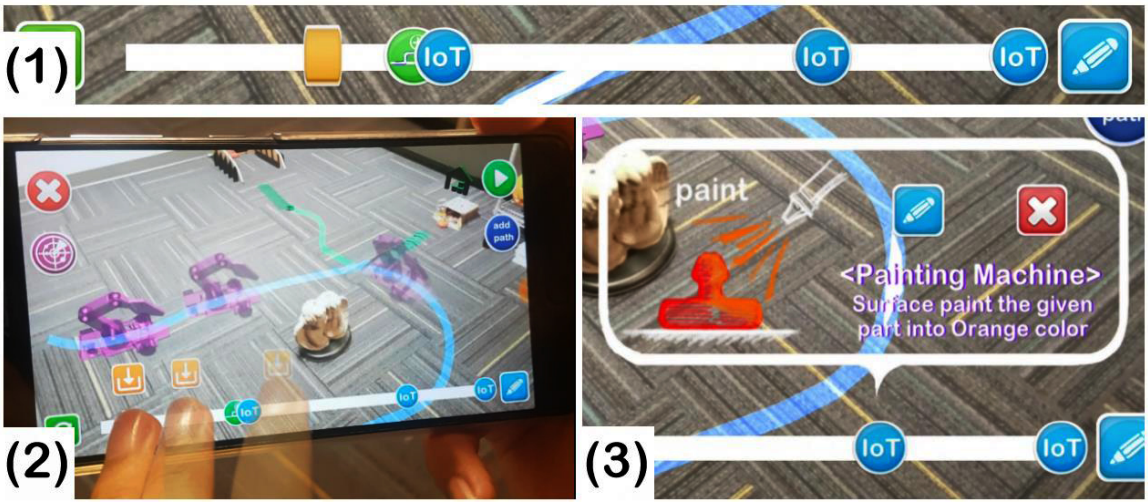


Fig. 4.7. (1) EventLine represents the task in a linear and compact format. (2) User can drag the handlebar to preview with a virtual robot. (3) User can tap on the icon to review its detailed information, and to edit or delete it.

monitoring and manipulation of the entire task, especially when the task is authored in a large cross-room environment. To compensate for this on a handheld device, we introduce an abstract visualization of the task, called ***EventLine***. The design of EventLine is inspired by the timeline concept used commonly in the animation industry. The difference being that, in our case, the task is governed by events, such as robot navigation and IoT interaction. As is illustrated in Figure 4.7 (1), the EventLine has all the non-navigation Nodes shown on it as icons, and the user can tap on it to view its details, edit it or delete it (Figure 4.7 (3)). By dragging the handle on the EventLine, the user can preview the task with a virtual robot (Figure 4.7 (2)). This is designed to help users simulate the robot path execution to avoid unexpected errors. When multiple task lines exist, only the currently selected task line will show its EventLine on the screen, to keep the screen view clean. User can switch the selected task line by tapping on it in the AR view. The selected task line will be highlighted with the white indicator flowing through it.

4.4.4 Task manipulation

The use of EventLine not only helps one to visualize the task in a linear abstract form, it also provides users with an editing tool to access the task details visually and manipulate them.

Insert. By dragging the handle, users can insert new Nodes into the designated position in the TaskSequence, which is illustrated by the position of the virtual robot (Figure 4.8 (1)). These Nodes are 1) non-robotic IoT Action Nodes, 2) Time Nodes, and 3) Logic Nodes. To insert an IoT function, the system provides the user with a list of all the IoT devices that are connected to the system (Figure 4.8 (2)). Users then select from the list, access the function of that IoT, and insert it into the TaskSequence. To insert a Time Node, users either set a fixed wait time (Figure 4.8 (3)), or define a *wait...until* condition that is triggered by the IoT working status or sensing values. User can repeat the process and create composite AND/OR boolean conditions. In terms of the Logic Node, upon selecting, an alternative TaskSequence will be created and user will be asked to define the trigger condition, which is the same condition definer interface for the Time Node (Figure 4.8 (4)). The newly created TaskSequence has all the Nodes prior to the insert point copied from the original TaskSequence. This allows users to define new task line that branches from the Logic Node position (Figure 4.8 (5)). When executing a task with multiple TaskSequences, the system will run from the default TaskSequence (the first created TaskSequence) and decides which TaskSequence to continue at an Logic Node, based on the condition check.

Edit. By utilizing the EventLine, V.Ra allows user to edit their authored task by looping, mirroring, or deleting part of the selected EventLine. The copy and mirror functions are designed to increase the authoring efficiency for scenarios like *repeat this floor sweeping path 10 times* (loop), or *go back to where you came from* (mirror). When accessing the Edit mode, two interactive markers will appear on the EventLine with the middle part highlighted.

Fig. 4.8. The Insert function. (1) User can drag the EventLine handlebar and choose a location to insert (2) non-robotic IoT function Action Node, (3) Time Node, or (5) Logic Node that represents logic driven event with an alternative task line. (4) It's trigger condition is defined from the working and sensing status of the connected devices.

Users can drag the markers to define the edit range, and the corresponding part in the AR view will also be highlighted (Figure 4.9).

4.4.5 Post-play features

V.Ra's system interaction does not end at the Play mode. Guided by DG3, we want to keep the user in the loop during the entire process. Even during and after the robot execution. As illustrated in Figure 4.10 (1), our system allows users to live monitor the task execution using an external smartphone, by video streaming via the front camera of the authoring device (the rear camera is used for SLAM tracking). User can stop the whole operation via the STOP button if he notices something goes wrong or simply changes his mind. During the play mode, our system will automatically record the video feed from the front camera and generate an event-icon-embedded video log and stores inside the device (Figure 4.10 (2)). User can later access this video log to review what have happened during the Play mode, for process analysis and debugging.

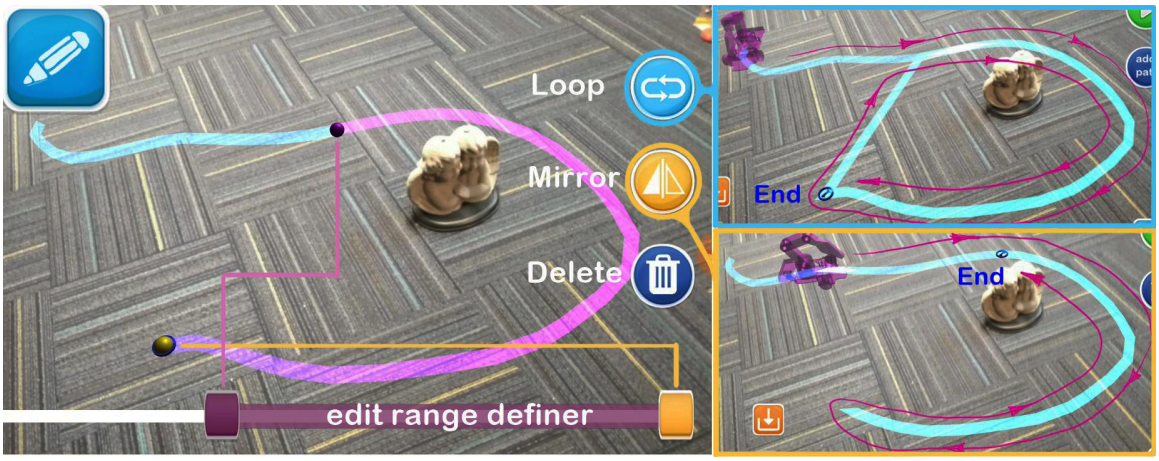


Fig. 4.9. The Edit function for partially loop, mirror, or delete the authored task.

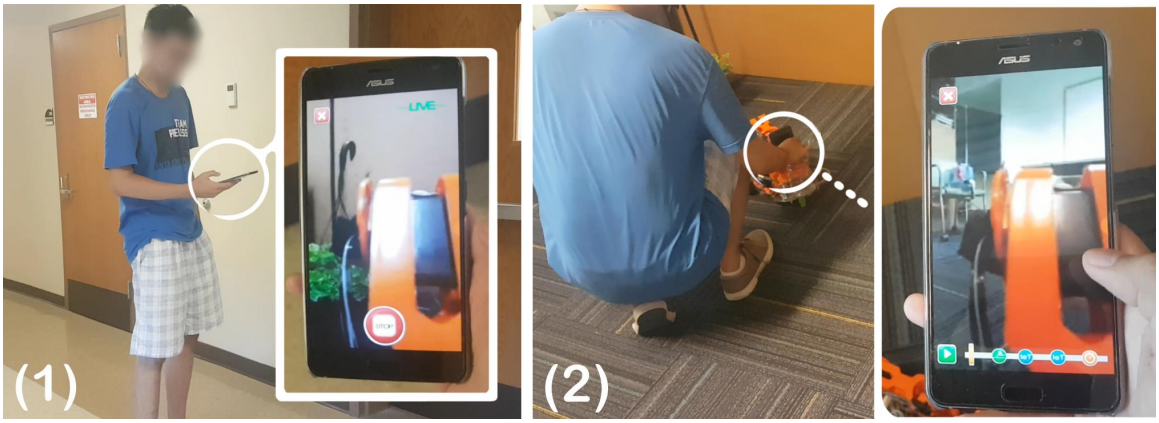


Fig. 4.10. Post-play features of V.Ra system. (1) User can monitor the robot execution during its Play mode using an external smartphone. (2) Our system also creates video log that records the robot's execution.

4.5 Implementation

4.5.1 Software platform

Our software interface is implemented as an application that runs on ASUS Zenfone AR mobile device. The AR SLAM feature is achieved using Google's software SDK - Tango Core, and the application is built with Unity3D engine. The live task monitor feature is implemented with the WebRTC video stream service. It is noted that Tango Core relies a built-in depth camera to produce point cloud based user interaction. We chose this device due to the technology availability at the time of our initial development. However, our system is not limited to depth camera based Tango device. V.Ra is fully compatible with the latest AR-SLAM platforms which use RGB cameras of the regular smart phones (e.g., ARCore [68], ARKit [69]) for SLAM an plane detection.

Fig. 4.11. Prototyped robots and IoTs in V.Ra system. (1) TowerBot (2) GripperBot (3) SweeperBot (4) WaterBot (5) Charging Station (6) Painting Machine (7) 3D printer (8) Sorting Box (9) Storage Station (10) Water Station

4.5.2 Hardware prototyping

To showcase the concept of V.Ra system, we prototyped four robots (Figure 4.11 (1-4)) and six different kinds of IoTs (Figure 4.11 (6-10)) for our use case demonstrations. All the robots and IoTs are equipped with wifi communication capability using UDP protocol, which is implemented using ESP8266 and Arduino Mega microcontroller. The motor functions of some robots and IoTs are provided by the HerkuleX servo and Arduino Braccio robot arm. All of our robots and IoTs are designed to prove the concept of our proposed human-robot-IoT task authoring ecosystem, and therefore they are mockup prototypes with fairly low fidelity.

4.5.3 Robot navigation and IoT interaction

During the play mode, the authoring device instructs the robot to perform navigation and interaction activities. To navigate the robot along a user-defined path, the device

constantly checks its current position and orientation in the SLAM map coordinate system, and compares with the target Node's coordinate information to guide the robot's movement. In other words, the SLAM device is the 'eyes' for the robot to navigate. To interact with an IoT, the robot first docks into the interaction position of the IoT by going through a short docking path embedded within the interaction Node. All the IoTs have similar docking target which is a red rounded object. At the end of the docking path, the robot reaches close enough to the docking target and it can finalize the docking process using the front color detection camera (Pixy CMUcam5). Once the robot is docked with an IoT device, precise manipulation (like grabbing an object from the Storage Station) can be ensured and the interaction is proceeded via a three-way communication among the authoring device, robots and IoTs. For example, to grab from the storage station, after successful docking, the robot first asks the Storage Station about how many objects are currently stacking on it, and based on the answer it grabs at different positions and then completes this Robot-IoT interaction.

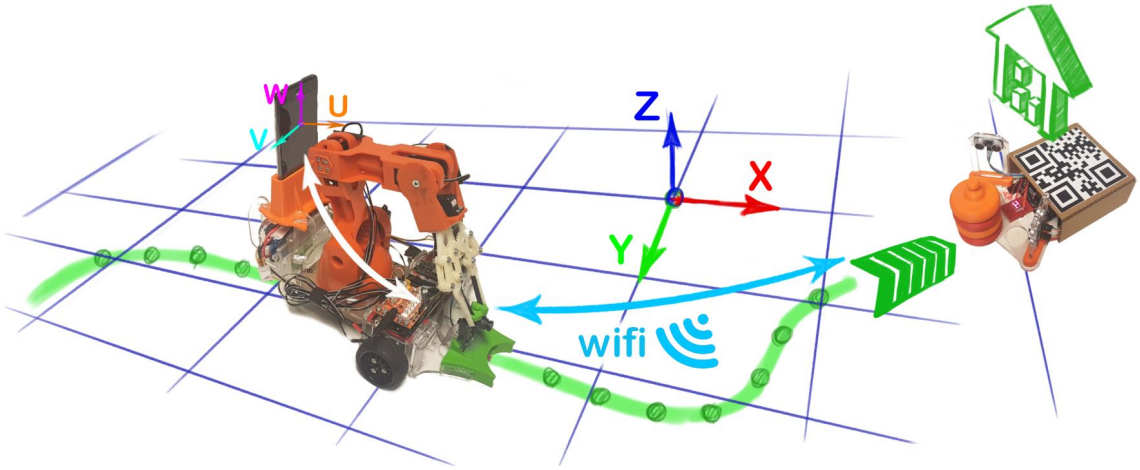


Fig. 4.12. Communication among the robot, IoT, and the authoring device during navigation and robot-IoT interaction.

4.6 Use Cases

In this section, we demonstrate three different use cases that showcase the potential use of V.Ra system in household scenarios. For better visualization of the use cases, please refer to our demo video.

4.6.1 Case 1: SweeperBot for smart floor cleaning

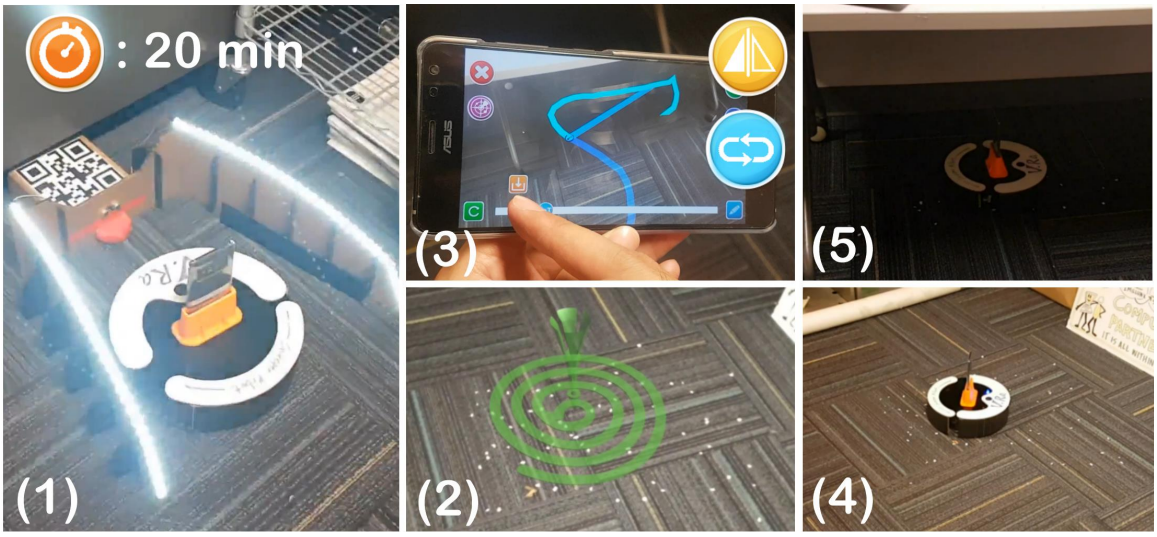


Fig. 4.13. Use case 1. (1) Battery charging for 20 minute. (2) Using the spotSweeping feature to author floor cleaning. (3) Using the Mirror and Loop feature to author repeated sweeping path under the table. (4) SweeperBot cleaning the floor. (5) Robust navigation under the table with poor lighting condition.

Our first use case features SweeperBot as a mock-up representation of the commercial sweeping robots, for user defined smart floor sweeping. As opposed to commercial products that try to survey the entire room with very little user interaction, our system allows user to pinpoint the area that needs cleaning, thus greatly increase the cleaning efficiency. In this demo, the user programs the SweeperBot to clean the paper debris on the floor and perform

an intensive sweeping under the table. Before the user starts, he notices the power LED on the SweeperBot blinking, indicating a low battery status. While trying to finish the task authoring without any delay, the user programs the robot to go into the Charging Station to charge for 20 mins using the *Timer* delay function (Figure 4.13 (1)), then pinpoints the area for cleaning using the *SpotSweeping* robot function (Figure 4.13 (2)). The user also authors the curved sweeping route under the table and uses *Mirror* and *Loop* functions to repeatedly clean that area. This use case demonstrates how V.Ra system can increase the household job efficiency by providing smart human instructions. It also showcases the robustness of the system's navigation capability, that the robot is able to successfully cruise under the table with poor lighting conditions (Figure 4.13 (5)).

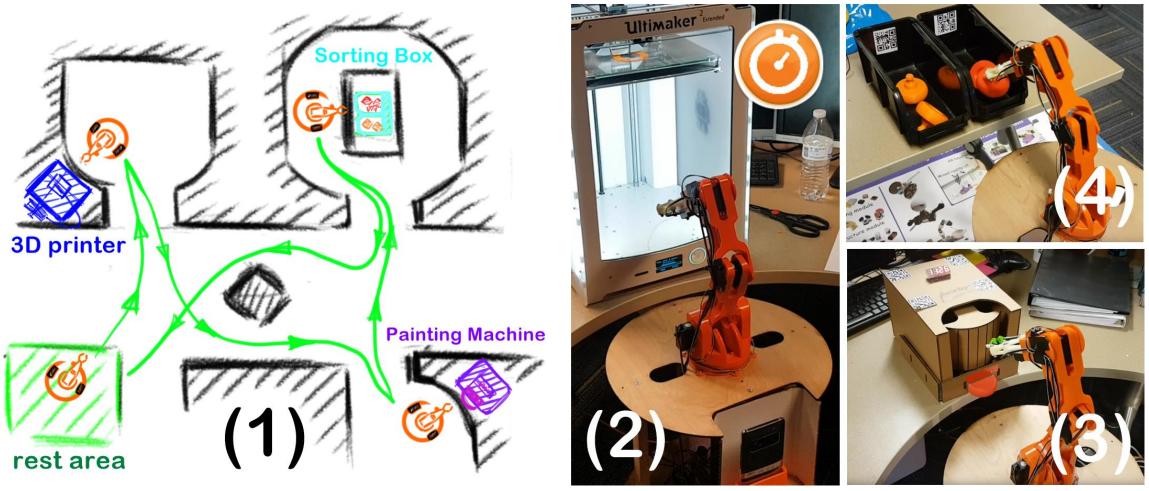


Fig. 4.14. Use case 2. (1) Navigation in a large clustered room. (2) Waiting for the 3D printer to finish its current printing job, and then pick it up. (3) Surface coat the part in the Painting Machine. (4) Placing the part inside the Sorting box.

4.6.2 Case 2: TowerBot for automated fabrication

Our second use case features TowerBot in a large clustered room (Figure 4.14 (1)), helping makers with automated fabrication process. In this demo, the user wants to fabricate a few parts through the following process. Each part is 3D printed, surface coated in the Painting Machine and then placed into the Sorting Box. The part needs to be printed one by one and each printing takes 3 hours to finish. To automate the above task and fabricate three parts, he first uses a triggered *Time* delay for the robot to wait until the 3D printer finishes printing the current part, then picks it up (Figure 4.14 (2)). The user then authors the 3D printer to start printing another part. After that, he plans the path for the TowerBot to navigate through the clustered room and interact with the Painting Machine (Figure 4.14 (3)) and the Sorting Box (Figure 4.14 (4)), then comes back to the rest area to charge the battery. Before executing, the user authors a *Repeat* function upon the entire task for three time with an interval of 1 hour for battery charging. This use case demonstrates V.Ra system's robust navigation capability in a large cluster room environment with a human-scale robot. It also demonstrates the real life application of logic triggered timer and periodically repeated task planning.

4.6.3 Case 3: WaterBot for daily plant watering

Our third use case features WaterBot for automatic daily watering of home plants (Figure 4.15). The user is leaving for a long vacation and he wants to make sure that his two favorite plants (Flower and Grass) are well taken care of. The Flower needs regular watering everyday, while the Grass needs much less water, and over watering would be harmful. To cater to the two plants with different watering frequency needs, the user first authors the WaterBot to water the flower and then comes back to the Charging Station, then repeat the task everyday. On the way back to the Charging Station, he *Insert*(s) an alternate

task line which is triggered by the moisture sensor planted in the Grass, to water it when needed. He also *Insert*(s) another alternate task line triggered by the WaterBot water level sensor, to go to the Watering Station and refill its tank when it's running out of water. This use case demonstrates our system's ability to author flexible logic driven events and shows the potential for home environment automatic plant and pet care taking.

4.7 Preliminary User Study

To evaluate the navigation and overall usability of our system, we invited 10 users (7 male) from various backgrounds to our two-session preliminary user study. None of them had prior experiences with our system and their age ranged from 22 to 30. The two-session

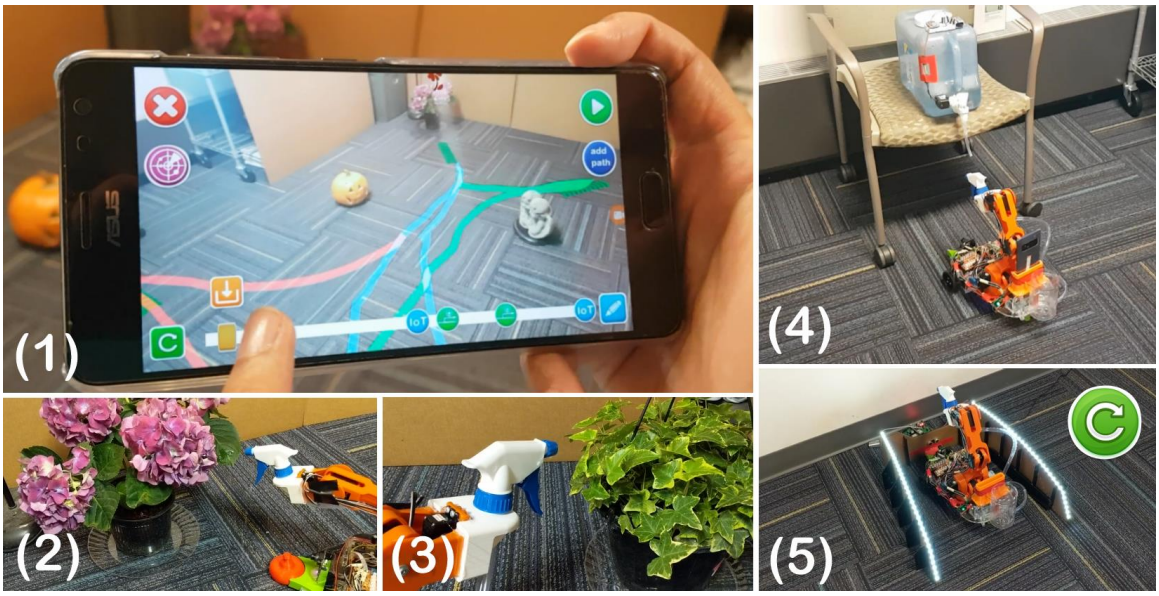


Fig. 4.15. Use case 3. (1) Use authors multiple task lines to handle different scenarios and set the task to repeat on a daily basis. (2) The Flower needs watering every day, while (3) the Grass only needs water indicated by the moisture sensor. (4) The robot goes to refill the tank when it's running out of water. (5) And returns to the Charging Station after.

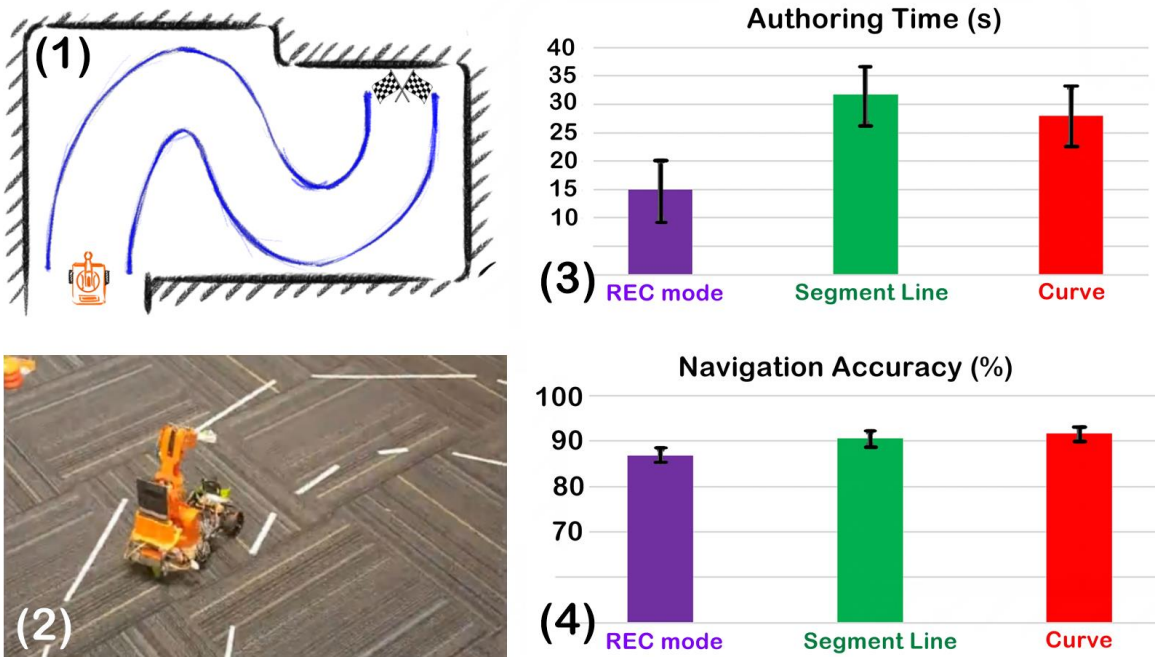


Fig. 4.16. (1) Illustration of the ground setup for session 1. (2) User authors navigation paths for the robots to travel within the track. Result of session 1 are shown as (3) authoring time, and (4) navigation accuracy.

study was conducted in a 6.5x5 meter room using the GripperBot, and the entire process was video recorded. Each user was given a 15 min tutorial before proceeding to the task in session 1. After each session, the user was given a survey to answer subjective and objective Likert-type questions. Each Likert-type item is graded by users from 1 to 5, on the usefulness of the feature and the level of agreement.

4.7.1 Session 1: Navigation Accuracy Evaluation

Using a SLAM embedded AR interface, our system is capable of fast and accurate in-situ navigation authoring, which is one of the system's core features. The first session of the study is designed to evaluate this.

Procedure. The setup for study session 1 is illustrated in Figure 4.16 (1). We have drawn an 'S' shaped track on the floor and asked the participants to author navigation path(s) for the robot to go through it while trying to maintain within the track. The participants were asked to use all three methods (spatial RECord movement, hand-draw segment, hand-draw curve) to author the path at a normal speed. The navigation accuracy was measured visually as *distance the robot traveled within the track / the overall length of the track*. The width of the track is 40 cm, which is only 40% wider than the robot. If any part of the robot goes out of the track, the condition was recorded as not met.

Result and Discussion. The result of the task in session 1 is shown in Figure 4.16 (3-4). All users were able to complete the authoring within 36 seconds using any of the three methods. All three methods were able to achieve high navigation accuracy with stable performance (low SD). Among the three methods, REC mode is the fastest to author navigation path, while still maintaining a good accuracy. It is noted that the accuracy of the off-the-shell SLAM tracking (Tango Core) is within a few mm. However, the real navigation accuracy applied on a physical robot is affected highly by the driving mechanism of the robot itself. Therefore the purpose of this study is to develop a qualitative preliminary accuracy test for our system. For example, if a user plans a room-scale path to go around the obstacles, how well will it actually turn out? In fact, even a perfect trajectory (in the middle of the track) still only has 90%-95% accuracy according to the above criteria.

Most participants stated that their most handy method to author robot path is through using a Segment Line. This is because the segment method is the easiest means for a small room-scale area. "*The segment method is my favorite, I can simply tap on the screen and create a path without needing to move my body (P2).*" "*I like the segment method, it is fast and intuitive, just touching a few times on the screen (P8).*" The second favorite is the curve method, especially at a corner region. While its feature is beneficial to create curved trajectory with ease, some users found it hard to master. "*I think drawing on the screen to*

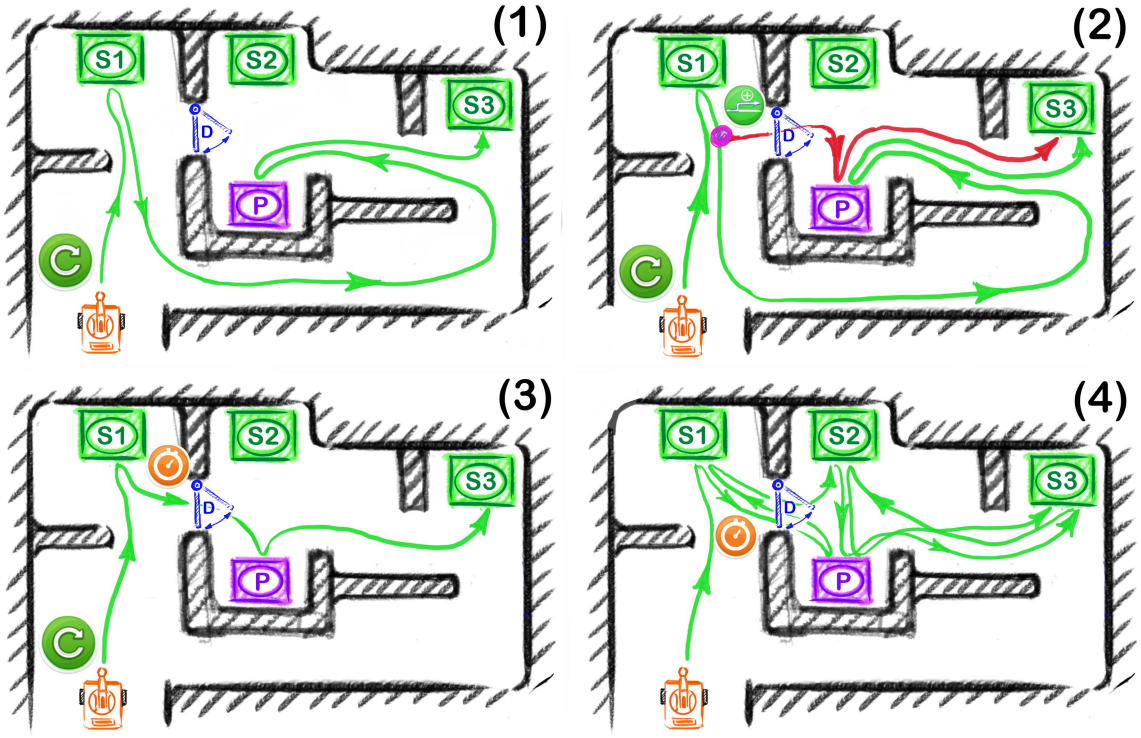


Fig. 4.17. Results of study session 2 on a comprehensive task with different approaches from the users.

create a path is interesting, even though it needs steady hands (P7)." It is noted that not many users appreciate the REC mode for this task, due to the relatively small area of the study room, they preferred to avoid walking in the cluttered scene setup. However, most users admitted that REC mode is more suitable for larger area cross-room navigation authoring, where looking at and tapping on the screen along the way would be very inconvenient.

4.7.2 Session 2: System Usability Evaluation

The second part of the study is to evaluate the overall usability of our system by asking the participants to complete a comprehensive task.

Procedure. The setup for this session of the study is illustrated in Figure 4.17 (2). Where the Storage Station 1 (S1) is stacked with orange objects, while S2 and S3 are empty. The Painting Machine (P) can paint one object at a time, into red color, using 3 minutes. There is a Doorway (D) in the setup that periodically opens and closes. The task is to stack two red objects onto S3. To achieve this, participants need to author the robot to navigate in the scene, first get the orange object from S1, paint it to red in P, then place it onto S3. Each participant was given 30 mins for this task and they were encouraged to explore as many different approaches as they like to complete the task. Participants were given full access to all the functions of the system, including the post-play features, to thoroughly experience the usability of V.Ra system.

Results and Discussion. All participants were able to successfully complete the tasks. The average authoring time for each approach is 2 min 16 s. Figure 4.17 (1) illustrate the most commonly used approach, which have been tried by 8 out of 10 participants. Though it is the most straightforward method, it is not the most efficient authoring approach in terms of robot travel distance and execution time for this task setup. Many participants were interested in trying different approaches after the basic solution. Figure 4.17 (2-4) illustrates some of the more exciting task authoring. P4 has created alternative TaskSequence that allows the robot to take the shortcut if the Doorway is open, otherwise take the detour (Figure 4.17 (2)). While P7 determines to take the shortcut no matter what, if the Doorway is closed, the user authored the robot to wait at the entrance until it opens (Figure 4.17 (3)). Since this task requires two objects to be stacked onto S3, most participant use *Loop* and/or *Repeat* function to automate the authoring for the second object. However, P11 had a different and interesting approach. While the Painting Machine is processing the first object, instead of waiting idly, the user programed the robot to go take the second object and put it onto S2 temporarily (Figure 4.17 (4)). It is worth mentioning that this method takes less time for robot to execute, but it did take more time for the user to author (5 min 27 s).

4.7.3 Observation and feedback: meeting the design goals

The Likert-type result from the two-session preliminary user study is shown in Figure 4.18. From our observation, participants could quickly learn the features and interactions of V.Ra, the 15 min tutorial was more than enough to shake off the cold feet for a novice user and they are generally excited to try V.Ra on their own. The decision of using an app-based smartphone device (Q1) greatly lowers the cognitive load for a novice user because they feel they are already familiar with the system. "*I like the idea of using my own smartphone to control the robots and IoTs, makes me feel more comfortable about using the technology (P12).*" The clean style of our UI design (Q2) and the nature of physical spatial authoring (Q4) also helps users boost up a quick start by creating a basic robot task within a minute. "*It's very easy to plan a task, just walk around to the device and click a few buttons (P3).*" This indicated that our prototype system is accessible and ready-to-use for a new user, with a fairly low skill floor to get started, which meets our DG1 and DG4.

Participants were generally receptive to the features and functions embedded within V.Ra. some of the task manipulation functions were highly appreciated by the participants. "*I really like the edit function, it's so simple and effective to create repetitive robot task. (P9)*" On the other hand, the *Insert* function received mixed feedback from the user. While most users acknowledged that the feature of time and logic based events has the potential to increase the level of task complexity and real-life usefulness (Q7) "*I think the Insert-alternative-task function is very useful to create real-life comprehensive robotic jobs (P7)*". Two users (P4, P6) felt that the interaction flow of authoring the logic as well as the capability of the programming can be further improved. "*It took me some moment to figure out how to make an if...else task, I think the UI and interaction about this part can use some more work (P6).*" Overall, the participants agreed that the features and function of V.Ra are well integrated (Q6) and the system has a high skill ceiling that allows users to evolve through iteration and produce complex robot task authoring, therefore meeting our DG3 and DG4.

Survey responses were positive about the visual feedback of the planned task provided by our system. The AR view of planned robot path with IoT interaction was highly appreciated by most users (Q5), "*I think it's a really cool idea to use augmented reality to visualize the robot task in 3D environment, helps me to simulate what's going to happen and remove my doubt (P2).*" the same appreciation was received for other task visualization features like the EventLine (Q8) and post-play feature (Q9). "*My favorite thing about V.Ra is that I can know what's going on through the entire process, even after the programming (P11).*" We believe these feedbacks have confirmed that our system has engaged the user-in-the-loop during the robot-IoT task planning lifecycle (DG2 and DG3).

4.8 Limitation and Future Work

SLAM tracking. The current system relies solely on SLAM to navigate at room-scale level during the execution. Though fairly robust, the tracking can be lost when the camera view lacks sufficient features for a short period of time, i.e. facing towards a white wall for

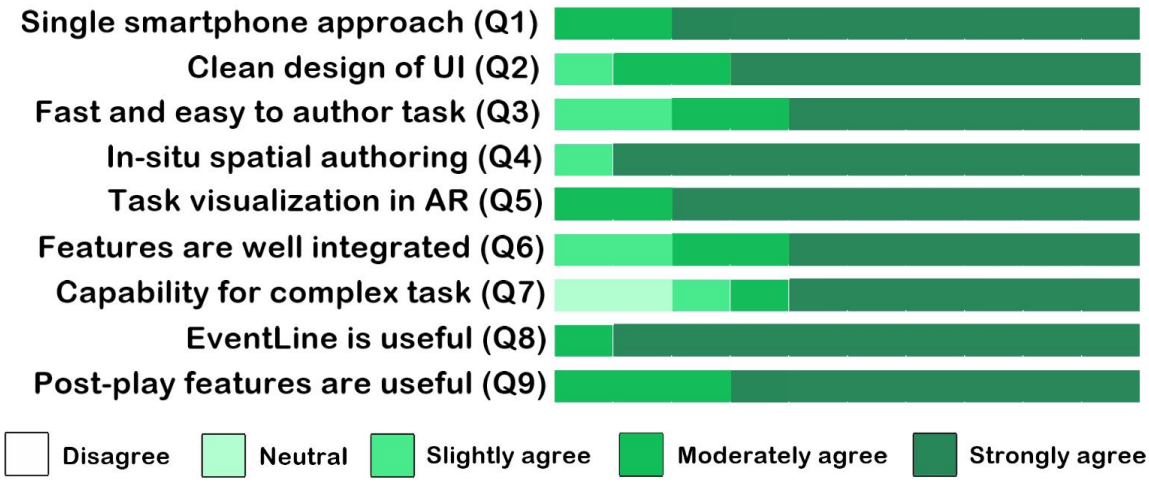


Fig. 4.18. Likert-type result after the two-session study.

a few seconds. Current system has no way of recovery in the event of lost tracking, and is therefore handicapped. To deal with this issue, a future system can embed a lost-tracking response protocol that allows the robot to automatically restore the tracking.

Logic interaction. Current system allows users to create logic driven events based on the boolean value from the robot's or IoT's working status and sensing value. This is designed to lower the user's interaction cognitive load, but also limits the level of task complexity. We believe this limitation is partly due to the mobile platform, which is better suited for touch-based toggle interaction. Future endeavors should explore other interaction modalities, with different platforms like head-mounted ones, to achieve high level of complex programming with intuitive interactions.

Robotic capability. The robots demonstrated in this work are proof-of-concept prototypes and are very limited in their functional capabilities. Since successful robot programming requires the balance of human interaction with robot automation, the level of intelligence for the robot does make a difference in terms of the authoring system design. With more advanced robots, we can expect future robotic authoring systems to focus more on high-level user intent authoring, and leave the execution of middle processes to the robots to deal with.

Single device approach. In current system, we use one single mobile device for user interaction and robot execution, which is designed based on the household application scenario. Despite the advantages of this approach, we would like to point out that the nature of the current approach limits the task to only one mobile robot at a time. If multiple authors want to collaborate on developing multi-robot collaborative task authoring, each robot needs individual navigation capability and their navigation map needs to be synchronized through some kind of a mechanism.

Human-robot-IoT ecosystem. We propose an envisioned ecology in this paper and develop a proof-of-concept prototype system. Currently we are focusing on the initial design

of the system workflow and the authoring interface, yet much more needs to be explored and developed for future endeavors. For example, a cloud based data management system needs to be setup with a new protocol created for the human-robot-IoT communication. This enables better handling of more complex tasks that requires coordinated scheduling among multiple devices, through efficient distribution of available resource, and intelligent control of information flow. The expandability of the system also needs to be considered. For example, new devices and can easily be developed to be connected into the ecosystem in a plug-and-play manner.

4.9 Conclusion

This paper has presented V.Ra, a spatially situated visual programming system for household robot task planning. We have explained our design rationale and demonstrated our user-oriented system design process. We adopted a workflow approach of one single AR-SLAM device for task authoring and robot execution, and developed our authoring interface for robot-IoT in-situ visual programming. We have shown three different use cases for household applications, featuring floor cleaning chores, DIY maker fabrication, and daily plant watering. Finally, the promising results from our 2-session preliminary user study have validated the navigation robustness and the system useability, showing that the prototype system has reached our design goals. In V.Ra, humans and smartthings enhance each other's capability within the fluidly connected ecology, such that spatially oriented collaborative tasks can be operated with lightweight system requirements. We believe that V.Ra opens an inspiring perspective for researchers to reconsider human's role in the coming era of Internet-of-Robotic-Things.

5. TIME-SPACE EDITING FOR HUMAN-ROBOT COLLABORATIVE TASK AUTHORING (GHOSTAR)

This chapter is a slightly modified version of "***GhostAR: A Time-space Editor for Embodied Authoring of Human-Robot Collaborative Task with Augmented Reality***" [3] published in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* and has been reproduced here with the permission of the copyright holder.

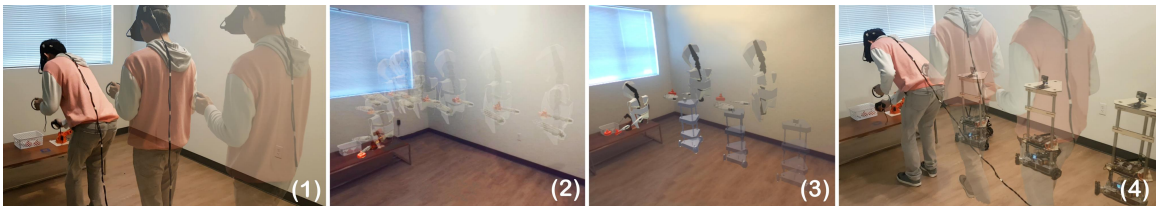


Fig. 5.1. GhostAR workflow. To author HRC tasks that achieve time-space coordination, (1) user first authors a human ghost by recording his body movement, (2) then using the ghost as a visual reference, (3) he authors collaborative robot actions. (4) When acting the task, our system's collaborative model captures the body movement as input, maps it with the authored human motion, and outputs the corresponding collaborative robot motion.

We present *GhostAR*, a time-space editor for authoring and acting Human-Robot Collaborative (HRC) tasks in-situ. Our system adopts an embodied authoring approach in Augmented Reality (AR), for spatially editing the actions and programming the robots through demonstrative role-playing. We propose a novel HRC workflow that externalizes user's authoring as demonstrative and editable AR *ghost*, allowing for spatially situated visual referencing, realistic animated simulation, and collaborative action guidance. We develop a dynamic time warping (DTW) based collaboration model which takes the real-time captured motion as inputs, maps it to the previously authored human actions, and outputs

the corresponding robot actions to achieve adaptive collaboration. We emphasize an in-situ authoring and rapid iterations of joint plans without an offline training process. Further, we demonstrate and evaluate the effectiveness of our workflow through HRC use cases and a three-session user study.

5.1 Introduction

Robotics has been extensively used to automate a large number of particular and repetitive tasks with high accuracy and throughput in manufacturing environments. The tremendous economics and social impacts projected by robotics will likely to expand in our future by infiltrating into broader fields in both commercial and consumer markets [163]. Unlike the traditional manufacturing environments, these new segments including medical, healthcare, and services, usually heavily involve human activities in the working environments. Thus, enabling robots to co-work with humans in collaborative tasks has become a major pillar of the next generation robotics technology.

A typical human-robot-collaborative (HRC) task involves generating a joint intention, planning actions, and acting cooperatively [164]. In a human-centered task, the joint intention usually aligns with humans' implicit or explicit expressions. Explicit communications such as speech and gestures have been widely studied for commanding robots [165,166]. But using these modalities may cause inefficiencies and ambiguities in spatially and temporally coordinated collaborations which require comprehensive understanding of the contexts. On the other hand, embodied demonstrations from humans directly convey the intentions to the robots. More importantly, to avoid programming robots' behaviours for the highly dynamic human robot interactions, researchers propose programming by demonstrations (PbD) to generate task and action plans for the robots [167]. Further, to safely and robustly execute the action plans in a coordinated manner, humans and robots need to timely communicate with their status, actions and intentions [168]. To this end, we primarily endeavor to explore

the design of an embodied authoring workflow to support real-time human motion inference, demonstrating examples actions to robots and creating joint plans.

The advents of mobile computing foster the evolution of authoring workflows in an in-situ and ad-hoc fashion [57, 58]. However, existing workflows primarily target at pre-defined and rigorous tasks where robots operate in isolation and interact with the environment only. To enable novice user friendly PbD in the authoring workflow, we need to support human motion capture and inference which traditionally involve a motion capture system. Since a body-suit [169] or an external-camera [170] based capture system requires heavy dependencies, demonstrations are often only captured off-line [84]. Moreover, for ad-hoc tasks, demonstrating with users' own body is preferable [167]. Recently, the emerging augmented/virtual reality (AR/VR) technologies, e.g., head-mounted AR/VR devices [153, 171], show a strong potential to enable embodied authoring [172]. Further, in HRC tasks, robot partners are desired to adapt to and coordinate with humans actions. Thus, to create a joint action plan, the counterpart motions of the robots can only be demonstrated with the humans' part as contexts. In this work, we promote a critical advantage of using AR/VR authoring, namely externalizing the users' body asynchronously [110, 112]. This way, the teachers can always view, manipulate, and edit their own recorded actions, and use them as contexts when demonstrating the counterpart motions for robots.

We promote an embodied authoring in AR for HRC tasks in this work because of the following reasons: (i) realistic visualization with contextual and spatial awareness, enabling creating, editing, and previewing the collaborative flow intuitively, (ii) easy programming with natural embodied interaction through real human demonstration via role-playing to establish time-space correspondence, (iii) supporting real-time motion inference, activity detection, and visual feedback on robots' intents when conducting the HRC. We present *GhostAR* workflow which uses AR with body-tracking to enable visual, spatial, and embodied HRC tasking authoring, as illustrated in Figure 5.1. A typical authoring session starts

when users role-play the human's actions. We render the recordings as AR ghost. Users can freely observe, edit and infer the actions and use it as reference when role-playing the robot's counterpart actions. Then, users designate correspondences between humans' own action plan and the demonstrated actions for robots. Further, *GhostAR* provides visual preview with AR simulation in-situ. When users act the HRC tasks, *GhostAR* continues to capture user's motion and use it to derive the robot's motion plan. Also, users can refer to the next-step guidance and robot's intentions with AR visual feedback. In summary, we highlight our contribution as follows.

- A **system workflow** for authoring human-robot collaborative task through AR ghost as contextual references and role-playing with natural embodied interaction.
- A **collaboration model** that achieves time-space correlation for the human-lead-robot-assist adaptive collaboration task based on dynamic time warping (DTW) algorithm.
- An **AR interface and interaction design** for human-robot ghost creation and visualization, editing and manipulation, previewing and simulation, and guidance throughout a successful collaborative action.

5.2 Design Goals

Based on the knowledge acquired from the previous works as well as our own experience on the subject of human-robot-interaction (HRI) and robot task planning and programming, we have extracted the following *Design Goal*s (DG) as guidelines of our HRC authoring system design. The practice of these *DG*s is reflected throughout our system design and implementation.

DG1: Adapting robot behavior to human. Author human-lead-robot-assist typed collaborative tasks that are initiated by human, where the robot always act adaptively to the human partner's actions.

DG2: Programming with natural interaction. Lower the barrier for users to effectively program complex HRC tasks, with natural body movement and intuitive interactions.

DG3: Authoring with contextual awareness. Provide spatial and contextual awareness that is important for Human-Robot task authoring. Both parties need to be aware of each other's position and status as well as the surrounding environment.

DG4: Visualizing with realistic simulation. Give active and accurate visual feedback about what the user has authored, to ensure efficiency and correctness of the authoring through realistic simulations.

DG5: Iterating with real-time feedback. Enable a real-time process and rapid iterations from collaborative task authoring to action, with no need for offline programming and testing.

5.3 GhostAR

5.3.1 Human-Robot Collaboration Model

It is important to first define the meaning of *collaboration* in our work as it touches a wide range of aspects, even just for tasks between humans and robots. In *GhostAR*, we essentially present a robot programming tool that controls robots' action based on its human partner's body movement. In other words, the robot collaborates with the human in the sense that it must act adaptively according to its human partner. In order to achieve this, we present a collaboration model that is dynamically generated based on user's authoring and is able to output robot action corresponding to the input human motion.

In a human-lead-robot-assist HRC task, we achieve the motion coordination by defining user's action segments first. Our system allows users to record their body movement as a ***Human Motion Clip*** (a sequence of ***Motion Frames*** with different timestamps) and to use it to create HRC tasks. Note that the authored human motion could consist several meaningful

movements and user can put them into ***Group***s to author HRC tasks correspondingly. For example, the human character in Figure 5.2 records the following motion: he walks, stops and waves his hand, then walks for some distance and waves again. The HRC task he wishes to author is to make the robot come over when he first waves, follow him and shoot videos for him as he walks, and then leave when he waves hand again. To achieve this, he needs to put the two *hand waving*s and a *walking* into three *Group*s and author the robot to behave as *come over*, *follow and shoot videos*, and *leave* correspondingly in these three *Group*s. For each *Group* of human motion, our system provides two types of collaborative tasks for user to author. They are ***Synchronize*** and ***Trigger*** tasks.

- A ***Synchronize*** task authors a robot action to take place *at the same pace* of the reference human group. In this type of HRC task, robot and human will perform their own task, but at the same speed or progress, i.e. if the human moves faster, the robot will move faster to keep up, and vice versa. This is applicable to HRC tasks such as joint object manipulation, motion following for lighting or camera shooting, and coordinated movements like hand-shaking, etc.

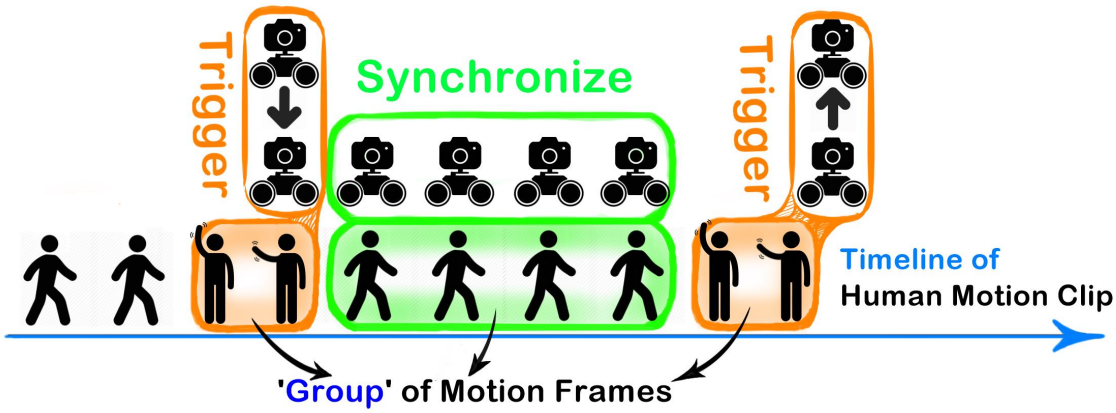


Fig. 5.2. Authoring collaborative robot actions using *Group*s.

- A ***Trigger*** task authors a robot action to take place *after* the human group. In this type, the robot starts executing its authored task right after the human has completed the reference group, i.e. human snaps his finger and the robot starts sweeping the floor. This is applicable to HRC scenarios such as sequential joint assembly, and gesture signalling, etc.

As for the example in Figure 5.2, the user will author the *come over* and *leave* robot action as *Trigger* tasks for the two *hand-wave Group*s, and author the *follow and shoot video* as a *Synchronize* task for the *walk Group*.

So far the user has been preparing the collaboration by creating the ***HRC TaskSequence***. As shown in Figure 5.3, the *HRC TaskSequence* is a list of *Group*s that represents the authored task in an accessible and manageable manner. Note that adjacent ungrouped human *Motion Frame*s will be automatically grouped as *Empty Group*s. The *HRC TaskSequence*

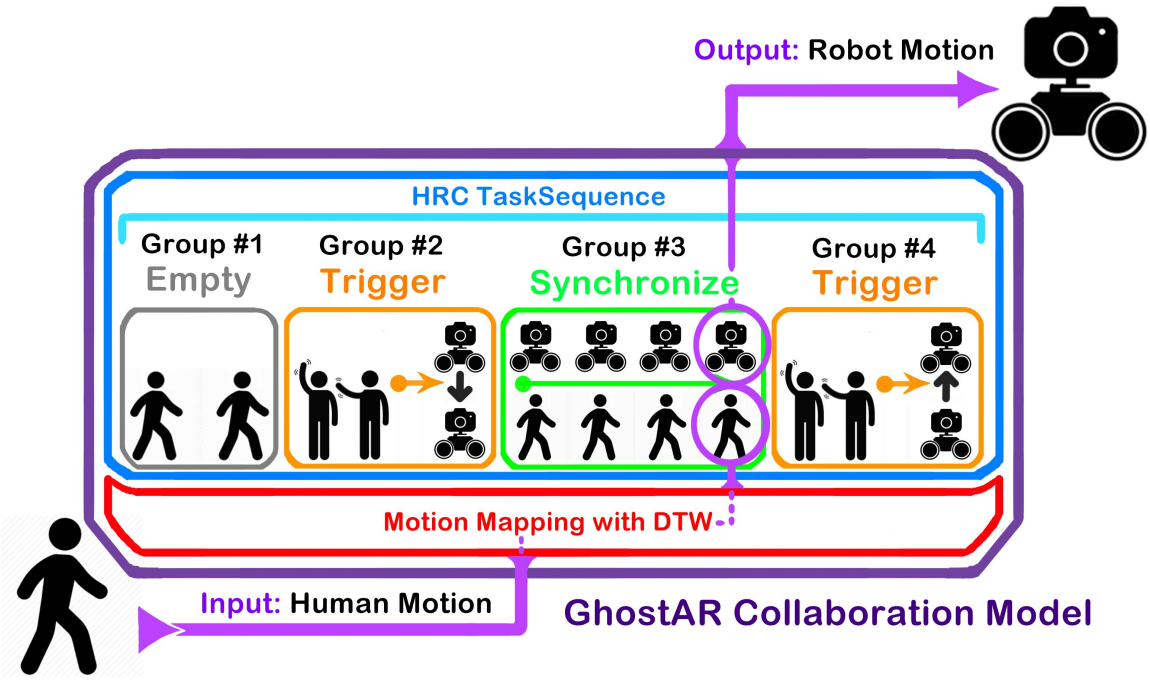


Fig. 5.3. GhostAR collaboration model.

together with the ***Motion Mapping*** module, form the collaboration model of *GhostAR*. When the HRC action is started, user needs to repeat his authored motion in the sequential order. Meanwhile our system will activate the first *Group* and start the motion mapping between the real-time captured human motion and the grouped *Human Motion Clip*. When the mapping progress indicates the current *Group* is completed, our system activates the next one and repeats this process until all *Group*s in the *HRC TaskSequence* are completed. For a *Synchronize* task, the system calculates the progress and output robot behaviour at the corresponding timestamp. For *Trigger* task (which is generally shorter), system focuses on recognizing the completion of the human movement, and then issues commencement instructions for the authored robot actions. Note that *Empty Group* will be treated the same way as a *Synchronize Group*, for proper progress monitoring and activation of the next *Group*.

5.3.2 Motion Mapping using Dynamic Time Warping

We describe how our system achieves motion mapping for both *Synchronize* and *Trigger* tasks. Essentially, in order to recognize user's status, we rely on positions of the user's head and both hands which are provided by our AR interface. We then introduce DTW to infer the user's activities using the 9 degree-of-freedom (DOF) inputs. At time t_i, the user's state is represented by an $\mathbb{R}^9$ vector:

$$v_{t_i} = [x_{t_i}^{head}, y_{t_i}^{head}, z_{t_i}^{head}, x_{t_i}^{left}, y_{t_i}^{left}, z_{t_i}^{left}, x_{t_i}^{right}, y_{t_i}^{right}, z_{t_i}^{right}]^T$$

In this manner, each *Human Motion Clip* derives an $\mathbb{R}^9$ curve as: $\boldsymbol{L}_{record} = [v_1, v_2, v_3, \cdots, v_N]$. And we denote the human motion in *Group* G_i as l_{G_i} which is a continuous segment within $\boldsymbol{L}_{record}$.

To reduce the DOF of inputs and keep the most relevant information from the raw gesture data $\boldsymbol{l}_{G_i}$, we apply principal component analysis (PCA) [173] to project this $\mathbb{R}^9$ curve onto a $\mathbb{R}^2$ plane. A projected curve $\boldsymbol{f}_{G_i}$ and a projection matrix P_{G_i} are derived as well as in Algorithm 1. For each activated *Group* G_i, the real time data $v_{t_{now}}$ is projected by P_{G_i} and then compared with the $\boldsymbol{f}_{G_i}$ to acquire the corresponding progress in G_i.

Algorithm 1 Calculate Projected Curve and Projection Matrix

1: **procedure** PCAPROJECTION($\boldsymbol{l}_{G_i}[1 \dots n]$)
2: $\quad \overline{\boldsymbol{l}_{G_i}} \leftarrow (\Sigma \boldsymbol{l}_{G_i})/(9 * n)$
3: $\quad \boldsymbol{V} \leftarrow (\boldsymbol{l}_{G_i} - \overline{\boldsymbol{l}_{G_i}})(\boldsymbol{l}_{G_i} - \overline{\boldsymbol{l}_{G_i}})^T$
4: $\quad$ Let $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ be two eigen vectors associated with the largest eigen values of $\boldsymbol{V}$.
5: $\quad$ output $\boldsymbol{P}_{G_i} \leftarrow [\boldsymbol{v_1}, \boldsymbol{v_2}]^T$
6: $\quad$ output $\boldsymbol{f}_{G_i} \leftarrow \boldsymbol{P}_{T_i} \boldsymbol{l}_{T_i}$

Trigger Task Detection. Assume that an activited *Group* G_i is a *trigger Group* and we want to determine whether the user has finished performing the human motion l_{G_i}. We first collect the motion that the user has just performed: $\boldsymbol{l}_{realtime} = [v_{t_{now}-n+1}, \cdots, v_{t_{now}-1}, v_{t_{now}}]$ where n is the length of l_{G_i}. After that we get the projected curve $\boldsymbol{f}_{realtime} = \boldsymbol{P}_{G_i} \boldsymbol{l}_{realtime}$ and compare it with $\boldsymbol{f}_{G_i}$. This method is close to a conventional human action recognition problem [174]. We use Dynamic Time Warping (DTW) algorithm [175] to calculate the similarity. DTW is an algorithm to find the alignment between two time series data. Given two time series $\boldsymbol{s} = [s_1, s_2, \cdots, s_n]$ and $\boldsymbol{t} = [t_1, t_2, \cdots, t_m]$ with length n and m, a distance matrix $\boldsymbol{D}$ is calculated using Algorithm 2. Each element $\boldsymbol{D}[i, j]$ in the distance matrix $\boldsymbol{D}$ is the distance between $\boldsymbol{s}[1 : i]$ and $\boldsymbol{t}[1 : j]$ with best alignment. And we define $\boldsymbol{D}[n, m]$ as DTW distance between $\boldsymbol{s}$ and $\boldsymbol{t}$, note as $< \boldsymbol{s}, \boldsymbol{t} >$. In our specific case, if $< \boldsymbol{f}_{realtime}, \boldsymbol{f}_{G_i} >$ reaches its global minimum, we assume that user finishes performing G_i at the current time. However, the future behaviour of the user is unavailable, so it is hard to identify when the global minimum is achieved. To this end, we use a threshold ε to conclude a global minimum given the existing behaviours of the user. Basically, if $< \boldsymbol{f}_{realtime}, \boldsymbol{f}_{G_i}, >$ reaches

Algorithm 2 Calculate DTW Distance Matrix

procedure DTWDISTANCEMATRIX($\boldsymbol{s}[1\dots n]$,$\boldsymbol{t}[1\dots m]$)
  $\boldsymbol{D} \leftarrow array[0\dots n, 0\dots m]$
  for $i \leftarrow 1, n$ **do** $\boldsymbol{D}[i,0] \leftarrow \infty$
  for $i \leftarrow 1, m$ **do** $\boldsymbol{D}[0,i] \leftarrow \infty$
  for $i \leftarrow 1, n$ **do**
    for $j \leftarrow 1, m$ **do**
      $\boldsymbol{D}[i,j] \leftarrow \|s[i]-t[j]\| + min(D[i-1,j], D[i-1,j-1], D[i,j-1])$
  return $\boldsymbol{D}$

a local minimum and this minimum value is smaller than ε, we assume that this minimum value is the global value and report to the system that G_i is triggered by the user. To adapt this threshold for different $\boldsymbol{f}_{G_i}$ with various lengths, we set $\varepsilon = a * n$ where a is a fixed coefficient.

Synchronize Task Progress Estimation. If an activited *Group* G_i is a *Synchronize* task, we need the user's progress ($0\% \sim 100\%$) in order to temporally coordinate the robots' motions. We propose to compare the the real time data $\boldsymbol{l}_{realtime} = [v_{t_{start}}, \cdots, v_{t_{now}-1}, v_{t_{now}}]$ with the subsequence of $\boldsymbol{l}_{G_i}$: $\boldsymbol{l}_{G_i}[1]$, $\boldsymbol{l}_{G_i}[1:2]$, $\cdots$, $\boldsymbol{l}_{G_i}[1:n]$, where t_{start} is the time when G_i is activated. And we derive the user's progress as n^*/n if the subsequence $\boldsymbol{l}_{G_i}[1:n^*]$ approximates $\boldsymbol{l}_{realtime}$ the most. In another word, we first project $\boldsymbol{l}_{realtime}$ to $\boldsymbol{f}_{realtime}$ using

Algorithm 3 Progress Estimation Using DTW

$\boldsymbol{d}_{old} \leftarrow array[0\dots n]$, $\boldsymbol{d}_{new} \leftarrow array[0\dots n]$
for $i \leftarrow 0, n$ **do** $\boldsymbol{d}_{old}[i,0] \leftarrow \infty$
for $i \leftarrow 0, n$ **do** $\boldsymbol{d}_{new}[i,0] \leftarrow 0$
while Synchronized Task S_i has started **do**
  if $v_{t_{now}}$ is updated **then**
    $f_{t_{now}} \leftarrow \boldsymbol{P}_{S_i} v_{t_{now}}$
    for $i \leftarrow 1, n$ **do**
      $\boldsymbol{d}_{new}[i] \leftarrow \|\boldsymbol{f}_{S_i}[i] - f_{t_{now}}\| + min(\boldsymbol{d}_{new}[i-1], \boldsymbol{d}_{old}[i-1], \boldsymbol{d}_{new}[i])$
    $n^* \leftarrow \arg\min_{1 \le i \le n}(\boldsymbol{d}_{new}[i]/\sqrt{i})$
    $\boldsymbol{d}_{old} \leftarrow \boldsymbol{d}_{new}$
    output $progress \leftarrow n^*/n$

$\boldsymbol{P}_{G_i}$ and calculate the DTW distances between $\boldsymbol{f}_{realtime}$ and the sub-sequences of $\boldsymbol{f}_{G_i}$: $\boldsymbol{f}_{G_i}[1]$, $\boldsymbol{f}_{G_i}[1:2], \cdots, \boldsymbol{f}_{G_i}[1:n]$, noted as $d_1, d_2, \cdots, d_n$. And find $n^* = \arg\min_{1 \leq i \leq n}(d_i)$. However, we note that the scale of d_i is influenced by the length of sub-sequence $\boldsymbol{f}_{G_i}[1:i]$. To eliminate this influence, a modified DTW distance $d'_i = d_i / \sqrt{i}$ $(i = 1, 2, \cdots, n)$ is introduced. Then we determine a sub-sequence $\boldsymbol{f}_{G_i}[1:n^*]$ that is best aligned with $\boldsymbol{f}_{realtime}$ while n^* is given by $n^* = \arg\min_{1 \leq i \leq n}(d'_i)$, and thus the user's progress is n^*/n. Recall the property of DTW distance matrix $\boldsymbol{D}$, $d_1, d_2, \cdots, d_n$ are actually the last row of $\boldsymbol{D}$, so in practice, we use Algorithm 3 to calculate $\boldsymbol{D}$ and n^* iteratively.

5.3.3 Embodied Authoring with Augmented Reality

Our system's interaction workflow is implemented as a state machine, where a HRC task is authored with the following five modes: ***Human Authoring Mode***, ***Robot Authoring Mode***, ***Observation Mode***, ***Preview Mode***, and ***Action Mode***. At the beginning of a new task authoring session, a user is first asked to choose the robot collaborator(s). Note that in case of simultaneously collaborating with multiple robots, each robot will share the same *Human Motion Clip* but has its own *HRC TaskSequence*. After initialization, the user will be promoted to the *Human Authoring Mode* to create the first *Human Motion Clip*. After finishing the creation, the current tasks are displayed as AR ghost for visualization and manipulation in the *Observation Mode*. The user uses the cursor to perform *Group*ing operation, and authors robot tasks for the selected *Group* in the *Robot Authoring Mode*. In the *Observation Mode*, user can choose to enter *Preview Mode* to visualize the entire HRC task simulation with AR ghost animation. Once the user is satisfied with the authored task, he/she can act out the authored HRC tasks by entering the *Action Mode*. The system utilizes the dynamically generated collaboration model to derive the corresponding robot behaviours based on user's real-time motions.

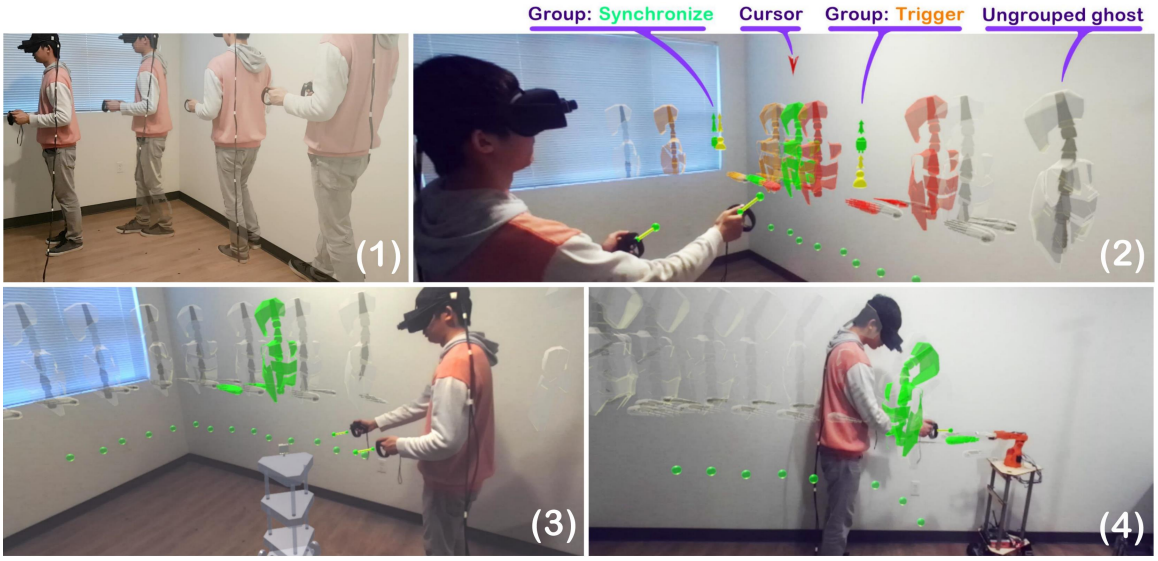


Fig. 5.4. GhostAR system interface in (1) Human Authoring Mode, (2) Observation Mode, (3) Robot Authoring Mode, and (4) Action Mode.

Human Authoring Mode. The *Human Motion Clip* is the baseline of the HRC task authoring. It contains the human motions that robot will collaborate with, as well as the movement that the user needs to repeat during the *Action Mode*. When authoring human motion, the system records the user's body motion by tracking the position and orientation of the AR headset and two hand-held controllers. Then the *Human Motion Clip* will be represented by segmented ghost avatars and displayed in the user's AR view, as illustrated in Figure 5.4-(1). The ghost avartar also plays the authored human movement repeatedly as an animation in real time scale for review. To extend the *Human Motion Clip*, the user first needs to trigger the last pose in the recorded clip and then act new human motion, which will automatically be tailed to the end of the current *Human Motion Clip*.

Robot Authoring Mode. Once the *Human Motion Clip* is created, user can pick a segment from it and generate a *Group*, then author a *Synchronize* or *Trigger* robot task for it. For each selected robot collaborator, there exists a virtual robot avatar in *GhostAR*

that mimics the behaviour of the real robot. User can control the virtual robot, with the hand-held controllers and physical movements, to facilitate the robot motion authoring. For a *Synchronize* task, the time-length of the robot clip is the same with the human group. As the user is authoring robot and progressing, the human ghost with the same timestamp will be displayed as AR reference to assist the user, as illustrated in Figure 5.4-(3). The user can pause/resume and walk around anytime during the authoring process, in order to observe and operate the robot avatar from the optimal perspective. In terms of a *Trigger* task, the user authors robot actions independently which will be placed *after* the *Trigger Group*. Once robot authoring is finished, the authored HRC task will be animating repeatedly, with both human and robot ghosts, to visualize and preview the task before the user decides to accept or redo.

Ghost Visualization and Manipulation. Our system provides in-situ authoring experience by exploiting the advantage of AR interfaces. In the *Action Mode*, the authored tasks are displayed as AR ghosts for user to preview and manipulate. The ungrouped raw human ghosts are displayed as transparent segmented snapshots while the grouped ghosts are displayed with Start/End *Motion Frames* with a uniquely assigned color and a floating 3D icon indicating its collaboration type, as illustrated in Figure 5.4-(2). Using the interactive cursor, user can edit the *Human Motion Clip* and perform operations such as *Group*ing, un*Group*ing, 'trimming', etc. If the cursor is pointing at any un*Group*ed raw human ghost, the pointed ghost *Human Frame* will be highlighted. Otherwise if the cursor is inside a *Group*, the Human-Robot task of that group will animate repeatedly until the cursor is moved outside. Note that user can also enter the Preview Mode and visualize the entire task as a continuous simulated AR ghost animation.

Action Mode. The *Action Mode* is where the user carries out the collaboration tasks. In this mode, the system captures the real-time movement of the user and maps it with the recorded *Human Motion Clip*, then issues corresponding instructions to drive the robot and

perform the collaborative task. To help user repeat his authored motion and alleviate the mental burden of memorization, the system provides numerous AR guidance to assist the user. As illustrated in Figure 5.4-(4), our system not only projects a dotted trail for the user to follow, but also plays the next-to-act *Group*'s animation to refresh user's memory. Therefore, user only needs to focus on the current task and the system is guiding him/her step-by-step. Besides, our system also provides numeric progress information for user to keep track of him/herself as well as the robot's working status.

5.4 Implementation

5.4.1 System Setup and Development

We build our see-through AR platform by attaching a stereocamera (ZED Dual 4MP Camera (720p)) in front of a VR headset (Oculus Rift). The human body motion is tracked by four external Oculus IR-LED Sensors with an effective working area of 5mx5m. Interactions used in the system are enabled by two Oculus Touch Controllers. The major part of *GhostAR* software system is developed with Unity3D engine and Robot Operating System (ROS) [176], including the AR interface and embodied interaction, motion recording and DTW calculation, etc. The authored *Human Motion Clip* and robot clips are recorded at the rate of 90Hz. It's worth to note that this prototyped AR platform still relies on external tracking and tethered computer which limits the interaction volume. However, with the newly developed mobile AR/VR technologies, e.g., Hololens [153] and Oculus Quest [171], we believe that implementing GhostAR with stand alone devices is effortless.

5.4.2 Robot Simulation and Prototyping

We have prototyped several robots, including three physical robots (GripperBot, CamBot, Armbot) and a virtual robot drone, for the purpose of use case demonstration and studying

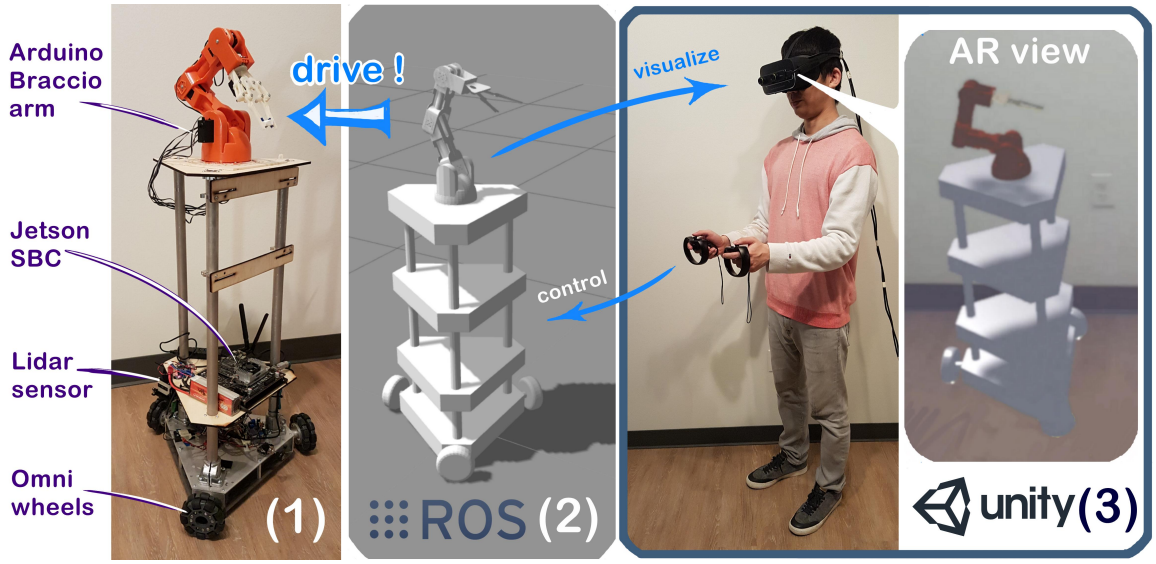


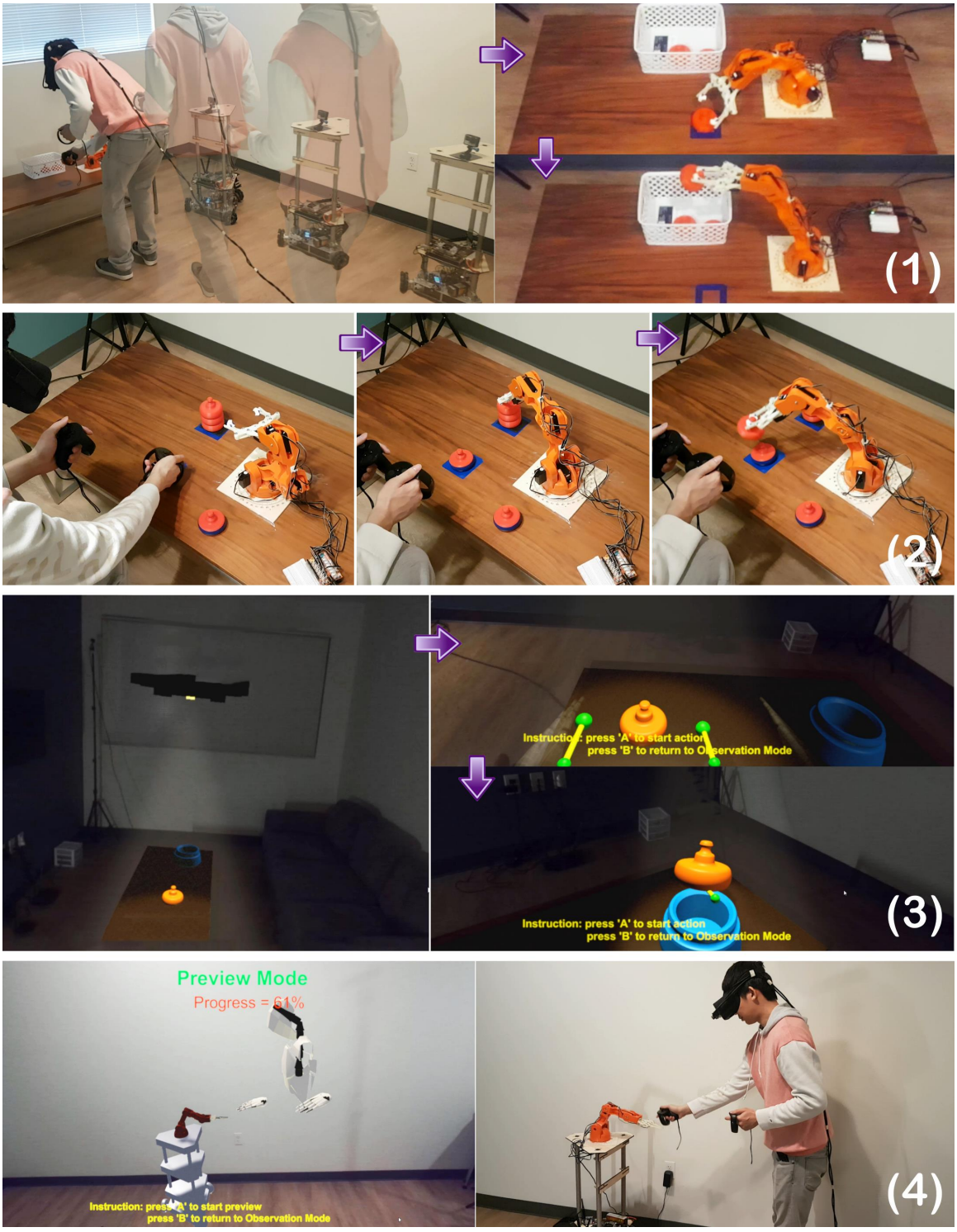
Fig. 5.5. Robot implementation workflow with ROS-Gazebo for realistic back-end simulation and Unity for front-end interaction and visualization.

the robot authoring user interaction effectiveness. The CamBot is an omni-mobile robot with a camera mounted. The ArmBot is a fixed 6-DOF robot arm (Arduino Tinkerkit Braccio). The GripperBot is an omni-mobile robot with the 6-DOF robot arm sitting on top of it. As is illustrated in Figure 5.5, the mobile robot base is powered by 3 DC motors (locally controlled by Arduino) driven by omni wheels that are capable of moving towards any direction while rotating. The robot is equipped with an NVIDIA Jetson TX1 Development Kit running ROS as robot's central controller and with a SICK TiM 561 2D LIDAR for SLAM navigation. The robot is powered by four LiPo batteries (11.1V, 5000mAh for each battery). During the *Robot Authoring Mode*, in order to deliver realistic virtual robot simulation that closely resembles the dynamics and physical behaviour of the real robot, we adopt ROS-Gazebo [177] as back-end robot simulator, the workflow is illustrated in Figure 5.5. In details, the controller inputs are sent to ROS-Gazebo using ROS#-Unity protocol [178] via WiFi communication. ROS-Gazebo then simulates the motion of robot under dynamic and physical constraints

(maximum torque, speed, acceleration, etc). Meanwhile, it simultaneously pushes the real-time robot status back to Unity3D where the virtual robot is then rendered accordingly in user's AR view. In this way, users are able to experience realistic robot manipulation and visualization with virtual robot avatars. Within the *Action Mode*, our collaboration model derives the corresponding robot behaviour into ROS-Gazebo, which then instructs the physical robot to act accordingly.

5.5 Use Case Scenarios

Figure 5.6 illustrates four use case scenarios of *GhostAR*. Figure 5.6-(1) demonstrates a HRC task where user carries an object and delivers it on the table, then the ArmBot put it into a basket (*Trigger*). The whole process is videotaped by the CamBot which follows the user (*Synchronize*) to get the best shooting angle. Figure 5.6-(2) demonstrates a joint assembly task with the ArmBot where user provides the bottom part of the assembly and the ArmBot grabs the top part and assemble them together. The task is authored as a *Trigger* action and can be performed repeatedly. Figure 5.6-(3) demonstrates a scenario where a drone is providing spot light for the user while he/she walks towards the couch, sits down, and puts the round object into the container. The entire HRC action is authored as one *Synchronize* task. Figure 5.6-(4) demonstrates a *Synchronize* hand-shaking scenario where the robot reaches out its gripper at the same pace as the human reaches out his/her hand, e.g., it pauses if the human pauses, and proceeds when the human proceeds.



Picture 5.6. Use cases. (1) Object handover with CamBot videotaping and following. (2) Joint assembly with ArmBot. (3) Object manipulation with drone providing spot light. (4) Hand shaking with GripperBot.

5.6 User Study

To evaluate our collaboration model accuracy, robot authoring interactivity, and overall usability of our system, we invited 12 users with various backgrounds to our three-session preliminary user study. None of them had prior experiences with our system and their ages ranged from 19 to 31. The study was conducted in a 5mx5m area using only virtual robots (the GripperBot and the Drone) for safety concerns. The entire process was video recorded for post-study analysis. Each user was given a 15 min tutorial about the background of the project before proceeding to the task in session 1. After each session, each user was given a survey to answer objective Likert-type questions. Each Likert-type item is graded by users from 1 to 5, on the usefulness of the feature and the level of agreement. After all the sessions, a conversation-style interview was conducted to acquire subjective feedback and a standard System Usability Scale (SUS) questionnaire was also given to each user. (P = participant)

5.6.1 Session 1: Human Authoring and Motion Mapping

One of the core features of *GhostAR* is to recognize user's body gestures and map it with previous authoring to output the corresponding robot behaviour. This is achieved by our in-situ generated collaboration model using DTW based algorithm. The first session of the study is designed to evaluate this with novice users.

Procedure. Users were asked to perform a continuous motion in the *Human Authoring Mode* that included six regular gestures (Figure 5.7-(1)): stand up from a chair (G_1), wave hand (G_2), pick up a virtual item (G_3), walk to another place and put down the virtual item (G_4), bow and reach out to the handles of a chair (G_5), push the chair a short distance and stand up straight (G_6). The whole motion series took approximately 30 seconds. The users then forwarded into the *Observation Mode* and put each of the above gesture into a *Trigger*

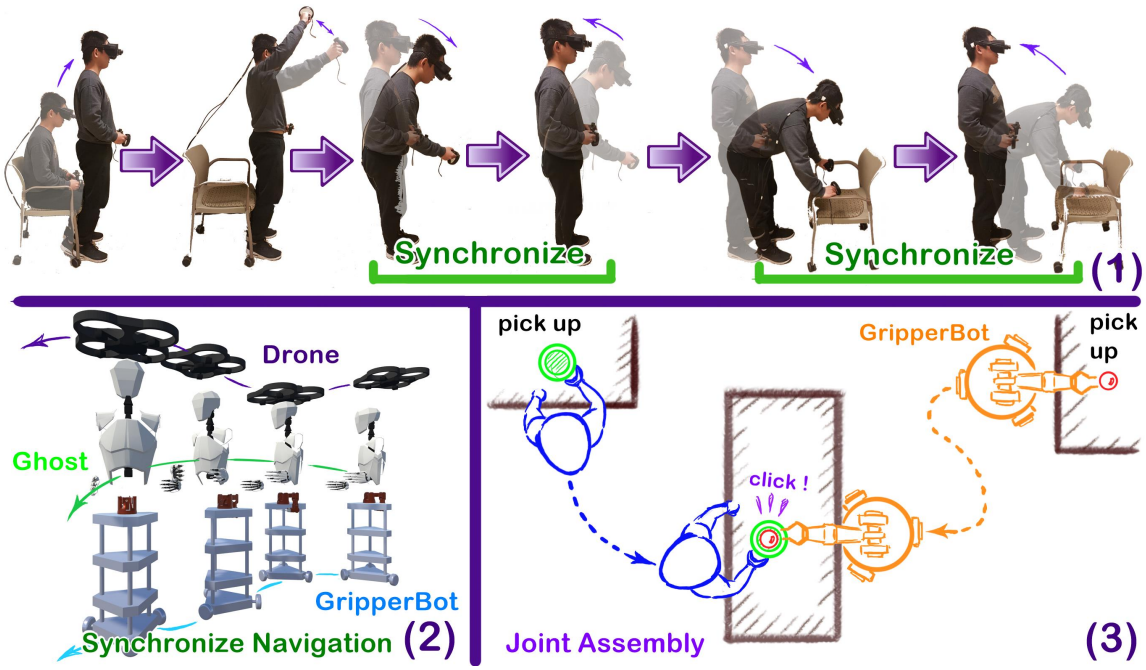


Fig. 5.7. User study setup. (1) Session 1: Human authoring and motion mapping. (2) Session 2: Robot authoring interactivity. (3) Session 3: System usability evaluation.

Group $T_i, (i = 1, \cdots, 6)$. Also, the object-moving motion between G_3 and G_4, and the chair pushing motion between G_5 and G_6 are *Group*ed as two *Synchronize* tasks $S_i, (i = 1, 2)$, respectively. Each user repeated the above process 4 times and all data set were recorded for a *cross validation*: using 1 set of data as authoring and 1 set as acting, to acquire large amount of evaluation results. For each *Trigger* task T_i, we collected the detection time from the collaboration model, t_{T_i}. For each *Synchronize* task S_i, we collect the estimated progress at time t by the collaboration model instead, noted as $P^{est}_{S_i}(t)$. The end time, $t^G_{T_i}$ of each *Trigger* gesture G_i, as well as the start time $t^{start}_{S_i}$ and end time $t^{end}_{S_i}$ of each *Synchronize* task S_i were manually labeled as ground truth.

Evaluation of *Trigger* task detection accuracy. Figure 5.8-(top) shows an example of the DTW distance values of a user (P4) in the *Action Mode*. All 12 users authored

846 valid *Trigger* tasks in total (6 gestures $\times$ 12 comparisons $\times$ 12 users), 840 of which were successfully detected (99.3%). For every detected *Trigger* task, we calculated the error of detection time $|t^G_{T_i} - t_{T_i}|$ and display the distribution in Figure 5.8-(Bottom). To better illustrate the *Trigger* detection accuracy, we calculate the 80% medians of the errors associated each gesture: $G_1 : 375ms, G_2 : 358ms, G_3 : 857ms, G_4 : 685ms, G_5 : 642ms, G_6 : 517ms$. These results indicate that in most of the cases ($> 80\%$), *GhostAR* system was able to detect the *Trigger* task within 1 second before or after the user had completed that gesture. We also observed that the accuracy of detecting the pick-up (G_3) and put-down (G_4) gesture was lower than that of the stand-up (G_1) and wave-hand (G_2). This is because of the motion involved in G_3 and G_4 has less amplitude, with only one hand moving in relatively smaller distance, resulting in lower detection accuracy.

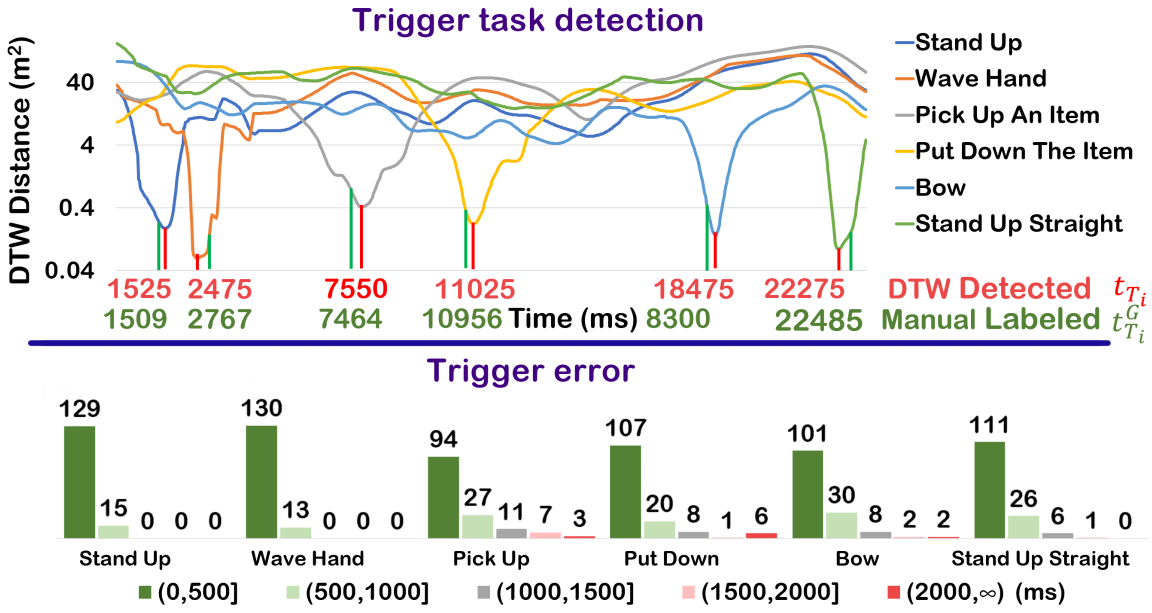


Fig. 5.8. *Trigger* task detection test. Top: DTW distance example from P4. Bottom: The distribution of *Trigger* task detection time error.

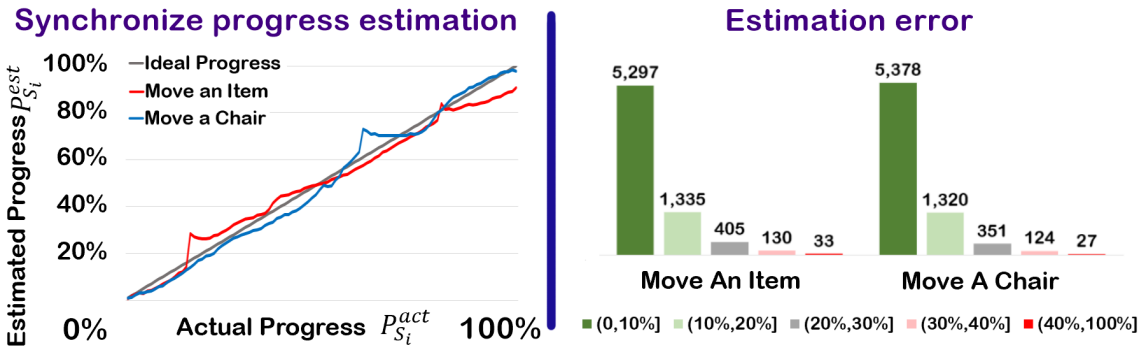


Fig. 5.9. *Synchronize* task progress estimation. Left: A progress estimation example from P4. Right: The distribution of estimation error.

Evaluation of *Synchronize* task progress estimation. We used the timestamp values t to characterize a user's progress in the *Synchronize* task. The actual progress is defined as $P_{S_i}^{act}(t) = (t - t_{S_i}^{start})/(t_{S_i}^{end} - t_{S_i}^{start})$ $(t_{S_i}^{start} < t < t_{S_i}^{end})$. Figure 5.9-(Left) shows an example of the $P_{S_i}^{act}(t)$ - $P_{S_i}^{est}(t)$ curve. For each *Synchronize* task S_i, we uniformly selected 100 data points from the $P_{S_i}^{act}(t)$ - $P_{S_i}^{est}(t)$ curve $(P_{S_i}^{act}(t) = 1\%, 2\%, \cdots, 100\%)$ and calculate the estimation error $|P_{S_i}^{act}(t) - P_{S_i}^{est}(t)|$. All 12 users contributed 14400 data points (2 *Synchronize* tasks $\times$ 100 data points $\times$ 6 comparison $\times$ 12 users) in total. The distributions of the estimation errors are shown in Figure 5.9-(Right). The 80% medians are 12.24% (object moving) and 11.73% (chair pushing), which implies that in most of the time ($> 80\%$), the robot will not surpass or fall behind a user for more than 15% of overall progress. Based on our observation during the user study, we suspect that the error may come from the minor inconsistency (e.g. irregular pause) of the user's behaviour during some of the motions.

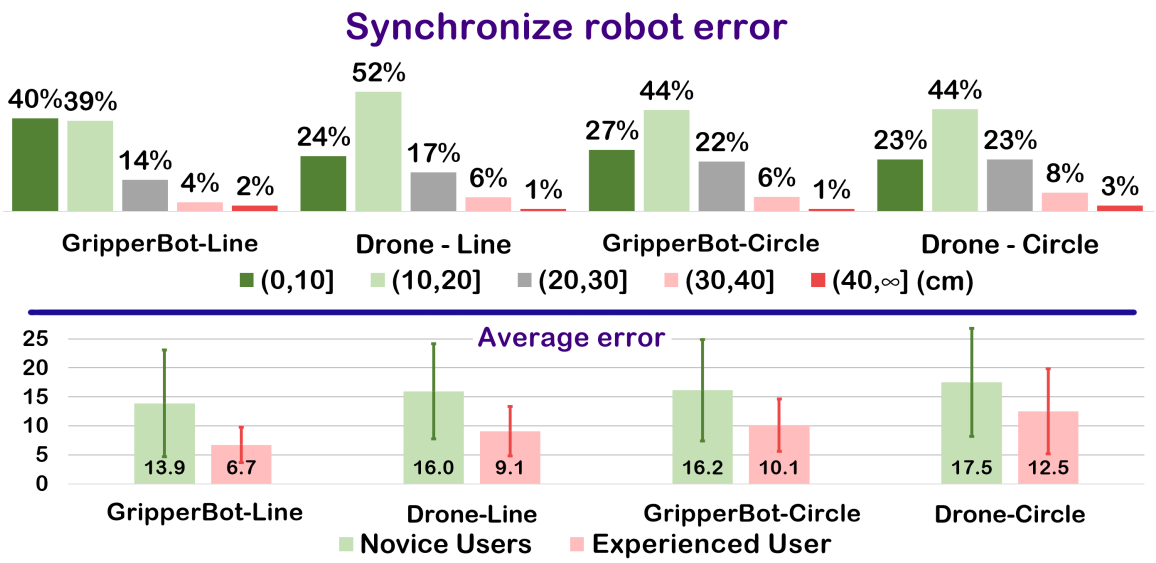


Fig. 5.10. Robot authoring interactivity test. Top: The distribution of robot authoring error. Bottom: Average error of novice users and an experienced user.

5.6.2 Session 2: Robot Authoring Interactivity

Another highlighted feature of *GhostAR* is to author robot to perform spatially and temporally synchronized motion with the human reference. In this session, we tested the robot interactivity and system interface towards authoring a *Synchronize* HRC task.

Procedure. A user first defined the human motion ghost by traveling through two routes: a straight-line and a circular path within the 5mx5m arena. Then we asked the user to author two virtual robots to travel alone with the human ghost while trying to coincide with the footprint (for the GripperBot) and the head position (for the Drone) of the human ghost, as illustrated in Figure 5.7-(2). The authoring data was recorded for accuracy analysis, and each user repeated the process twice.

Result and Discussion. In general, users were able to understand the robot authoring interaction quickly and all users successfully authored the described task. Many users frequently use the "pause/resume" feature to adjust themselves for better observing and

maneuvering perspective during the authoring. The histogram in Figure 5.10-(Top) shows the distributions of the robot authoring errors. Since the GripperBot and the Drone both has a radius of 25*cm*, we consider that the human ghost and the robot are aligned if the captured distance is shorter than 25*cm*. Based on this criteria, we calculated an *alignment rate* which is defined by the percentage of errors which are smaller than 25*cm*. The values of *alignment rate* are 89.57% (the GripperBot following a line), 86.87% (the Drone following a line), 84.29% (the GripperBot followed a circle) and 81.46% (the Drone following a circle). This result indicates that in most of the time ($> 80\%$), the users were able to author the robot to be precisely aligned with the human ghost for this *Synchronize* task.

By observing the study and analyzing the results, we find that keeping the error below 10*cm* was generally a hard task for regular users, especially for the Drone which has one added DOF than the GripperBot. We believe this is mainly because the users were not familiar with the kinetics mechanism of the robots. Restricted by the physical principals, the robots had large inertia and could not strictly follow the users' authoring behaviors as assumed. So that many users tended to overshoot while controlling the robots. Additionally, the Drone is always swinging due to its aerodynamics properties (simulated by ROS-Gazebo), which makes it even harder for maneuvering. Besides, the circular route evidently produced more error than the straight-line, which we assume is caused by the lack of next-position reference. We also compare the novice users with an experienced user who had practiced the authoring process 5 times. And display their average error in Figure 5.10-(Bottom). The result shows that the experienced user achieved much better accuracy result than the novice users. This indicates that the proposed robot interaction can be easily mastered with a few rounds of practise, and therefore better *Synchronize* performance can be achieved.

5.6.3 Session 3: System Usability Evaluation

In this session, we evaluated the overall usability of our system by asking users to author an HRC task, then act out the collaboration.

Procedure. The users were asked to complete a joint assembly task with the GripperBot, during which the user and the robot each picked up one part and met in the middle to put them together. The HRC task consists of a *Synchronize* action and two *Trigger* actions. As illustrated in Figure 5.7-(3), the collaboration scenario is described as follow: the users picked up his green part, *Trigger*ing the robot to pick up the red part; then they traveled towards the middle workstation at a *Synchronize* pace; when met, the users put down their parts first, *Trigger*ing the robot to place its red object and complete the assembly.

Result and Discussion. All participants were able to successfully act out the collaboration task with our system issuing the correct robot behaviour according to the authoring. The average task authoring time for task completions is 2 min 16 s.

The system feature related Likert-type results collected from the 3-session study are shown in Figure 5.11. After the tutorial, participants were generally confident to author the HRC task and agreed on the smoothness of our system workflow (Q9: avg = 4.25, sd = 0.62). "*It's fast and easy to plan a task, just role-plays your action and use the ghost reference to play the robot part. (P2)*" The timely authoring process and rapid iteration were appreciated by the users. "*I like how fast it is from planning the task to acting it out, encourages me to try more. (P4)*" We believe these feedbacks indicate that our system enables real-time and in-situ authoring, meeting our DG5. Users are also impressed with the motion mapping accuracy and robustness of our system during the Action Mode. "*I thought my acting was not that consistent with multiple pauses, but surprisingly your system recognized it and issues the correct robot behaviours. (P3)*" This comment indicates that we have achieved robot collaborative adaption in terms of coping with human partner's uncertainty (DG1).

The embodied authoring and interaction method (referred to as 'role-playing') is receptive to our participants, for both human ghost authoring (Q1: avg = 4.17, sd = 0.94) and robot avatar control (Q7: avg = 4.08, sd = 0.79). "*Moving a virtual robot in AR space was much easier than I thought. (P4)*" These comments have reflected positively to our DG2. The visualization accuracy of the ghost in terms of time-space reference is high according to (Q3: avg = 4.5, sd = 0.67). Further, the realistic robot simulation used for robot avatar interaction and visualization is also generally appreciated (Q6: avg = 3.83, sd = 0.94). "*That drone was kind of difficult to control. But I think the interaction method you provide is super realistic. The robot didn't move to where you were pointing to, it moved slowly to the target like a real robot. And for the drone, it was swinging and tilting when moving. (P7)*" We believe these comments confirm the necessity of adopting professional robotics

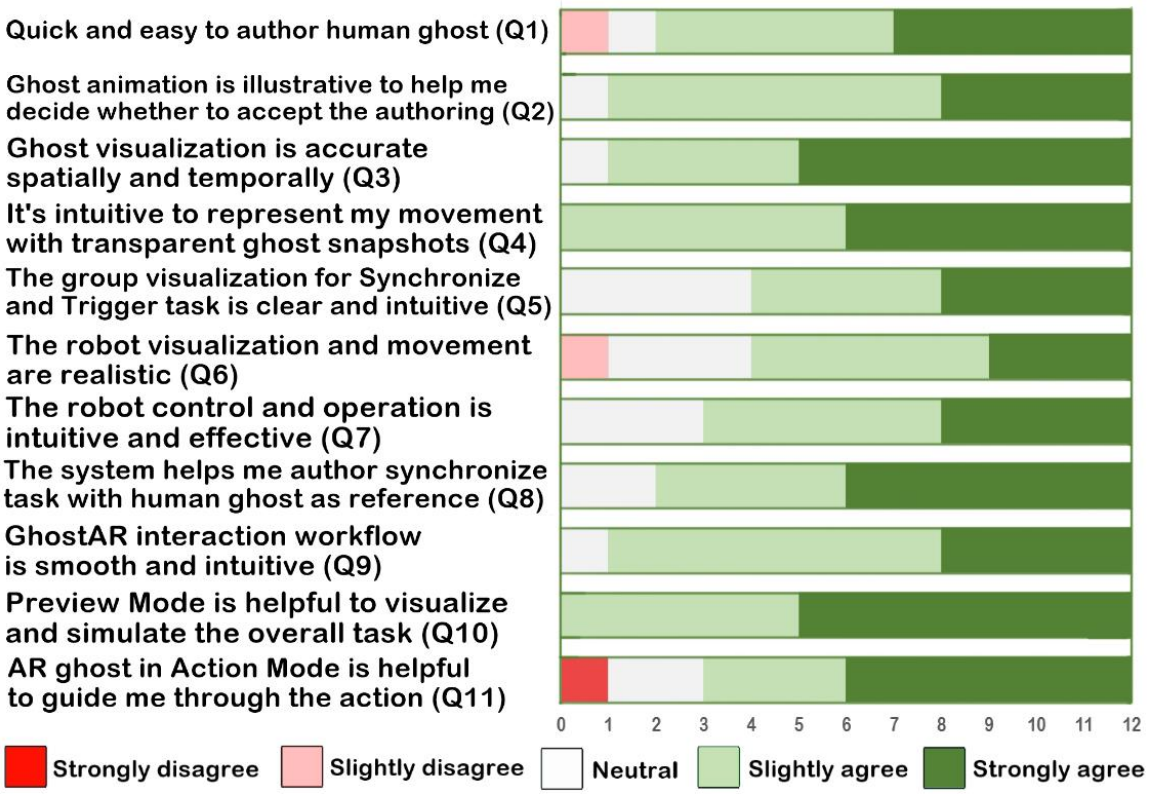


Fig. 5.11. Likert-type result after the three-session study.

engine (ROS-Gazebo) to enhance user interaction experience by providing back-end realistic simulation, meeting our DG4.

Survey responses were positive about the AR ghost to display the authored task in a spatially situated manner (Q4: avg = 4.5, sd = 0.52) with intuitive visual representation (Q5: avg = 4, sd = 0.85). The ghost images are welcomed as a time-space context for authoring collaborative robot task (Q8: avg = 4.33, sd = 0.78), as well as a visual guidance during the *Action Mode* for successful collaboration execution (Q11: avg = 4.08, sd = 1.24) "*It's very interesting like Sci-Fi, when I'm able to see what I have done with ghosts. (P3)"* The most popular feature of our system is the animation preview capability for the newly authored ghost (Q2: avg = 4.25, sd = 0.62) and the entire HRC task before action (Q10: avg = 4.58, sd = 0.51). "*The ghost animation is definitely my favourite part of the system, I can see so many potential applications for this technique. (P12)"* We believe these feedback match our goal of providing contextual aware authoring experience (DG3). The SUS survey was also deployed after the study, the response result is 80 with a standard deviation of 6.75, indicating high usability of the proposed system.

5.7 Discussion and Future Work

While users all appreciated the usefulness of AR ghost in terms of contextual visualization and task simulation, they have almost unanimously raised one interestingly conflicting problem. 6 out of 12 users have mentioned in one way or another that, the AR ghosts can occasionally become distracting and obtrusive. "*There are too many ghosts in front of me when I am trying to see and act. (P10)"* This feedback emerges after user get familiar with the system and they start feeling not needing the AR guidance *all the time*. This finding brings out an important question when designing such systems: **how shall we balance between *demonstrative ghost reference* and *clear authoring view*, and provide both for the user?** While this may be a research question for future endeavor, we have some initial

thoughts. A quick fix could be giving user the ability to manually toggle all the AR ghost. However if user only wants to hide *some* of the ghost images, the added interaction could increase the cognitive load of the user. Another potential solution involves intelligently detecting user's intention and only display the most relevant and needed ghost. For example, during the *Action Mode*, the ghost appears only when user is about to go off-track.

In this work, we prototyped our system with see-through HMD AR and achieved body externalization with IR-based tracking device. The current hardware setup provides only 3-joints tracking (head and 2 hands) and we utilized only the position value, resulting in a 9-dimensional input data for our collaboration model. Note that this setup is largely limited by the currently available hardware platform, and is likely to change. For example, future AR-based body tracking technique is expected to have multiple-joints and provides more realistic humanoid ghost. Furthermore, with additional sensory input embedded, such as tactile force feedback, we can achieve force sensitive collaborative authoring with our system, such as joint object carrying.

Although the *GhostAR* system is able to detect the user's motion status with fair accuracy, the DTW algorithm we are currently using largely relies on user's consistency in order to achieve satisfying performance. As a result, the user in *Action Mode* is constrained to the previously authored motions and has very limited flexibility. To tackle this problem in the future, our initial guess could be utilizing the state-of-the-art human action recognition approaches, such as probabilistic methods and deep neural networks, to capture the key features in the user's motion. Thus granting more freedom to the user and enabling for intuitive authoring and acting behaviour while maintaining collaborative accuracy.

It is worth emphasizing that *GhostAR* is a HRC task authoring and acting platform designed as complimenting workflow for the more advanced human-robot-collaborative learning frameworks, as discussed in the Related Work section. Our system can essentially be applied to many other HRC models specializing in different applications, to achieve

higher level of collaborative intelligence while empowering users with real-time, spatially situated visual task authoring capability.

5.8 Conclusion

We have presented *GhostAR*, a human-robot-collaborative task authoring system featuring role-playing embodied interaction and contextually situated visual editing. In this paper, we have demonstrated how an AR interface can be synergistically integrated with embodied authoring to create elevated HRC experience. We have proposed key guidelines for HRC authoring system design, highlighting 1) robust motion adaption, 2) natural embodied interaction, 3) contextual authoring reference, 4) realistic visual simulation, and 5) fluid real-time iteration. Our three-session system evaluation received positive results, indicating that the proposed system has reached the design goals, while also unveiling the potential directions for future endeavors. *GhostAR* has created a brand new perspective to solve the balancing problem between sophisticated functionality and intuitive interaction in an adaptive collaboration context, thus offering future inspirations to the HCI and HRI community.

6. AN EXPLORATORY STUDY OF AUGMENTED REALITY PRESENCE FOR TUTORING MACHINE TASKS (AVATUTAR-STUDY)

This chapter is a slightly modified version of "***An Exploratory Study of Augmented Reality Presence for Tutoring Machine Tasks***" [4] published in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* and has been reproduced here with the permission of the copyright holder.

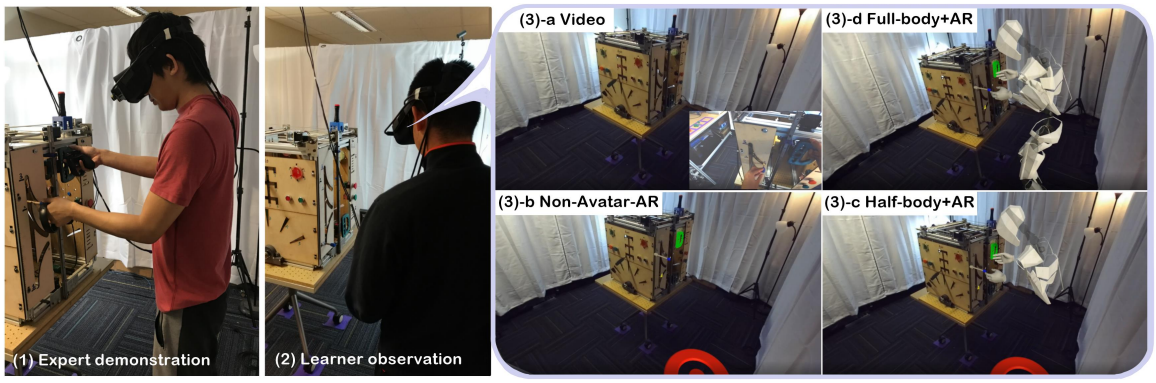


Fig. 6.1. An overview of our exploratory study setup. An expert first generates a tutorial of a machine task on the mockup machine through embodied demonstration (1). Later a student tries to repeat the task by following this tutorial through an augmented reality (AR) headset (2). We propose to explore four tutor presence options for machine task tutoring, including: *video* (3)-a, *non-avatar-AR* (3)-b, *half-body+AR* (3)-c and *full-body+AR* (3)-d.

Machine tasks in workshops or factories are often a compound sequence of ***local***, ***spatial***, and ***body-coordinated*** human-machine interactions. Prior works have shown the merits of video-based and augmented reality (AR) tutoring systems for *local* tasks. However, due to the lack of a bodily representation of the tutor, they are not as effective for *spatial*

and *body-coordinated* interactions. We propose avatars as an additional tutor representation to the existing AR instructions. In order to understand the design space of tutoring presence for machine tasks, we conduct a comparative study with 32 users. We aim to explore the strengths/limitations of the following four tutor options: ***video***, ***non-avatar-AR***, ***half-body+AR***, and ***full-body+AR***. The results show that users prefer the *half-body+AR* overall, especially for the *spatial* interactions. They have a preference for the *full-body+AR* for the *body-coordinated* interactions and the *non-avatar-AR* for the *local* interactions. We further discuss and summarize design recommendations and insights for future machine task tutoring systems.

6.1 Introduction

Contemporary manufacturing facilities are changing to focus on flexible, modular, and self-configuring production, a trend that is sometimes called *Industry 4.0* [179]. Human workers, as the most adaptive part of the production process, are expected to operate various machinery and equipment in a constantly changing working environment [180]. This creates a new challenge that requires workers to rapidly master new machine operations and processes, what we refer to in this paper as ***machine tasks***. Researchers have proposed low-cost, easy-to-distribute, and highly-scalable machine task tutoring systems as a way to resolve this challenge. Recent novel tutoring systems show potential to reduce and eventually eliminate real-human one-on-one tutoring [181].

Machine tasks in a workshop or factory environment are usually a mixed sequence of various types of interactive steps. Based on our observations and literature reviews, we categorize the steps of the machine tasks into three types: ***local***, ***spatial***, and ***body-coordinated*** [182, 183]. A *local* step refers to one-hand interactions in the user's immediate vicinity (i.e., within arms reach), which involves no spatial movement. A *spatial* step requires a large spatial navigation before proceeding to interact with the target machine

interface. And in a *body-coordinated* step, an operator must coordinates his/her body, hands, and eyes to complete the interaction.

Video content has been widely adopted into modern tutoring systems because they are capable of illustrating the fine details of operations [184–188]. Despite their popularity, video tutorials fundamentally suffer from the lack of a spatial connection between the digital representation and the user's physical presence. This flaw of video tutorials can lead to a fractured learning experience, especially for physically interactive tasks. To address this challenge, augmented reality (AR) approaches have been proposed that superimpose virtual tutorial guidance directly onto the interaction target in-situ [189]. Due to this advantage, AR tutoring systems have been particularly favored for interactive tasks within the physical environments, such as in machine-related operations [117, 120, 123, 124].

However, existing AR tutoring systems for machine-related operations predominantly focus on *local* interactions. The virtual tutoring contents in these works usually apply visual illustrations, such as static and dynamic symbols and text, to represent the operations within the local regions of interest. Previous works have shown their effectiveness for highly-complex local instructions, such as computer assembly [123], machinery diagnosis [113], and vehicle maintenance [114]. However, due to the lack of an explicit visual representation of the human tutor for spatial and bodily movements, these symbol-only AR illustrations are inadequate to provide clear cues for interactions that require large spatial navigation movements and full body coordinative operations, such as the machine tasks.

To guide the development of improved AR tutoring systems, we propose to use avatars as an enhanced tutoring presence to the existing AR instructions. In our approach, the embodied demonstration of the tutor is presented in the operator's AR view while they interact with the physical machines in-situ. Virtual avatars have been broadly used to represent the embodiment of the human users in various virtual reality (VR) consumer applications, such as VR-chat [190]. Avatars have also been explored and adopted in the

area of mixed reality (MR) remote assistance [135, 136, 191], body movement guidance and training [130, 134, 141], and telepresence AR conference [192, 193]. Most recently, Loki [137] has demonstrated the avatar's potential for facilitating physical tasks via remote instructions. However, a systematic study of an avatar based AR presence is still lacking, especially in the context of machine task tutoring.

To this end, we investigate two research questions to reveal future research directions for the design of machine task tutoring systems. (i) Is the additional avatar presence beneficial to the user's experience and performance in a comprehensive machine task tutoring scenario, compared with the *non-avatar-AR* and *video* tutorial options? (ii) How to optimize the design of the tutor presence to achieve improved tutoring experience for future machine task applications?

To answer these questions, we develop two different avatar tutor presentations: *half-body* and *full-body*. Together we compare the following four tutor presence options: ***video***, ***non-avatar-AR***, ***half-body+AR***, ***full-body+AR***. Along these options, we gradually increase the guidance visualization levels, aiming to provide insights for an ideal design. All four options of the machine task tutorials are created from one single source, which is the embodied physical demonstration of the expert human tutor, as illustrated in Figure 1. We conduct a study with 32 users across four different tutor options, with a specially created mockup machine as the machine task testbed. The contributions of our paper are as follows.

- **Study System Design and Implementation** of a machine task scenario to compare all four tutor options in parallel, where *local*, *spatial*, and *body-coordinated* interactions are composed into multi-step tutorial sessions.
- **Quantitative and Qualitative Results** showing users' objective/subjective responses and tutor preferences after completing the sessions of machine tasks while following different tutor options.

- **Design Recommendations and Insights** summarized from the results and discussions of the study, providing valuable guidance for future machine task tutoring system design.

6.2 Machine Task tutoring

6.2.1 3.1 Machine Task: Local, Spatial, and Body-coordinated

This paper presents a study of AR presence for ***machine tasks*** tutoring system design. We define a machine task as a sequence of steps involving machine operations and spatial navigation, particularly for applications in production. A machine task is commonplace in workshop and factory environments, for the purposes of parts manufacturing, assembly, and equipment maintenance, repair, and overhaul. A ***step*** is the unit of a machine task sequence, which represents a meaningful inseparable action of the human-machine interaction. The steps in a machine task are usually a mixture of various types of interactions. In this paper, we focus on transferring knowledge regarding human actions. Therefore we elect to categorize the machine task steps by the level of movement required for the human-machine interaction. Based on our observation and engineering knowledge, as well as reviews from prior literature [182, 183], we classify the steps into the following three categories:

- A ***local*** step is a one-hand human-machine interaction in the user's current location and perspective. The user does not need body-scale spatial movements before interacting with the machine, nor does he/she need compound body-hands-eyes coordination for the action. Example *local step*s are simple actions with machine interfaces, such as with buttons, sliders, handles, knobs, and levers.

- A ***spatial*** step requires the user to perform noticeable spatial navigation before the machine interaction. The key challenge of this type of action is locating the target interface. Example *spatial step*s are tool change tasks during the machining operation

Fig. 6.2. An example real-life machine task scenario involving *local* (1), *body-coordinated* (2), and *spatial* (3) interactions.

that require the user to navigate to the designated area and find the right tool; or interactions with the machine interfaces that are away from the user's current location.

- A ***body-coordinated*** step is usually a two-hand action that requires the user to coordinate his/her body, hands, and eyes to complete the task. Example *body-coordinated step*s are the actions that operate two machine interfaces with two hands, respectively, in a synchronized or cooperative manner.

Figure 6.2 illustrates an example machine task using a band saw machine to manufacture a part. The user first needs to configure the machining parameters through a button and a knob, which are *local step*s (Figure 6.2-(1)). The user also needs to adjust the cutting angle and cutting-saw height using both hands in a coordinated manner, which is a *body-coordinated step* (Figure 6.2-(2)). Before starting the machine, he needs to choose a base material meeting his production requirement from the material storage station, which is a *spatial step* (Figure 6.2-(3)). Note that in this study we focus on human-machine operations performed by the hands only, machine operations involving the feet are outside of the scope of this study.

6.2.2 Tutor Design from Embodied Authoring

When an apprentice is trying to learn a new machine task in a factory, the most effective way is to observe and follow the demonstration of an experienced master. We take the master-apprentice paradigm as an inspiration for our tutoring system design. The machine task tutorials in this paper are created from recording the physical demonstration of an expert (Figure 1-(1)), and it is displayed to users with the different visual presentations of the tutor (Figure 1-(2,3)). This paper focuses on exploring the tutor's *visual* representation only and does not include input to other senses, such as audio and tactile. We explore a design space of tutor presence in AR which involves spatial recording of the embodied authoring from an expert (i.e., the expert creates the tutorial by demonstrating the procedure).

Video. This tutor option mainly serves as a benchmark, since video is a popular tutoring media. To adapt video to fit our design space of embodied AR authoring, this option uses a video recording of the expert's first-person view while they demonstrate the task. The video recording is displayed to the user in a picture-in-picture style (Figure 1-(3)-a) at a fixed location and orientation in their visual field. Similar approaches have been used in prior work [185, 194].

Non-avatar-AR. This tutor option is similar to the existing AR instructions found in the machine-related tutoring systems discussed in our review of related work. It utilizes animated superimposed virtual models to represent the movement of the real part, aided with guiding symbols like arrows and text (Figure 1-(3)-b). A red circle on the ground indicates the spatial location of the tutor when they were recording. This tutor option represents the baseline of the existing AR instructions. A more detailed demonstration list for various machine interfaces can be found in Figure 6.3.

Half-body+AR. This tutor option displays an additional half-body avatar on top of the *non-Avatar-AR* option. The half-body avatar only has a visualization of the upper body and two arm-less hands, with the red circle indicating the ground position (Figure 1-(3)-c). Since

all of the human-machine interactions in this paper are hands-only, we expect the virtual hands of this avatar to be sufficient for expressing the interaction. The head model indicates where to look and pay attention, while the upper-body plus the ground circle represent the spatial location of the tutor. This style of the avatar visualization focuses on simplicity and is similar to the approaches used in prior research and commercial products [137, 171].

Full-body+AR. This tutor option displays an additional full-body avatar on top of the *non-avatar-AR* option. The avatar has a complete humanoid body structure, including arms and legs (Figure 1-(3)-d). Even though our tasks do not involve feet interactions, we choose the style of this avatar visualization due to its higher similarity to a real human tutor. The full-body avatar has already been widely adopted by prior work in various applications, such as ballet [195] and tennis training [196], Tai-Chi practice [197], MR remote collaboration [135], and telepresence meeting [192]. In our case, we are particularly interested in finding out whether and in what way the added avatar visualization would improve the user's understanding of the tutor's bodily movement.

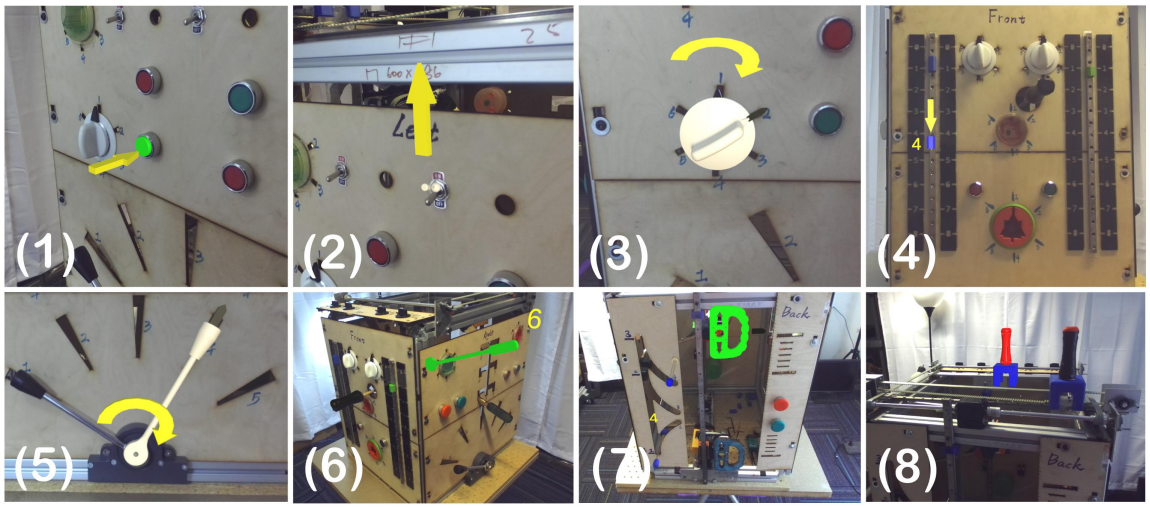


Fig. 6.3. Example AR instructions for various machine interfaces: (1) button, (2) switch, (3) knob, (4) slider, (5) lever, (6) side-shift, (7) back-shift and (8) 2-DOF curve handle.

Both the avatar tutor options include the AR instruction of the *non-avatar-AR*. While we agree on the necessity for intuitive and accurate instructions, our interests lie in understanding the effect of the added avatar visualization in the machine task tutoring scenarios. The four proposed tutor options represent the current mainstream AR-avatar related tutorial media. We design them to present the same instruction accurately while gradually incrementing their levels of guidance visualization. By studying the users' reactions under these four conditions, we aim to scale the weight of avatars in the AR tutoring systems and reveal the potential strengths and limitations of using them. Further, based on the study results, we seek the balance points in the level of visualization details for practical AR tutoring scenarios.

6.2.3 Implementation

Our see-through AR system is developed by attaching a stereo camera (ZED Dual 4MP Camera with a 2560 $\times$ 720 resolution at 60 fps and a field of view (FOV) of 90° (H) $\times$ 60° (V) $\times$ 110° (D) [198]) in front of a VR headset (Oculus Rift [171]), which is connected to a PC (Intel Core i7-9700K 3.6GHz CPU, 48GB RAM, NVIDIA GTX 1080). The positional tracking is enabled by four external sensors (Oculus IR-LED cameras), covering an effective area of 3 $\times$ 3m. To represent our *half-body* and *full-body* avatar, we choose a robotic humanoid avatar created by Noitom [199] due to its unbiased sexuality. We also adopt the hands model from Oculus Avatar SDK due to an expressive gesture visualization. Our system is developed using Unity3D (2018.2.16f1) [200] for both tutorial authoring and playback. The full-body avatar is estimated from the three-point tracking (head and two hands) via inverse kinematics powered by a Unity3D plugin (FinalIK [201]).

6.3 Exploratory User Study

6.3.1 Study Setup: the Mockup Machine

In order to conduct our study, we first need to create a study scene to simulate machine tasks, that is capable of *local*, *spatial*, and *body-coordinated* human-machine interactions. We therefore created a mockup machine as the testbed for our study. The design of the mockup machine is guided by the following considerations: 1) The mockup machine should mimic real-life machine operation with realistic physical interfaces. 2) The size of the machine should be large enough to facilitate spatial navigation and bodily movement. 3) The machine should be designed with enough complexity to support the test sequence designed for the machine tasks. 4) Each interface on the mockup machine should provide multiple interaction possibilities in order to test and measure the user's performances.

Figure 6.4-Top illustrates the detailed design of our mockup machine ($0.7\times0.7\times0.7$ m), which is placed in the center of the study area (Figure 6.4-(f)), on top of a table (height = 0.78 m). The mockup machine can support *local* interactions via the following five interfaces: button, switch, knob, slider, and lever. It can support *spatial* interactions by asking the users to operate an interface on another side of the machine, which requires the users to first navigate spatially then locate the target interface before the interaction. We also designed a *spatial* 'key' interaction, simulating real-life tool change and assembly operation. In this interaction, users first need to go to the *key station* (Figure 6.4-(e)) and find the correct key, then walk back and insert it into a designated keyhole. As for *body-coordinated* interactions, we present a list of example interactions in Figure 6.4-(a-d). The first type of *body-coordinated* interaction supported by the mockup machine is operating two interfaces (slider-slider, slider-lever, lever-lever) with two hands respectively, in a synchronized manner (Figure 6.4-(a)). We've also specially designed three *body-coordinated* interfaces, including two 'shift' interfaces (Figure 6.4-(b,c)) that require user's

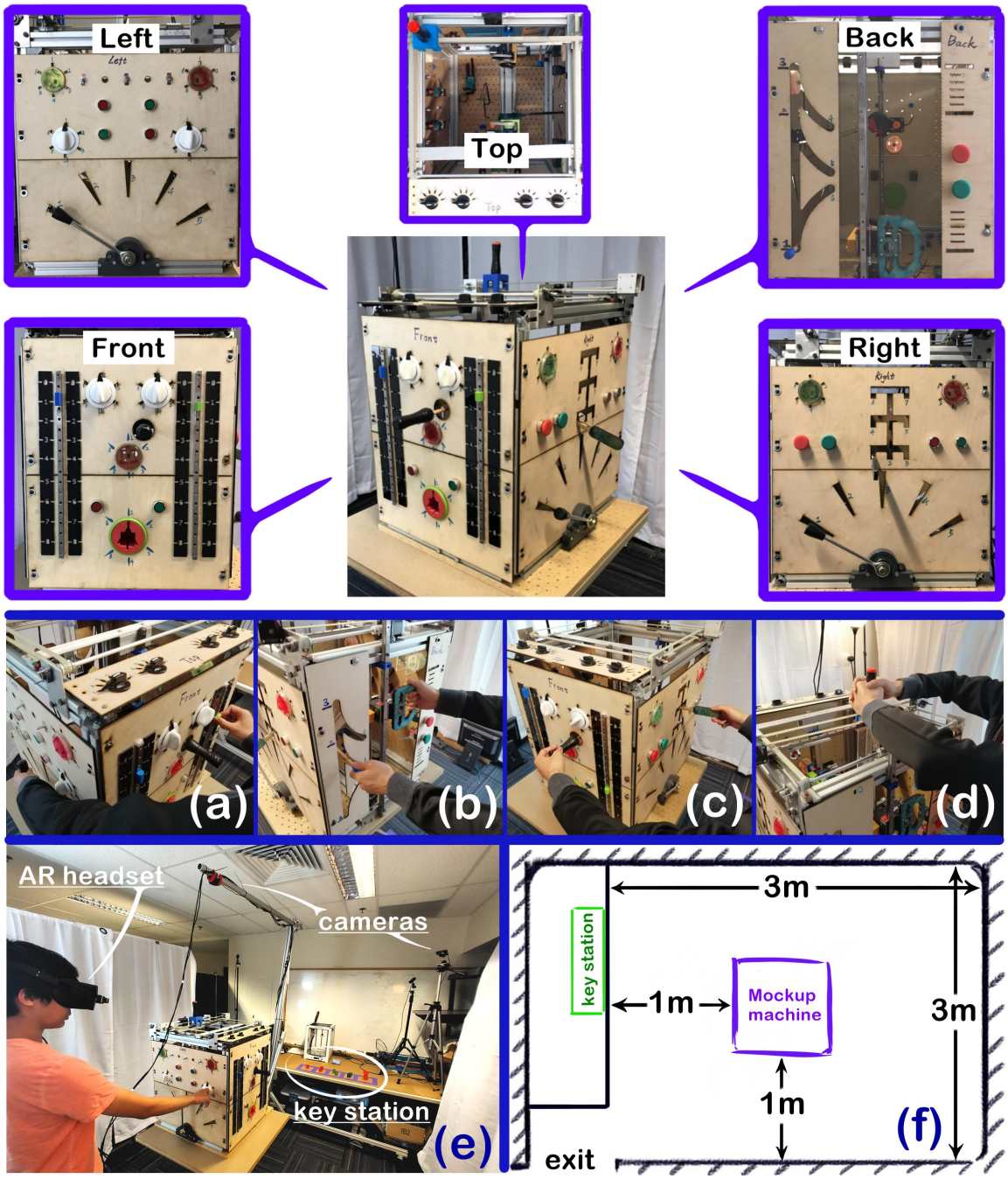
both hands to operate in a cooperative manner; and a ‘curve’ interface requiring user’s body-hands-eye coordination while operating the 2-DOF handle to repeat the trajectory in the tutorial (Figure 6.4-(d)).

6.3.2 Study Design

During the study, each user was asked to follow the tutor and complete four sessions of machine-operating task sequences. For each different session, the user followed a different tutor option to complete a different task sequence.

Sequence design. Each machine task sequence in the study consists of 36 steps, that are roughly evenly-distributed into three interaction categories: 1) *local* (10 steps including 2*button, 2*switch, 2*knob, 2*slider, and 2*lever), 2) *spatial* (14 steps including 2*button, 2*switch, 2*knob, 2*slider, 2*lever, and 4*‘key’; these steps require large spatial navigation before interacting with the target interface), and 3) *body-coordinated* (12 steps including 2*slider-slider, 2*slider-lever, 2*lever-lever, 2*‘side-shift’, 2*‘back-shift’, and 2*‘top-curve’). The four sequences are designed with the same step composition and execution order, to ensure the same task difficulty. To avoid memorization from previous sequences, the corresponding steps across different sequences have different detailed interactions. For example, step-1 on sequence-3 asks the user to twist the *right knob* to *position-3*, while the same step on sequence-4 asks the user to twist the *left knob* to *position-4* instead.

Tutorial length normalization. It’s likely that the duration of a tutorial demonstration will affect users’ task completion time. Since we created the tutorial for each of the four machine task sequences separately, the duration of corresponding steps across the different tasks are different. To enable a direct comparison of task completion time for corresponding steps across tasks, we scaled each step’s duration to the average duration across the four corresponding steps, by slowing down or speeding up their playback. This procedure was performed for each set of four corresponding steps across the 36 steps in each sequence.



Picture 6.4. Top: The mockup machine detail design. Middle: example Body-coordinated machine interaction, including (a) two-interface synchronized operation, (b) back-shift, (c) side-shift, (d) top curve. Bottom: (e,f) study area setup layout.

Data counterbalancing. To mitigate learning effects, the order in which participants used the different tutor options was counterbalanced across participants, such that each tutor option was tested on each ordinal position (first, second, third, fourth) with equal frequency. This was achieved by shuffling tutor options evenly with respect to the session order, resulting in a pre-arranged rotation list of 4 (sessions) * 4 (tutor options) = 16 participants. In total, we invited 16*2 = 32 participants for a balanced data acquisition.

6.3.3 Participants

We recruited 32 users from our university via emails, posters, and networks (22 male and 10 female students between the ages of 18 and 35, $M = 23.8, SD = 3.64$). Each user was compensated $10. We did not particularly seek participants with AR/VR experience or machine operation skills for unbiased potential insights. We measured their familiarity with AR/VR on a 7-point Likert scale, with 1 being a total non-experienced user and 7 being an expert developer, yielding a result of $M = 3.63, SD = 1.43$. We also surveyed their general experience with hands-on interactions with machine-like objects ($M = 3.47, SD = 1.54$). Further, we asked users to rate their familiarity of self-teaching using any forms of tutorials ($M = 4.03, SD = 1.57$). An illustration of the user's demography survey results can be found in Figure 6.5.

6.3.4 Procedure

After completing the demographic survey, each user received a 5 minute introduction about the study background and a brief demonstration of how to interact with each interface on the mockup machine. The users then proceeded to the four sessions one by one, each session took the user about 10 minute to interact with the mockup machine and 5 minute afterward to fill out a user experience survey questionnaire. During each session, users

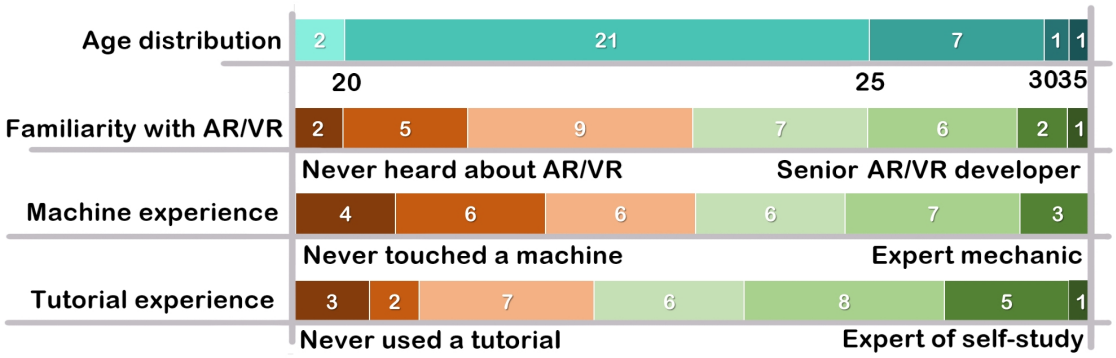


Fig. 6.5. Demography of 32 participants.

were asked to wear the AR HMD and follow the machine task tutorial step by step. Users were asked to perform the machine operations at the comfortable speed of their choices, with no need to hurry or drag. A researcher monitored the entire process through the users' first-person AR view. If the researcher observed the user had completed the current step, he would switch to the next step and notify the user verbally. After completing the four sessions, users filled out a preference survey comparing the four tutor options, then finished up the study with a conversational interview.

6.3.5 Data Collection

Each user's study result contains three types of data: (1) tutorial following performance, (2) 7-point Likert subjective rating and user preference survey, and (3) conversational feedback.

Video Analysis. We recorded the entire study process using three cameras. The main source of objective data came from the video record of users' first-person AR view during the human-machine operation. We segmented this video into steps and manually coded

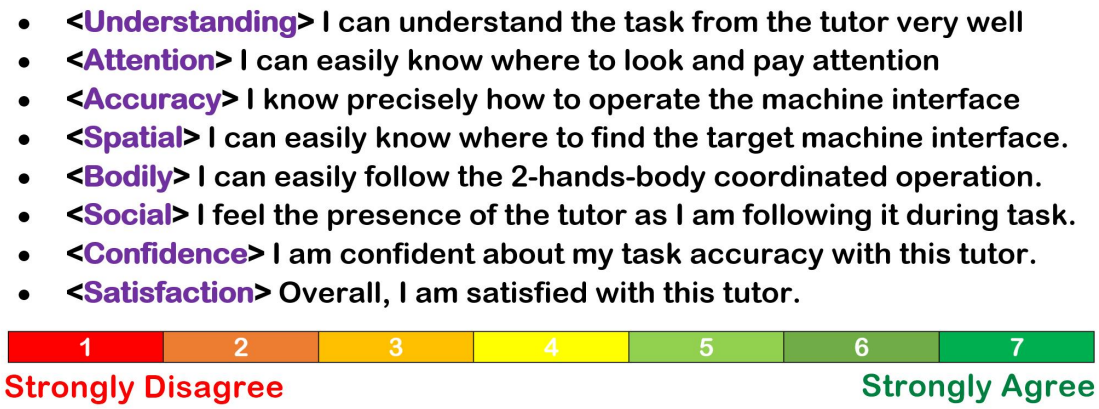


Fig. 6.6. User experience survey questionnaire.

the completion time and correctness of each step. Here we consider a step as completed if the user finished interacting with the interface and retrieved his/her hand. Also, we regard a step as completed correctly if the user interacted with the correct target interface and performed the correct positional manipulation (for slider, knob, lever, etc.), according to the corresponding tutor's demonstration. This yielded a total of 32(users)*4(sessions)*36(steps) = 4608 steps of objective analysis data across the entire study. We also had a top-view camera capturing the trajectory of the 'curve' interaction for accuracy analysis and a third-view camera recording from the top corner of the study scene for additional references.

Questionnaire. After each session, users rated their experience and subjective feelings for this session's tutor option using a 7-point Likert survey. The design of the survey question was derived from the standard user experience surveys, including *Single Ease Question (SEQ)* [202], *Subjective Mental Effort Question (SMEQ)* [203], *System Usability Scale (SUS)* [204], and *Networked Mind Measure of Social Presence (NMMSP)* [205], with added machine task elements and fine tuned specifically to our application scenario. The detailed questions are shown in Figure 6.6.

Interview. We audio-recorded all the subjective comments and suggestions from the users for post-study analysis and summary. During the study, we encouraged the users to 'Think Out-loud' to capture any on-the-fly insights as they were following the machine-operating tutorial. After the four sessions, we interviewed the users by asking their preference comparing all the tutor options for the machine task overall, as well as specifically for *local*, *spatial*, and *body-coordinated* interactions. The subjective feedback is later used in the paper to explain the study results and inspire for future design insights.

6.4 Results

In this section, we present the results of this study. We first show the users' objective performances and subjective ratings, as well as tutor preferences. Then we provide a summary and explanatory analyses for the results using interview feedback and our observation.

6.4.1 Objective Performance

We first demonstrate the overall user performance by comparing four different tutor options. Then we present detailed user performances regarding each interaction category: *local*, *spatial*, and *body-coordinated*. We measure the completion time and accuracy, which reveals how efficiently and accurately the users understand the tutorials. Since the tutorial for each step has a different duration, we normalize the completion time of each step as: actual step completion time divided by the duration of the step demonstration in the tutorial. The tutorial duration for a machine task sequence (36 steps) is: 6 minutes 15 seconds, with each step's length ranges between 4.9 to 19.3 seconds, while the average completion time of a sequence is: 7 minutes 21 seconds. The accuracy of a category of steps is calculated as: the number of correct steps divided by the total step number. To characterize the accuracy of the 2D 'curve' operation, we calculate the Modified Hausdorff Distance (MHD) [206] between

the trajectory performed by the user and the one in the corresponding tutorial, with a smaller distance indicating more similarity and higher accuracy. The normal distribution assumption is violated by our dataset as indicated by Shapiro-Wilk normality test ($p < 0.005$). Hence to examine the statistical significance across the four tutorial options, we conduct a Friedman test with a Wilcoxon signed-rank, rather than the repeated ANOVA measures. All results are presented in Figure 6.7.

Overall performance. The average normalized completion time shows that the users spend the longest amount of time following the *video* tutorials ($M = 1.58, SD = 0.70$) and is significantly slower than *non-avatar-AR* tutor option ($M = 1.16, SD = 0.57$) ($Z = -18.416, p < 0.0005$). Among all the AR options, the *half-body+AR* tutorials ($M = 1.14, SD = 0.55$) shows marginally shorter completion time than *non-avatar-AR* ones, with

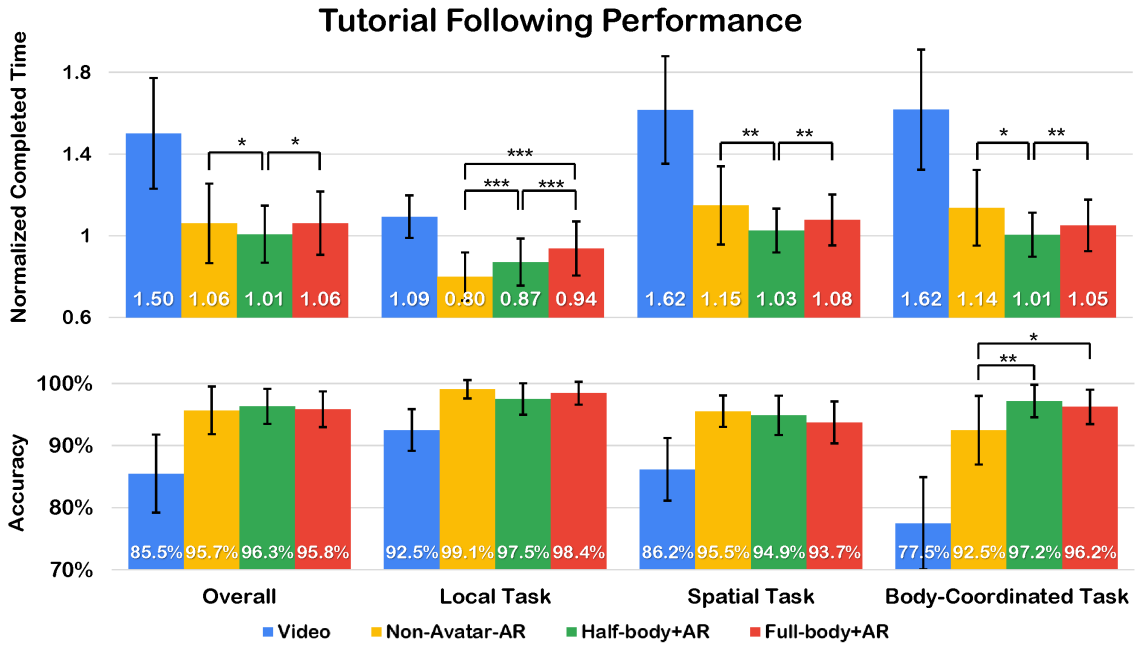


Fig. 6.7. Tutorial following performance. (***=p<.0005, **=p<.005, *=p<.05. If not specified, *** between the video options and other three tutor options.) Error bars represent standard deviations.

no significant edge ($Z = -0.854, p = 0.393$). Meanwhile, users with *full-body+AR* tutorials ($M = 1.15, SD = 0.47$) perform slightly slower than the ones with *half-body+AR* ($Z = -2.527, p < 0.05$). The accuracy result reveals the same trend as the completion time. The *video* tutorials has the lowest accuracy ($M = 85.4\%, SD = 6.28\%$) while the accuracy of *non-avatar-AR* ($M = 95.6\%, SD = 3.82\%$), *half-body+AR* ($M = 96.3\%, SD = 2.82\%$) and *full-body+AR* ($M = 95.8\%, SD = 2.87\%$) options are approximately equally high (pairwise $p > 0.05$).

***Local* steps performance**. Similar to the overall performance, the *video* option still has the poorest performance in terms of task completion time ($M = 1.09, SD = 0.21$) and accuracy ($M = 92.5\%, SD = 3.36\%$). Interestingly, users with *non-avatar-AR* tutor option ($M = 0.80, SD = 0.24$) are significantly faster than the ones with *half-body+AR* tutor ($M = 0.87, SD = 0.23$) and *full-body+AR* tutor ($M = 0.93, SD = 0.26$) ($Z = -4.487, p < 0.0005$ and $Z = -6.189, p < 0.0005$ respectively), which implies that the existence of an avatar may have negative influence on the user's perception for *local* task understanding. In terms of the accuracy, no significant difference was found among *non-avatar-AR* ($M = 99.1\%, SD = 1.48\%$), *half-body+AR* ($M = 97.5\%, SD = 2.54\%$), and *full-body+AR* ($M = 98.4\%, SD = 1.84\%$) (pairwise $p > 0.05$).

***Spatial* steps performance**. The *video* option takes the longest time to complete ($M = 1.62, SD = 0.52$) and receives the lowest accuracy ($M = 86.2\%, SD = 5.05\%$). While the *half-body+AR* tutorials achieves relatively shorter completion time ($M = 1.02, SD = 0.22$) than *non-avatar-AR* ($M = 1.15, SD = 0.38$) and *full-body+AR* ($M = 1.07, SD = 0.25$) ($Z = -2.19, p < 0.028$ and $Z = -2.750, p < 0.006$ respectively). On the other hand, the accuracy of *non-avatar-AR* ($M = 95.5\%, SD = 2.53\%$), *half-body+AR* ($M = 94.9\%, SD = 3.17\%$), and *full-body+AR* ($M = 93.7\%, SD = 3.36\%$) are roughly the same (pairwise $p > 0.05$).

***Body-coordinated* steps performance**. The *video* tutorials received the worst performance in both completion time ($M = 1.62, SD = 0.58$) and accuracy ($M = 77.5\%, SD =$

7.4%). Users with *half-body+AR* tutor option (normalized completion time: $M = 1.00, SD = 0.22$, accuracy: $M = 97.2\%, SD = 2.61\%$) are able to perform significantly faster ($Z = -2.19, p < 0.05$) with less mistakes ($p < 0.05$) than the ones with *non-avatar-AR* tutorials (normalized completion time: $M = 1.13, SD = 0.37$, accuracy: $M = 92.4\%, SD = 5.54\%$), which indicates the strengths of the avatar in demonstrating bodily movement. Between the two avatar options, the *full-body+AR* (normalized completion time: $M = 1.05, SD = 0.25$, accuracy: $M = 96.2\%, SD = 2.77\%$) has longer completion time ($Z = -2.75, p = 0.005$) and roughly the same accuracy ($p > 0.05$) compared with *half-body+AR* tutor option. For the 2D 'curve' operation, the *half-body+AR* tutor achieves the shortest average MHD ($M = 42.5$ *cm*$, SD = 13.7$ *cm*), followed by *non-avatar-AR* ($M = 46.1$ *cm*$, SD = 21.0$ *cm*) and *full-body+AR* ($M = 46.6$ *cm*$, SD = 20.0$ *cm*), while the *video* tutor option achieves the longest average MHD ($M = 50.7$*cm*$, SD = 18.25$*cm*). Yet the Friedman test ($\chi^2(3) = 5.81, p = 0.121$) does not reveal any significant difference among the four options.

6.4.2 Subjective Rating and User Preference

Figure 6.8 shows the user experience subjective ratings with the 7-point Likert questionnaire. To reveal the differences among the tutor options, we conduct a Friedman test followed by a Wilcoxon signed-rank test on each of the eight questions individually. We first look into the effectiveness of AR in the tutorial systems by comparing *video* and *non-avatar-AR*. The result shows that the latter option achieves significantly higher ratings ($p < 0.0005$) in 'Understanding', 'Accuracy', 'Confidence' and 'Satisfaction', while no significant difference is found in 'Attention' ($p = 0.22$), 'Spatial' ($p = 0.124$), 'Bodily' ($p = 0.167$) and 'Social' ($p = 0.355$). Secondly, we examine whether the existence of an avatar affects the user experience. The result reveals that in all eight ratings, the *non-avatar-AR* option has significant lower scores ($p < 0.05$) than either the *half-body+AR* or the *full-body+AR*. Thus, we believe the overall machine task user experience is improved by the presence of an avatar.

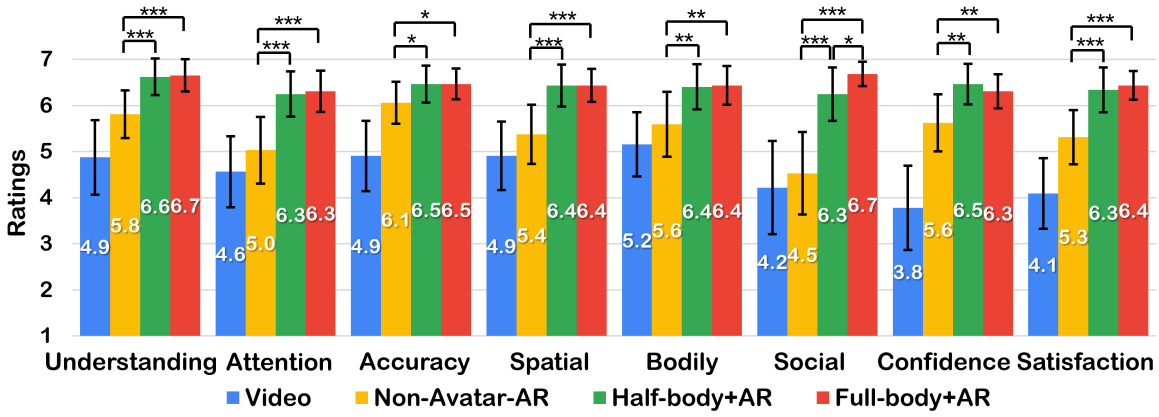


Fig. 6.8. User experience ratings. (***=p<.0005, **=p<.005, *=p<.05. If not specified, *** between the video options and other three tutor options.) Error bars represent standard deviations.

Finally, we inspect how the visual guidance level of avatar influences the user experience by comparing the ratings between *half-body+AR* and *full-body+AR*. We find no significant difference between the two tutor options except the 'Social' rating where *full-body+AR* is slightly higher ($p = 0.05$) than *half-body+AR*.

Figure 6.9 illustrates the user preference survey results for the overall machine task tutoring experience, regarding the *local*, *spatial*, and *body-coordinated* interactions, respectively. For each type of interaction, users are allowed to choose one or more tutor options as their favorite. Overall, the *half-body+AR* is most preferred tutor option (21 out of 39), followed by the *full-body+AR* (13 out of 39) and the *non-avatar-AR* (5 out of 39), while no users choose the *video* as their favorite tutor option. In terms of the *local* interactions, the *non-avatar-AR* option is the most favored (21 out of 45). The *half-body+AR* and the *full-body+AR* are tied in the second place (12 out of 45). Again, no users choose the *video*. In terms of the *spatial* interactions, the *half-body+AR* tutor option comes to the first place (20 out of 42). The *full-body+AR* option takes the second place with 14 out of 42 users, while *non-avatar-AR* (5 out of 42) and *video* (3 out of 42) are less preferred. As

for the *Body-coordinate* interactions, *full-body+AR* is the most popular choice (20 out of 45), that is followed closely by *half-body+AR* (18 out of 45). Only a few users choose the *non-avatar-AR* (4 out of 45) and the *video* (3 out of 45) tutor option as their favorite.

6.4.3 Result Summary and Analysis

We now summarize the main results and present explanatory analysis using our observation during the study as well as findings that come out from the interview.

Overall favorite: half-body vs. full-body

The ratings for the two proposed avatar tutor options are found to be similar across all categories, and are significantly better than the *non-avatar-AR* and the *video* options (Figure 6.8). Interestingly, when the users' are asked to pick their favorite tutor option overall, the *half-body* has a clear preference edge over the *full-body* (21 vs 13). From the post-study interview, we find that many users believe these two tutor options are functionally equal, while the *half-body* has less occlusion to the users' views. "*I think the half-body is the best because it can show me where to go and what to do without blocking too much of my sight (P7).*" The increased visual access to the physical machine in *half-body* as compared to *full-body* may also have resulted in lower mental effort ("*The full-body avatar tutor shows too many things, and sometimes is too exhausting for me (P8)*"), and less attention distraction ("*A full-body human is not necessary, its arms and legs distract my attention from the machine, half-body is cleaner and less distracting (P16)*"). This is also reflected in the objective performance result (Figure 6.7) where the *half-body* achieves similar accuracy with less time, compared with the *full-body*. The above discussion also explains the preference result for the *spatial* interactions, where the *half-body* is enough for instructing spatial navigation and target finding, with a cleaner observing view.

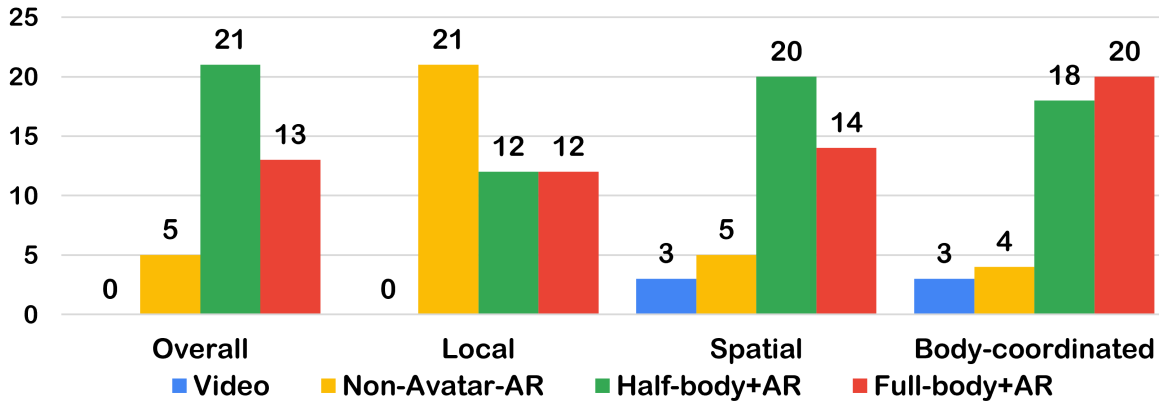


Fig. 6.9. User preference result.

As observed from the user preference result, the additional body features become helpful in the *body-coordinated* interactions. The users feel that the added limb representations, especially the arms, do provide a better understanding of the two-hand coordinated tasks. "*For the bodily tasks, I prefer full-body, because full-body gives me more spatial and embodied evidence. Just a hand is not enough sometimes. I feel like needing the extra arm information (P15).*" Another preference of the *full-body* over the *half-body* is on social presence, which is also reflected by the 'Social' subjective rating results. According to the feedback from the interview, the *full-body* is better than the *half-body* at representing a human tutor, which makes it a more friendly, believable, and reliable option. " *The full-body feels more like a human, like a real tutor and more friendly. In comparison, the half-body is obviously a robotic indicator (P26).*"

Local favorite: non-avatar-AR

Despite the lack of spatial and bodily presentations, the *non-avatar-AR* is selected as the favorite tutor option for the *local* interactions. This is because the *local* interactions does not require substantial spatial and body movements. The attention of the user does

not need to be directed effectively to locate the target interface, nor does a particular body gesture play an important role in terms of interaction execution. Therefore, the presence of a human avatar in *local* interactions does not provide extra benefits in most cases. "*I don't think avatar is useful for local tasks because AR instruction is enough, and I usually cannot see the avatar anyways because I am standing inside the avatar (P5).*" Several participants report that the avatar encumbers and slows down their actions, which is consistent with our finding that the *non-avatar-AR* is fastest for *local* tasks with equal accuracy (Figure 6.7).

Least preferred: video

It is clear that the *video* is the least popular tutor option among the four. According to our observation, the main problem for *video* tutoring is caused by the two separate dimensions of the tutoring and the application: users have to receive the instruction from the digital world, interpret it into his/her physical world, and then apply it to the corresponding machine interfaces. This translation gap causes many problems such as distracted attention, fractured spatial mapping, high mental effort from memorization, and a non-optimized observation perspective. "*My attention is changing from video content to reality all the time, and sometimes I need to think very hard to interpret what it means in the video (P1).*" However, the *video* still demonstrates some values from the user preference survey on *spatial* and *body-coordinated* categories. Some users have mentioned that the *video* option can occasionally be more expressive than the other options. "*To me, video is the best for spatial and embodied task, because you can best understand the body motion right away. The avatar is not obvious because I was standing inside the avatar, and I cannot notice the avatar (P2).*"

6.5 Discussion

Here we discuss the primary results of the study and contrast them with prior works. We also provide design recommendations and insights for future AR tutoring systems.

6.5.1 Benefits of Avatars for Tutoring

Our first research question focuses on understanding whether the proposed AR avatar presentations improve the machine task tutoring experience, and how they do so. Our findings indicate that the AR avatars receive significantly more positive feedback than the non-avatar and video tutor options, and provide several insights into why. We summarize these reasons below, and distill our findings into design recommendations for avatar-based tutoring systems.

Spatial Attention Allocation. When trying to follow a comprehensive machine task tutorial, one of the major challenges for a user is to know where to pay attention, especially during constant spatial movements that easily cause disorientation. Compared to the non-avatar tutor presence, the additional avatar provides more noticeable in-situ visual hints to guide the user's attention. "*Sometimes I cannot find the machine target until the human avatar moves over there and starts reaching out his hand. (P7)*" This result is aligned with prior works on mixed reality assistant, where a remote expert provides *live* guidance for a local learner via the presence of an AR avatar [135–137]. The avatars in these works are usually controlled by a remote human, thus are capable of communicating and responding to the user's action adaptively. However, when applying the avatars to *recorded* tutoring with no remote human involvement, we recommend that future system provide a feedback mechanism for user-responsive tutoring. For example, the *recorded* tutor should act only when the learner is paying attention to it [207]. The above finding inspires us to design attention indicators in the future to explicitly guide users' attention and reduce mental effort.

Bodily Movement Expression. The digital tutor is capable of intuitively expressing the human body in the context of a physical interactive target. This enables the users to understand the movement accurately and anticipate the tutor's actions, especially for the tasks involving head-hand-body coordination. "*Seeing the human tutor move in the space allows me to predict where he is going and what he is going to do next, and it prepares me to get ready for the task in advance (P20).*" This advantage of avatars is consistent with prior research on body movement training, such as the YouMove system [130]. While prior works on body movement training [130, 141, 208] have primarily focused on physical tasks being performed by humans in isolation, we show that these advantages have benefits for tasks where spatial and temporal connections must be made between virtual avatars and physical objects (in our case, the machine being manipulated).

Higher Social Presence. Due to the human-like visual presence, user feedback suggests that following the avatar resembles the tutoring experience of following a human teacher. This improves the user's confidence, which leads to a higher efficiency in tutoring information transfer. "*The human avatar is easy to follow, as long as you do that, you feel confident, and nothing is going to be wrong, it gives me less mental pressure (P24).*" Mini-Me [135] has a similar finding that the avatar option in their study yields a higher aggregated social presence and awareness score for task transfer collaboration than the non-avatar options, resulting in the reduced mental effort and improved performance.

6.5.2 Adaptive Tutoring

In the second research question, we explore how to optimize the tutoring experience in a comprehensive machine task scenario involving multiple interaction categories. In this paper, we study four tutor options with gradually increased guidance visualization level, aiming to provide insights for the ideal design. Our results do not show any one presentation method to be clearly superior, but rather reveal a number of considerations that must be

balanced to create a good avatar-based tutoring experience. In particular, we discuss three factors in the sections that follow: level of visual detail, tutor following paradigm, and playback progress. As some of these factors reflect individual preferences, we believe there is an opportunity for adaptive and personalized tutoring experiences that dynamically tailor the experience to individual users.

Level of Visual Detail. According to our results, users acknowledge the usefulness of avatars, but more visual details also cause confusion and occlusion of the physical world. Therefore the level of visual guidance details should be contextually adaptive to the interaction type and task difficulty. This also explains why the users prefer half-body avatar for the overall machine task and non-avatar for the *local* interactions. "*It should not display the whole action animation, only the key part should be played; otherwise, it is too distracting (P1).*" This finding aligns with a study conducted by Lindlbauer D, et al. [207] where they find that the dynamically adjusted AR contents lead to less distraction and higher performance. Further, the tutor's presence should also adapt to the learner's reliance on instructions. We have observed that some users were able to complete most steps fast and accurately by only following *non-avatar-AR* option, while some others needed *full-body+AR* instead.

Tutor Following Paradigm. Currently, the position of the AR tutor is connected to the physical interactive machine, and it is up to the learners to decide where to observe the tutor and how to follow it. We noticed that some users prefer to stand inside the tutor avatar and follow it in synchronization to achieve higher efficiency and accuracy. "*I like to stand inside of the avatar and follow its movement, makes me feel confident about my accuracy (P29).*" This paradigm has been acknowledged and adopted by an arm motion training system where the virtual guiding arms are superimposed in the user's egocentric AR view [131]. On the other hand, some users prefer to stand on the side of the avatar tutor because they consider it uncomfortable to collide into a virtual humanoid. "*I do not feel*

like standing inside the avatar, because it feels like a person and I don't want to crash into him (P6)." This can be explained by a study conducted by Kim et al. [209] on the physical presence of the avatar. They find that the conflicts between humans and virtual avatars reduce the sense of co-presence and should be avoided if possible. The above findings demonstrate the importance of providing spatially aware instructional contents based on the user's physical location and observation perspective.

Playback Progress. In our study setup, the playback speed of each tutorial step is fixed and determined by the authored demonstration. Also, the progress of the user is manually monitored and manipulated by the researchers. If a learner misses critical information of the step, he/she has to wait for the step animation to play again, leading to low learning efficiency. Based on our observation and feedback, we believe the future systems should incorporate an adaptive tutorial playback speed based on users' innate capability and task difficulty. This finding is aligned with the study done by Rajinder et. al [210], where they study projected visualizations for hand movement guidance and find that dynamically adjusted guiding speed has the potential of improving training efficiency. Further, an adaptive playback helps the users to preview the tutor's intent, such as using slow-motion to forecast the avatar's actions. "*I need to know what the avatar is about to do and where to pay attention, sometimes the avatar makes a sudden turn, and it's very hard to notice (P23).*"

6.6 Study Limitations

The hardware and performance of the AR headset may have influenced participants' experience in several ways. Though we used state-of-the-art technology (VR headset with a front-attached stereo camera to achieve see-through AR with a high-resolution and full eye-sight field of view), several participants reported minor motion sickness, and the inability for the cameras to fully simulate stereo vision that caused some participants to bump into the machine while trying to manipulate it. As the headset was tethered to a computer, cords

sometimes needed to be untangled, which may have slowed spatial and bodily movements as compared to operating the machine free from tethers. While we acknowledge that the above conditions may have impacted the user experience, they were consistent across the three tutor options we tested.

To conduct our study, we have created an interactive mockup machine capable of all three types of steps. Therefore the study result that we collected is largely based on the users' interaction performance on this mockup machine. Even though we designed the mockup machine based on the real-world machine interfaces and interactions, it is still a testbed. The mockup machine can only represent a portion of the real-world machine tasks, including the three interaction steps. We would like to acknowledge explicitly that the result of this study should be used mainly as a comparative reference among the four tutor options as an elicitation or informative study for future tutoring system design.

6.7 Conclusion

In this paper, we have presented an exploratory study of augmented reality presence for machine task tutoring system design. We created an AR-based embodied authoring system capable of creating tutorials with four types of tutor options: *video*, *non-avatar-AR*, *half-body+AR*, and *full-body+AR*. In order to conduct our study, we have designed and fabricated a mockup machine capable of supporting *local, spatial*, and *body-coordinated* human-machine interactions. We invited 32 users, each for a 4-session study experiencing all four tutor options for comparative feedback. From the quantitative and qualitative results of the study, we have discussed and summarized the design recommendations for future tutoring systems. These design insights form an important stepping stone to help the future researchers create a comprehensive and intelligent machine task tutoring system, that will enable fluid machine task skill transfer and empower an efficient, flexible, and productive workforce.

7. SUMMARY OF CONTRIBUTION

As is stated earlier, the theme of this Ph.D. thesis is to explore novel interactions for human smart-things interaction through the approach of augmented reality system framework design. In our completed research work, we have developed a modular robotic-IoT system for mixed reality interaction with content creation, editing, and animation authoring (Ani-Bot); an workflow design for human authored Robot-IoT collaborative task planning powered by one single AR-SLAM device (V.Ra); a time-space editor for Human-Robot Collaborative task authoring through AR embodied interaction (GhostAR); and an exploratory study on augmented reality tutor presence for machine task tutoring (AvaTutAR-study). In this chapter of the thesis, we are going to summarize the core contributions of this thesis, the knowledge generated from each project, and how they are connected to the central theme.

7.1 Thesis central theme

During my Ph.D., I have been working on 8 paper projects, and they can be connected to the central theme and visualized in this research road-map (Figure 7.1). The core contribution of this thesis is to explore the strength of AR and design system workflows around it, in order to answer questions like: When and where should we use AR, and how do we make the best of it in future smart-thing applications? And the answer can be summarized into four keyword: visual, spatial, contextual, and embodied.

From my thesis work, it has been demonstrated that AR is ultimately a visual interface, it has the advantage of traditional GUI, but not bond by a stationary monitor on a flat screen. Instead, the visual content can be projected anywhere, in any 3D format. Therefore AR is a

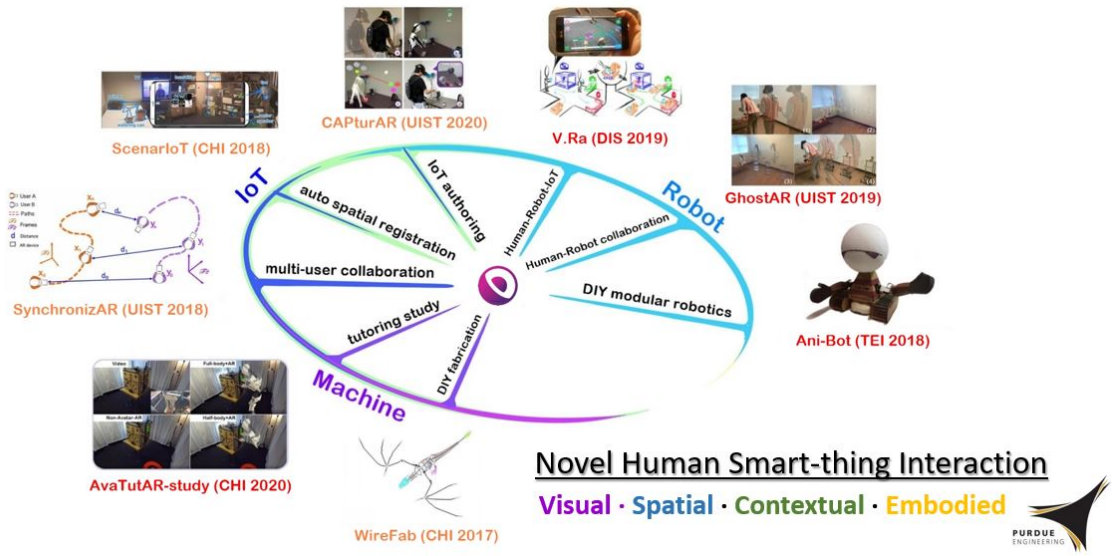


Fig. 7.1. PhD research road-map including lead-author and co-author paper projects.

spatially situated visual interface, which is especially useful for enhancing mobile targets, like robots, and interface crowded objects, like complex machines.

Furthermore, AR automatically include the nearby reality into the visual interface as contextual references, this makes it easy for users to interact their object targets with the surrounding environment. Besides, user can explicitly create virtual reference of their target objects in the AR view to leave an editable trace, allowing for sophisticated logic programming in both the temporal and spatial domain.

Finally, since AR is not limited by a physical screen, it supports embodied interaction which takes natural inputs from user's body gestures. This allows the users to transfer human intent to smart-thing behavior more easily by intuitively acting it out, hence programming-by-demonstration. Plus, the embodied demonstration can also be used for authoring human into a time-space visual reference for a variety of applications.

In my opinion, the development of AR for real-world application is still at a very early stage, if you ask a random person what is AR, he/she is probably gonna tell you: Pokémon GO.

There is no doubt that we should all thank Pokémon GO for preaching the idea of AR to the general public, however, is it the true use of AR though? I think not.

Because literally speaking, augmented reality, in order for AR to make sense, you first need to have a reality that is worth the augmentation, placing a virtual Pokémon in my camera view is not augmenting anything. In other words, we need to have a direct feed forward and feedback loop connection between the AR interface and the target object, in order to truly bring out the strength of AR.

And that is what I believe the core contribution of this thesis, to explore the use of AR, its application scenario, and its target use cases. During this process, we define the problems in AR applications and we try to find solutions via system framework design. That aims at delivering a visual, spatial, contextual, and embodied AR experience.

7.2 Virtual-physical diagram of each work

Ani-Bot's virtual-physical diagram can be illustrated in Figure 7.2. The workflow starts from user tangibly put together a DIY robot from a variety of provided modules, or disassembling the existing modular robotics into the parts in an iterative manner. Meanwhile, in the augmented reality domain, a virtual representation of the physical robot is automatically generated based on the assembly configuration. The AR UI are reflecting the indivisual property of each module, in a spatially-situatted manner. Because of assembly awareness, we can also take advantage of the contextual information and visual cues for assembly guidance and virtual-physical tryout. Furthermore, we also support embodied animation authoring through programming-by-demonstration. In this way, user can control the physical robot buy directly interacting with its corresponding virtual representation in-situ, hence

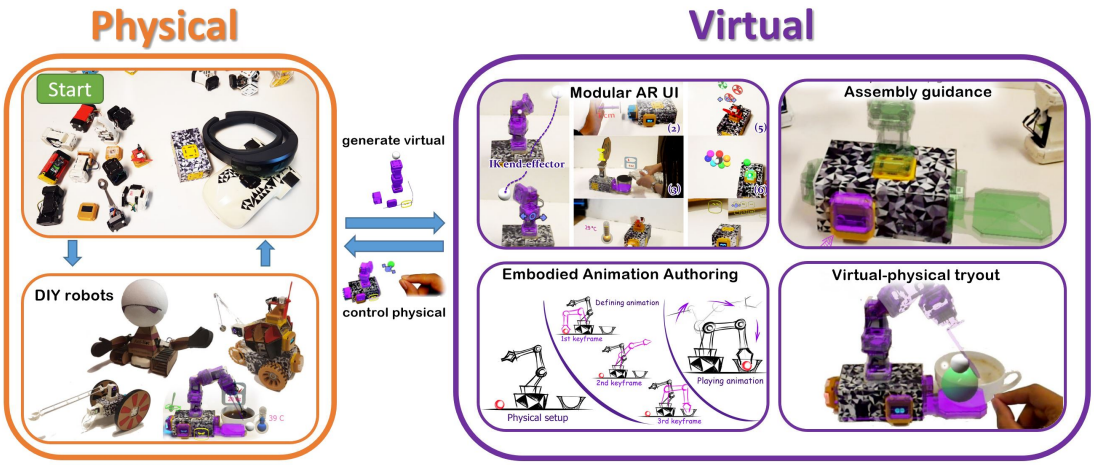


Fig. 7.2. Ani-Bot virtual-physical diagram.

completing the feed-forward and feedback loop and give users an interactive virtual-physical experience.

In this project, we have discovered a new method to apply augmented reality in modular robotics application. The key lies within the achievement of assembly awareness of the system, and also the balance between individual modular interface and combined functionality control. Each individual module in the assembly should be visually accessible if needed, however, users would probably prefer a higher level control in most of the occasions to achieve higher efficiency and better understanding of the constructed robot overall. The limitation of the current system is the way of achieving the auto virtual reconstruction from the physical via physical-electrical connection, and the alignment of the virtual onto the physical via image marker. This indicates future system should investigate more into marker-less tracking with computer-vision based virtual reconstruction, thus supporting a larger variety of DIY creating with more robust spatial tracking.

V.Ra's virtual-physical diagram can be illustrated in Figure 7.3. The workflow starts from the user embodied demonstrating the task by spatially walking around with the mo-

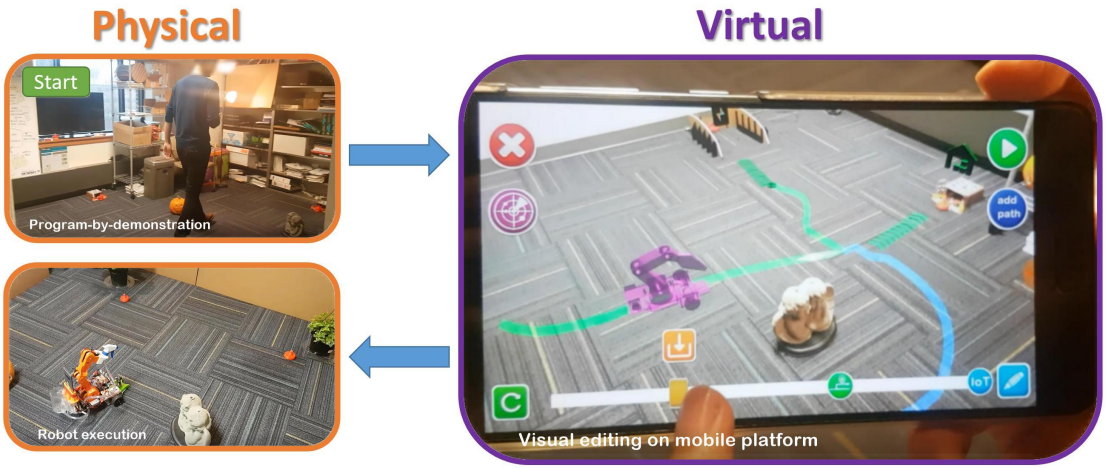


Fig. 7.3. V.Ra virtual-physical diagram.

bile SLAM-AR device, for the robot path planning and IoT interaction tasks. The user then visually edit the tasks and preview the planned task with virtual simulations in the contextually-aware AR interface. When finished, the device is placed onto the physical robot to begin the task execution in a what-you-do-is-what-robot-does manner

In this project, we have created a new concept of utilizing the mobile AR-SLAM device for both task planning and robot executing, in a lightweight self-dependent way. The work opens new ways to look at a mobile AR-SLAM device and its potential applications when cohesively interacting with humans, IoTs, and robots. V.Ra creates a new way to inspire future researchers to rethink the balance between human authoring and robot automation in the coming era of Internet-of-robotic-things. Since the current system is limited to one AR-SLAM device and one robot only, we encourage future researchers to focus on exploring a collaboration scenario including multiple robots and human users.

GhostAR's virtual-physical diagram can be illustrated in Figure 7.4. The workflow starts from user physically demonstrating the human part of the collaboration. Then our system visualizes the demonstration into an animating AR ghost. Using this ghost as a

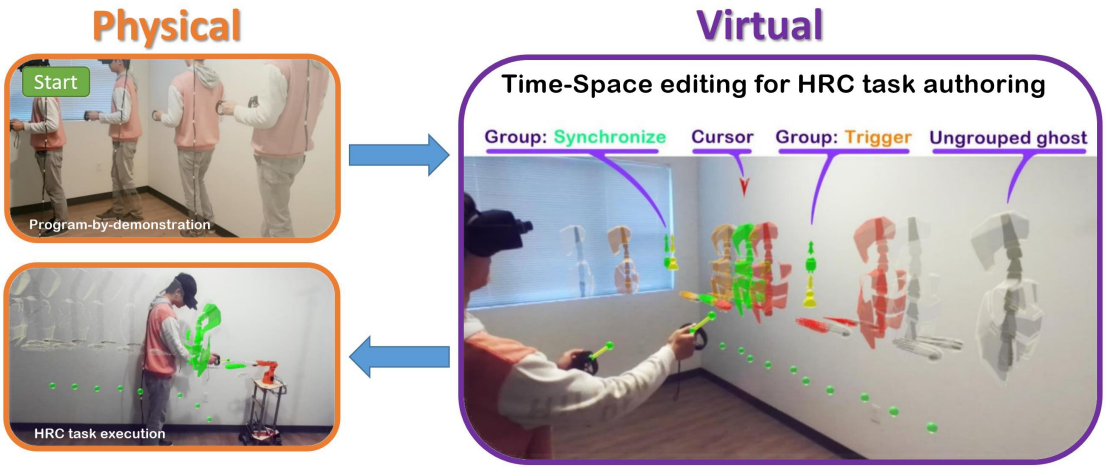


Fig. 7.4. GhostAR virtual-physical diagram.

time-space contextual reference, user can perform visual editing and author the human-robot collaboration tasks. When acting out the collaboration with real robot, the user simply need to repeat the demonstration physically, in order to complete the task.

In this project, we have invented a new way of creating HRC tasks by fully utilizing embodied demonstration, contextual reference, and AR visualization of time-space editing. This work truly emphasizes the strength of augmented reality and cohesively incorporated them into one system fluidly. GhostAR points out a promising direction for future researchers in terms of human smart-thing interaction, that leverage the advantage from both human's innate bodily movement and an AR interface's contextual visualization in the temporal and spatial domain.

AvaTutAR's virtual-physical diagram can be illustrated in Figure 7.5. The workflow starts from the expert physically demonstrating the task via embodied interaction with the machines. The demonstration of the task will be recorded and visualize into various types of tutor presence. In a remote AR view, the learners can physically follow the AR virtual tutor and repeat the machine task in-situ, with the real machine scenarios as contextual references.

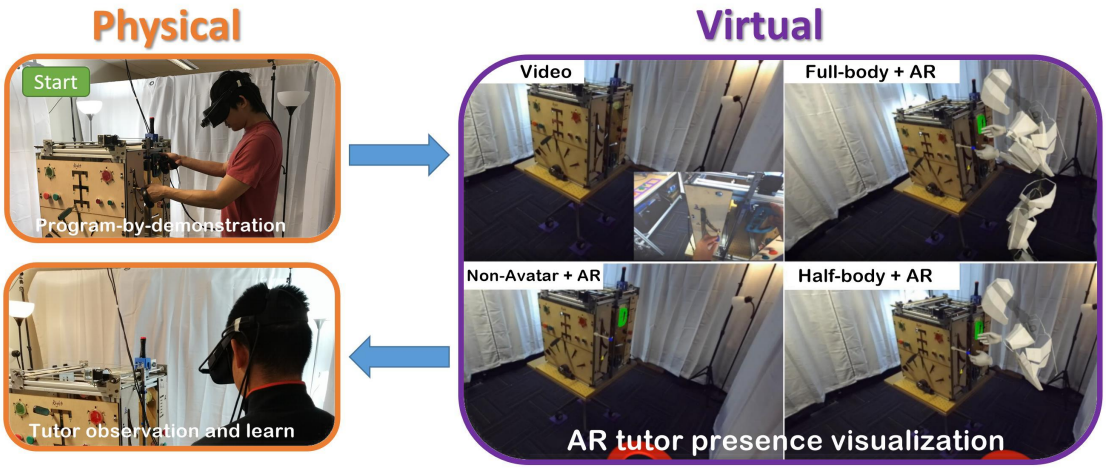


Fig. 7.5. AvaTutAR virtual-physical diagram.

In this project, we have learned the importance of customization in tutorial system design. Which points future researchers to the direction of adaptive tutoring. A future tutorial system should be constantly aware of the user's status: how well has he been doing so far? what is he doing currently? what is he going to do next? The system should know these answers by actively monitoring the user and the surrounding physical environment, in order to adjust the visual presence of the virtual tutor for optimized tutoring experiences.

8. FUTURE VISION

In this thesis, we have presented our research work in terms of exploring novel human smart-thing interaction enabled by the newly emerging Augmented Reality technology. I personally truly believe that AR is the future of digital interface, especially personal display. In the last part of this thesis, I will humbly present my own vision for the future of AR.

8.1 AR ecosystem: wearable, handheld, and environmental

In terms of devices that enable AR experience, currently there are three major categories: wearable (Head-Mounted-Device, etc), handheld (Mobile smartphone, etc), and environmental (AR projection on canvas with body tracking, etc). Each of these three categories has its own strength and unique characteristics.

Wearable is good at unintrusive and immersive experience. With the future development of hardware form factor, the AR enabling wearables will merge within user's daily life, like a pair of glasses, a wrist watch, or a piece of clothing. The key about wearable AR devices lie within the input and output modality. The devices need to effectively sense the status of the user and the surrounding environment in order to provide in-situ decision making assistance. Furthermore, it should support all natural human interactions, including but not limited to: touch, speech, gaze, and gesture. With the fast development of sensing technology, battery, and cloud computing, we can expect future AR wearable devices to shrink the size merge into user's daily life unnoticed.

Handheld is a different modality of AR interaction experience because it is an external device. The unique feature about a handheld device is that it is not only a display for AR content but also an input controller for the user. By interacting with the device, users can

achieve more specific and complex interaction than wearables. Also the mobility of the wearable device can be utilized, because it can be detached from the human user, it can serve as an independent operating mobile device for a variety of applications, such as robotics navigation, remote sensing, terrain exploration, etc.

Environmental AR is able to achieve the so-called 'naked-eye AR', meaning the recipients of the AR content do not need to be wearing or holding any special devices. This type of AR can be deployed in public areas for a large audience, or it can be applied into a controlled environment, such as smart car interior or personal theater for tailored experience. This type of system should also support natural interactions such as speech and gesture for users to interact with the AR content. Different from wearable and handheld AR platform, the environmental is focused on stationary and large-scale AR projection.

I believe the future of AR should consist of all these three types of AR platform, and they should be connected into one ecosystem to share and update data, in order to become smart and adaptive. Therefore, a major challenge for the future of AR is to create this AR ecosystem or operating system. With it, we will be able to apply AR to augment every aspects of our life.

8.2 Real-World AR: Stepping from local to global augmentation

Augmented reality is to use superimposing virtual content to augment the corresponding physical object. Therefore, before a physical object can be augmented, a virtual representation of the object needs to be created in the first place. For example, in our past work that uses AR to control and program robots and IoTs, we would need to create the virtual 3D model of the robots and IoT devices and use them as the interaction media in the AR environment.

Creating the virtual representation of the physical object is doable in a local environment with a finite number of known devices. However, if we want to extend the usage of AR in the

future and apply it on the global level, we will need to have a new way to achieve digitization of the physical world. Manually modeling the world is certainly not an viable option, since it is extremely time and resource consuming, plus our world is always changing, and it will require constant remodeling to keep up with the update.

Therefore we will need to find a way to digitize the physical world in-situ, and update on-the-fly, while also establishing the connection between the virtual content and its physical target. I believe this will be another major challenge we will face in the future of AR, and I think we might find inspirations from the emerging technology of crowd-sourcing, cloud computing, and 3D scanning and reconstruction. For example, I envision a possible workflow as follows: In the future, everyone is wearing lightweight AR glasses, these glasses are constantly scanning the surrounding environment. Due to the advanced on-board sensing capability, they can see well beyond what the users are seeing. All of the AR glasses are connected to the cloud ecosystem, which processes the uploaded sensing data and use computer vision and machine learning algorithm to reconstruct the scanned reality in 3D virtually. Since each user also upload the geometric position and orientation information, the cloud server is able to weave a global 3D map from all the collected information with real-time update capability. This 3D map is then accessible to the user on demand to achieve real-world in-situ augmentation.

REFERENCES

REFERENCES

[1] Y. Cao, Z. Xu, T. Glenn, K. Huo, and K. Ramani, "Ani-bot: A modular robotics system supporting creation, tweaking, and usage with mixed-reality interactions," in *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 2018, pp. 419–428.

[2] Y. Cao, Z. Xu, F. Li, W. Zhong, K. Huo, and K. Ramani, "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*, 2019, pp. 1059–1070.

[3] Y. Cao, T. Wang, X. Qian, P. S. Rao, M. Wadhawan, K. Huo, and K. Ramani, "Ghostar: A time-space editor for embodied authoring of human-robot collaborative task with augmented reality," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM, 2019, pp. 521–534.

[4] Y. Cao, X. Qian, T. Wang, R. Lee, K. Huo, and K. Ramani, "An exploratory study of augmented reality presence for tutoring machine tasks," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–13.

[5] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai, "Scratch: Programming for all," *Commun. ACM*, vol. 52, no. 11, pp. 60–67, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1592761.1592779

[6] B. Ullmer and H. Ishii, "Emerging frameworks for tangible user interfaces," *IBM Systems Journal*, vol. 39, no. 3.4, pp. 915–931, 2000.

[7] ——, "The metadesk: Models and prototypes for tangible user interfaces," in *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '97. New York, NY, USA: ACM, 1997, pp. 223–232. [Online]. Available: http://doi.acm.org/10.1145/263407.263551

[8] H. Ishii, "The tangible user interface and its evolution," *Commun. ACM*, vol. 51, no. 6, pp. 32–36, Jun. 2008. [Online]. Available: http://doi.acm.org/10.1145/1349026.1349034

[9] E. Sharlin, B. Watson, Y. Kitamura, F. Kishino, and Y. Itoh, "On tangible user interfaces, humans and spatiality," *Personal and Ubiquitous Computing*, vol. 8, no. 5, pp. 338–346, 2004. [Online]. Available: http://dx.doi.org/10.1007/s00779-004-0296-5

[10] M. Billinghurst, A. Clark, G. Lee *et al.*, "A survey of augmented reality," *Foundations and Trends® in Human–Computer Interaction*, vol. 8, no. 2-3, pp. 73–272, 2015.

[11] ——, "A survey of augmented reality," *Foundations and Trends® in Human–Computer Interaction*, vol. 8, no. 2-3, pp. 73–272, 2015.

[12] H. S. Raffle, A. J. Parkes, and H. Ishii, "Topobo: a constructive assembly system with kinetic memory," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2004, pp. 647–654.

[13] "Vex robotics," 2017, https://www.vexrobotics.com/.

[14] O. Shaer and E. Hornecker, "Tangible user interfaces: past, present, and future directions," *Foundations and Trends in Human-Computer Interaction*, vol. 3, no. 1–2, pp. 1–137, 2010.

[15] "Cubelets," 2017, http://www.modrobotics.com/cubelets/cubelets-twenty/.

[16] "Littlebits," 2017, http://littlebits.cc/.

[17] "Code-a-pillar," 2017, http://fisher-price.mattel.com/shop/en-us/fp/think-learn-code-a-pillar-starter-gift-set-fgn83.

[18] H. Oh and M. D. Gross, "Cube-in: A learning kit for physical computing basics," in *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 2015, pp. 383–386.

[19] M. Kazemitabaar, J. McPeak, A. Jiao, L. He, T. Outing, and J. E. Froehlich, "Makerwear: A tangible approach to interactive wearable creation for children," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2017, pp. 133–145.

[20] "Lego mindstorm," 2017, https://www.lego.com/en-us/mindstorms.

[21] "Tinkerbots," 2017, https://www.tinkerbots.com/.

[22] "Cagebot," 2017, https://www.cagebot.com/shop/.

[23] "Moss," 2017, https://theory.stanford.edu/~aiken/moss/.

[24] A. Sullivan, M. Elkin, and M. U. Bers, "Kibo robot demo: Engaging young children in programming and engineering," in *Proceedings of the 14th international conference on interaction design and children*. ACM, 2015, pp. 418–421.

[25] J. S. Seehra, A. Verma, K. Peppler, and K. Ramani, "Handimate: Create and animate using everyday objects as material," in *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 2015, pp. 117–124.

[26] S. Gupta, S. Jang, and K. Ramani, "Puppetx: a framework for gestural interactions with user constructed playthings," in *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*. ACM, 2014, pp. 73–80.

[27] R. Slyper, G. Hoffman, and A. Shamir, "Mirror puppeteering: Animating toy robots in front of a webcam," in *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 2015, pp. 241–248.

[28] Y. Kitamura, Y. Itoh, T. Masaki, and F. Kishino, "Activecube: a bi-directional user interface using cubes," in *Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on*, vol. 1. IEEE, 2000, pp. 99–102.

[29] M. P. Weller, E. Y.-L. Do, and M. D. Gross, "Posey: instrumenting a poseable hub and strut construction toy," in *Proceedings of the 2nd international conference on Tangible and embedded interaction*. ACM, 2008, pp. 39–46.

[30] D. Anderson, J. L. Frankel, J. Marks, A. Agarwala, P. Beardsley, J. Hodgins, D. Leigh, K. Ryall, E. Sullivan, and J. S. Yedidia, "Tangible interaction+ graphical interpretation: a new approach to 3d modeling," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 393–402.

[31] A. Miller, B. White, E. Charbonneau, Z. Kanzler, and J. J. LaViola Jr, "Interactive 3d model acquisition and tracking of building block structures," *IEEE transactions on visualization and computer graphics*, vol. 18, no. 4, pp. 651–659, 2012.

[32] S. J. Henderson and S. K. Feiner, "Augmented reality in the psychomotor phase of a procedural task," in *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*. IEEE, 2011, pp. 191–200.

[33] S. Makris, G. Pintzos, L. Rentzos, and G. Chryssolouris, "Assembly support using ar technology based on automatic sequence generation," *CIRP Annals-Manufacturing Technology*, vol. 62, no. 1, pp. 9–12, 2013.

[34] M. Yuan, S. Ong, and A. Nee, "Augmented reality for assembly guidance using a virtual interactive tool," *International Journal of Production Research*, vol. 46, no. 7, pp. 1745–1767, 2008.

[35] X. Wang, S. Ong, and A. Y.-C. Nee, "Multi-modal augmented-reality assembly guidance based on bare-hand interface," *Advanced Engineering Informatics*, vol. 30, no. 3, pp. 406–421, 2016.

[36] Z. Wang, S. Ong, and A. Nee, "Augmented reality aided interactive manual assembly design," *The International Journal of Advanced Manufacturing Technology*, vol. 69, no. 5-8, pp. 1311–1321, 2013.

[37] J. Zhang, S. Ong, and A. Nee, "Rfid-assisted assembly guidance system in an augmented reality environment," *International Journal of Production Research*, vol. 49, no. 13, pp. 3919–3938, 2011.

[38] L. X. Ng, S. Ong, and A. Nee, "Arcade: a simple and fast augmented reality computer-aided design environment using everyday objects," 2010.

[39] S. Ong, Y. Pang, and A. Nee, "Augmented reality aided assembly design and planning," *CIRP Annals-Manufacturing Technology*, vol. 56, no. 1, pp. 49–52, 2007.

[40] L. X. Ng, S. Oon, S. K. Ong, and A. Y. Nee, "Garde: a gesture-based augmented reality design evaluation system," *International Journal on Interactive Design and Manufacturing*, vol. 5, no. 2, pp. 85–94, 2011.

[41] S. Hashimoto, A. Ishida, M. Inami, and T. Igarashi, "Touchme: An augmented reality based remote robot manipulation," in *21st Int. Conf. on Artificial Reality and Telexistence, Proc. of ICAT2011*, 2011.

[42] S. Kasahara, R. Niiyama, V. Heun, and H. Ishii, "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, 2013, pp. 223–228.

[43] J. A. Frank, M. Moorhead, and V. Kapila, "Realizing mixed-reality environments with tablets for intuitive human-robot collaboration for object manipulation tasks," in *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*. IEEE, 2016, pp. 302–307.

[44] K. Ishii, Y. Takeoka, M. Inami, and T. Igarashi, "Drag-and-drop interface for registration-free object delivery," in *RO-MAN, 2010 IEEE*. IEEE, 2010, pp. 228–233.

[45] R. Fung, S. Hashimoto, M. Inami, and T. Igarashi, "An augmented reality system for teaching sequential tasks to a household robot," in *RO-MAN, 2011 IEEE*. IEEE, 2011, pp. 282–287.

[46] D. Leithinger, S. Follmer, A. Olwal, S. Luescher, A. Hogge, J. Lee, and H. Ishii, "Sublimate: state-changing virtual and physical rendering to augment interaction with shape displays," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2013, pp. 1441–1450.

[47] V. Heun, J. Hobin, and P. Maes, "Reality editor: Programming smarter objects," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 2013, pp. 307–310.

[48] V. Heun, S. Kasahara, and P. Maes, "Smarter objects: using ar technology to program physical objects and their interactions," in *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2013, pp. 961–966.

[49] P. Schoessler, D. Windham, D. Leithinger, S. Follmer, and H. Ishii, "Kinetic blocks: Actuated constructive assembly for interaction and display," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 2015, pp. 341–349.

[50] M. Sugimoto, T. Fujita, H. Mi, and A. Krzywinski, "Robotable2: a novel programming environment using physical robots on a tabletop platform," in *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*. ACM, 2011, p. 10.

[51] N. Linder and P. Maes, "Luminar: portable robotic augmented reality interface design and prototype," in *Adjunct proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, 2010, pp. 395–396.

[52] H. Fang, S. Ong, and A. Nee, "A novel augmented reality-based interface for robot path planning," *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 8, no. 1, pp. 33–42, 2014.

[53] J. W. S. Chong, S. Ong, A. Y. Nee, and K. Youcef-Youmi, "Robot programming using augmented reality: An interactive method for planning collision-free paths," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 3, pp. 689–701, 2009.

[54] J. Lambrecht, M. Kleinsorge, M. Rosenstrauch, and J. Krüger, "Spatial programming for industrial robots through task demonstration," *International Journal of Advanced Robotic Systems*, vol. 10, no. 5, p. 254, 2013.

[55] H. Fang, S. Ong, and A. Nee, "Interactive robot trajectory planning and simulation using augmented reality," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 227–237, 2012.

[56] C. C. Kemp, A. Edsinger, and E. Torres-Jara, "Challenges for robot manipulation in human environments [grand challenges of robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 20–29, 2007.

[57] K. Liu, D. Sakamoto, M. Inami, and T. Igarashi, "Roboshop: multi-layered sketching interface for robot housework assignment and management," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 647–656.

[58] R. Fung, S. Hashimoto, M. Inami, and T. Igarashi, "An augmented reality system for teaching sequential tasks to a household robot," in *RO-MAN, 2011 IEEE*. IEEE, 2011, pp. 282–287.

[59] D. Sakamoto, Y. Sugiura, M. Inami, and T. Igarashi, "Graphical instruction for home robots," *Computer*, vol. 49, no. 7, pp. 20–25, 2016.

[60] S. Zhao, K. Nakamura, K. Ishii, and T. Igarashi, "Magic cards: a paper tag interface for implicit robot control," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009, pp. 173–182.

[61] S. Magnenat, M. Ben-Ari, S. Klinger, and R. W. Sumner, "Enhancing robot programming with visual feedback and augmented reality," in *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*. ACM, 2015, pp. 153–158.

[62] J. Gimeno, P. Morillo, J. M. Orduña, and M. Fernández, "A new ar authoring tool using depth maps for industrial procedures," *Computers in Industry*, vol. 64, no. 9, pp. 1263–1271, 2013.

[63] A. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*. IEEE, 2012, pp. 1–8.

[64] M. Ciocarlie, K. Hsiao, A. Leeper, and D. Gossow, "Mobile manipulation through an assistive home robot," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 5313–5320.

[65] H. Nguyen, M. Ciocarlie, K. Hsiao, and C. C. Kemp, "Ros commander (rosco): Behavior creation for home robots," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 467–474.

[66] S. Mayer, M. Schalch, M. George, and G. Sörös, "Device recognition for intuitive interaction with the web of things," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 2013, pp. 239–242.

[67] S. Lin, H. F. Cheng, W. Li, Z. Huang, P. Hui, and C. Peylo, "Ubii: Physical world interaction through augmented reality," *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 872–885, 2017.

[68] "Arcore," 2017, https://developers.google.com/ar/.

[69] "Arkit," 2017, https://developer.apple.com/arkit/.

[70] K. Huo, Y. Cao, S. H. Yoon, Z. Xu, G. Chen, and K. Ramani, "Scenariot: Spatially mapping smart things within augmented reality scenes," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018, p. 219.

[71] B. Ens, F. Anderson, T. Grossman, M. Annett, P. Irani, and G. Fitzmaurice, "Ivy: Exploring spatially situated visual programming for authoring and understanding intelligent environments," in *Proceedings of the 43rd Graphics Interface Conference*. Canadian Human-Computer Communications Society, 2017, pp. 156–162.

[72] B. Ens and P. Irani, "Spatial analytic interfaces: Spatial user interfaces for in situ visual analytics," *IEEE computer graphics and applications*, vol. 37, no. 2, pp. 66–79, 2017.

[73] A. G. Millard, R. Redpath, A. Jewers, C. Arndt, R. Joyce, J. A. Hilder, L. J. McDaid, and D. M. Halliday, "Ardebug: an augmented reality tool for analysing and debugging swarm robotic systems," *Frontiers Robotics AI*, 2018.

[74] F. Ghiringhelli, J. Guzzi, G. A. Di Caro, V. Caglioti, L. M. Gambardella, and A. Giusti, "Interactive augmented reality for understanding and analyzing multi-robot systems," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 1195–1201.

[75] P. Birkenkampf, D. Leidner, and C. Borst, "A knowledge-driven shared autonomy human-robot interface for tablet computers," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. IEEE, 2014, pp. 152–159.

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

[77] A. Thomaz, G. Hoffman, M. Cakmak *et al.*, "Computational human-robot interaction," *Foundations and Trends® in Robotics*, vol. 4, no. 2-3, pp. 105–223, 2016.

[78] S. Nikolaidis, J. Forlizzi, D. Hsu, J. Shah, and S. Srinivasa, "Mathematical models of adaptation in human-robot collaboration," *arXiv preprint arXiv:1707.02586*, 2017.

[79] S. Nikolaidis, S. Nath, A. D. Procaccia, and S. Srinivasa, "Game-theoretic modeling of human adaptation in human-robot collaboration," in *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*. IEEE, 2017, pp. 323–331.

[80] S. Nikolaidis, Y. X. Zhu, D. Hsu, and S. Srinivasa, "Human-robot mutual adaptation in shared autonomy," in *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*. IEEE, 2017, pp. 294–302.

[81] J. Shah, J. Wiken, B. Williams, and C. Breazeal, "Improved human-robot team performance using chaski, a human-inspired plan execution system," in *Proceedings of the 6th international conference on Human-robot interaction*. ACM, 2011, pp. 29–36.

[82] S. Nikolaidis and J. Shah, "Human-robot cross-training: computational formulation, modeling and evaluation of a human team training strategy," in *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*. IEEE Press, 2013, pp. 33–40.

[83] S. Nikolaidis, R. Ramakrishnan, K. Gu, and J. Shah, "Efficient model learning from joint-action demonstrations for human-robot collaborative tasks," in *Proceedings of the tenth annual ACM/IEEE international conference on human-robot interaction*. ACM, 2015, pp. 189–196.

[84] H. B. Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 2831–2837.

[85] M. Ewerton, G. Neumann, R. Lioutikov, H. B. Amor, J. Peters, and G. Maeda, "Learning multiple collaborative tasks with a mixture of interaction primitives," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1535–1542.

[86] G. J. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer, and J. Peters, "Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks," *Autonomous Robots*, vol. 41, no. 3, pp. 593–612, 2017.

[87] G. Maeda, M. Ewerton, G. Neumann, R. Lioutikov, and J. Peters, "Phase estimation for fast action recognition and trajectory generation in human–robot collaboration," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1579–1594, 2017.

[88] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," *Springer handbook of robotics*, pp. 1371–1394, 2008.

[89] P. Evrard, E. Gribovskaya, S. Calinon, A. Billard, and A. Kheddar, "Teaching physical collaborative tasks: Object-lifting case study with a humanoid," in *2009 9th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2009, pp. 399–404.

[90] A. P. Shon, K. Grochow, and R. P. Rao, "Robotic imitation from human motion capture using gaussian processes," in *5th IEEE-RAS International Conference on Humanoid Robots, 2005.* IEEE, 2005, pp. 129–134.

[91] L. Peternel, T. Petrič, E. Oztop, and J. Babič, "Teaching robots to cooperate with humans in dynamic manipulation tasks based on multi-modal human-in-the-loop approach," *Autonomous robots*, vol. 36, no. 1-2, pp. 123–136, 2014.

[92] S. Niekum, S. Osentoski, G. Konidaris, and A. G. Barto, "Learning and generalization of complex tasks from unstructured demonstrations," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5239–5246.

[93] C. Daniel, G. Neumann, and J. Peters, "Learning concurrent motor skills in versatile solution spaces," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 3591–3597.

[94] M. Pardowitz, S. Knoop, R. Dillmann, and R. D. Zollner, "Incremental learning of tasks from user demonstrations, past experiences, and vocal comments," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 322–332, 2007.

[95] J. Saunders, C. L. Nehaniv, and K. Dautenhahn, "Teaching robots by moulding behavior and scaffolding the environment," in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*. ACM, 2006, pp. 118–125.

[96] D. Vogt, S. Stepputtis, S. Grehl, B. Jung, and H. B. Amor, "A system for learning continuous human-robot interactions from human-human demonstrations," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2882–2889.

[97] D. Vogt, S. Stepputtis, B. Jung, and H. B. Amor, "One-shot learning of human–robot handovers with triadic interaction meshes," *Autonomous Robots*, vol. 42, no. 5, pp. 1053–1065, 2018.

[98] H. Hedayati, M. Walker, and D. Szafir, "Improving collocated robot teleoperation with augmented reality," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2018, pp. 78–86.

[99] B. Larochelle and G.-J. M. Kruijff, "Multi-view operator control unit to improve situation awareness in usar missions," in *RO-MAN, 2012 IEEE*. IEEE, 2012, pp. 1103–1108.

[100] S. Magnenat, M. Ben-Ari, S. Klinger, and R. W. Sumner, "Enhancing robot programming with visual feedback and augmented reality," in *Proceedings of the 2015 ACM conference on innovation and technology in computer science education*. ACM, 2015, pp. 153–158.

[101] J. A. Frank, S. P. 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.

[102] F. Ghiringhelli, J. Guzzi, G. A. Di Caro, V. Caglioti, L. M. Gambardella, and A. Giusti, "Interactive augmented reality for understanding and analyzing multi-robot systems," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1195–1201.

[103] A. G. Millard, R. Redpath, A. Jewers, C. Arndt, R. Joyce, J. A. Hilder, L. J. McDaid, and D. M. Halliday, "Ardebug: an augmented reality tool for analysing and debugging swarm robotic systems," *Frontiers Robotics AI*, 2018.

[104] R. K. Ganesan, "Mediating human-robot collaboration through mixed reality cues," Ph.D. dissertation, Arizona State University, 2017.

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

[106] Y. S. Sefidgar, T. Weng, H. Harvey, S. Elliott, and M. Cakmak, "Robotist: Interactive situated tangible robot programming," in *Proceedings of the Symposium on Spatial User Interaction*. ACM, 2018, pp. 141–149.

[107] E. Rosen, D. Whitney, E. Phillips, G. Chien, J. Tompkin, G. Konidaris, and S. Tellex, "Communicating robot arm motion intent through mixed reality head-mounted displays," *arXiv preprint arXiv:1708.03655*, 2017.

[108] R. S. Andersen, O. Madsen, T. B. Moeslund, and H. B. Amor, "Projecting robot intentions into human environments," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2016, pp. 294–301.

[109] R. T. Chadalavada, H. Andreasson, R. Krug, and A. J. Lilienthal, "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, 2015, pp. 1–6.

[110] D. Lindlbauer and A. D. Wilson, "Remixed reality: Manipulating space and time in augmented reality," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018, p. 129.

[111] K. Huo, T. Wang, L. Paredes, A. M. Villanueva, Y. Cao, and K. Ramani, "Synchronizar: Instant synchronization for spontaneous and spatial collaborations in augmented reality," in *The 31st Annual ACM Symposium on User Interface Software and Technology*. ACM, 2018, pp. 19–30.

[112] H. Xia, S. Herscher, K. Perlin, and D. Wigdor, "Spacetime: Enabling fluid individual and collaborative editing in virtual reality," in *The 31st Annual ACM Symposium on User Interface Software and Technology*. ACM, 2018, pp. 853–866.

[113] T. Wójcicki, "Supporting the diagnostics and the maintenance of technical devices with augmented reality," *Diagnostyka*, vol. 15, no. 1, pp. 43–47, 2014.

[114] armedia, "I-mechanic, the ar app that turns yourself into a mechanic," 2019, retrieved September 1, 2019 from http://www.armedia.it/i-mechanic.

[115] J. Zhu, S.-K. Ong, and A. Y. Nee, "A context-aware augmented reality assisted maintenance system," *International Journal of Computer Integrated Manufacturing*, vol. 28, no. 2, pp. 213–225, 2015.

[116] A. Peniche, C. Diaz, H. Trefftz, and G. Paramo, "Combining virtual and augmented reality to improve the mechanical assembly training process in manufacturing," in *American Conference on applied mathematics*, 2012, pp. 292–297.

[117] E. Schoop, M. Nguyen, D. Lim, V. Savage, S. Follmer, and B. Hartmann, "Drill sergeant: Supporting physical construction projects through an ecosystem of augmented tools," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2016, pp. 1607–1614.

[118] J.-R. Chardonnet, G. Fromentin, and J. Outeiro, "Augmented reality as an aid for the use of machine tools," *Res. & Sci. Today*, vol. 13, p. 25, 2017.

[119] M. Funk, "Augmented reality at the workplace: a context-aware assistive system using in-situ projection," 2016.

[120] S. J. Henderson and S. K. Feiner, "Augmented reality in the psychomotor phase of a procedural task," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2011, pp. 191–200.

[121] S. Kim, G. Lee, W. Huang, H. Kim, W. Woo, and M. Billinghurst, "Evaluating the combination of visual communication cues for hmd-based mixed reality remote collaboration," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 2019, p. 173.

[122] G.-S. Jo, K.-J. Oh, I. Ha, K.-S. Lee, M.-D. Hong, U. Neumann, and S. You, "A unified framework for augmented reality and knowledge-based systems in maintaining aircraft," in *Twenty-Sixth IAAI Conference*, 2014.

[123] G. Westerfield, "Intelligent augmented reality training for assembly and maintenance," 2012.

[124] A. Monroy Reyes, O. O. Vergara Villegas, E. Miranda Bojórquez, V. G. Cruz Sánchez, and M. Nandayapa, "A mobile augmented reality system to support machinery operations in scholar environments," *Computer Applications in Engineering Education*, vol. 24, no. 6, pp. 967–981, 2016.

[125] Z. Zhu, V. Branzoi, M. Wolverton, G. Murray, N. Vitovitch, L. Yarnall, G. Acharya, S. Samarasekera, and R. Kumar, "Ar-mentor: Augmented reality based mentoring system," in *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2014, pp. 17–22.

[126] S. Webel, U. Bockholt, T. Engelke, N. Gavish, M. Olbrich, and C. Preusche, "An augmented reality training platform for assembly and maintenance skills," *Robotics and Autonomous Systems*, vol. 61, no. 4, pp. 398–403, 2013.

[127] S. Ong and Z. Wang, "Augmented assembly technologies based on 3d bare-hand interaction," *CIRP annals*, vol. 60, no. 1, pp. 1–4, 2011.

[128] F. De Crescenzio, M. Fantini, F. Persiani, L. Di Stefano, P. Azzari, and S. Salti, "Augmented reality for aircraft maintenance training and operations support," *IEEE Computer Graphics and Applications*, vol. 31, no. 1, pp. 96–101, 2010.

[129] P. T. Chua, R. Crivella, B. Daly, N. Hu, R. Schaaf, D. Ventura, T. Camill, J. Hodgins, and R. Pausch, "Training for physical tasks in virtual environments: Tai chi," in *IEEE Virtual Reality, 2003. Proceedings.* IEEE, 2003, pp. 87–94.

[130] F. Anderson, T. Grossman, J. Matejka, and G. Fitzmaurice, "Youmove: enhancing movement training with an augmented reality mirror," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 2013, pp. 311–320.

[131] P.-H. Han, K.-W. Chen, C.-H. Hsieh, Y.-J. Huang, and Y.-P. Hung, "Ar-arm: Augmented visualization for guiding arm movement in the first-person perspective," in *Proceedings of the 7th Augmented Human International Conference 2016*. ACM, 2016, p. 31.

[132] T. N. Hoang, M. Reinoso, F. Vetere, and E. Tanin, "Onebody: remote posture guidance system using first person view in virtual environment," in *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*. ACM, 2016, p. 25.

[133] P.-H. Han, J.-W. Lin, C.-H. Hsieh, J.-H. Hsu, and Y.-P. Hung, "target: limbs movement guidance for learning physical activities with a video see-through head-mounted display," in *ACM SIGGRAPH 2018 Posters*. ACM, 2018, p. 26.

[134] S. Yan, G. Ding, Z. Guan, N. Sun, H. Li, and L. Zhang, "Outsideme: Augmenting dancer's external self-image by using a mixed reality system," in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2015, pp. 965–970.

[135] T. Piumsomboon, G. A. Lee, J. D. Hart, B. Ens, R. W. Lindeman, B. H. Thomas, and M. Billinghurst, "Mini-me: an adaptive avatar for mixed reality remote collaboration," in *Proceedings of the 2018 CHI conference on human factors in computing systems*. ACM, 2018, p. 46.

[136] T. Piumsomboon, G. A. Lee, A. Irlitti, B. Ens, B. H. Thomas, and M. Billinghurst, "On the shoulder of the giant: A multi-scale mixed reality collaboration with 360 video sharing and tangible interaction," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 2019, p. 228.

[137] B. Thoravi Kumaravel, F. Anderson, G. Fitzmaurice, B. Hartmann, and T. Grossman, "Loki: Facilitating remote instruction of physical tasks using bi-directional mixed-reality telepresence," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM, 2019, pp. 161–174.

[138] Y. Kim and S.-H. Bae, "Sketchingwithhands: 3d sketching handheld products with first-person hand posture," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 2016, pp. 797–808.

[139] B. Lee, M. Cho, J. Min, and D. Saakes, "Posing and acting as input for personalizing furniture," in *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*. ACM, 2016, p. 44.

[140] Y. Zhang, T. Han, Z. Ren, N. Umetani, X. Tong, Y. Liu, T. Shiratori, and X. Cao, "Bodyavatar: creating freeform 3d avatars using first-person body gestures," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 2013, pp. 387–396.

[141] P.-Y. P. Chi, D. Vogel, M. Dontcheva, W. Li, and B. Hartmann, "Authoring illustrations of human movements by iterative physical demonstration," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 2016, pp. 809–820.

[142] D. Eckhoff, C. Sandor, C. Lins, U. Eck, D. Kalkofen, and A. Hein, "Tutar: augmented reality tutorials for hands-only procedures," in *Proceedings of the 16th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*. ACM, 2018, p. 8.

[143] A. Gupta, D. Fox, B. Curless, and M. Cohen, "Duplotrack: a real-time system for authoring and guiding duplo block assembly," in *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 2012, pp. 389–402.

[144] C. Barnes, D. E. Jacobs, J. Sanders, D. B. Goldman, S. Rusinkiewicz, A. Finkelstein, and M. Agrawala, "Video puppetry: a performative interface for cutout animation," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 5. ACM, 2008, p. 124.

[145] R. Held, A. Gupta, B. Curless, and M. Agrawala, "3d puppetry: a kinect-based interface for 3d animation." in *UIST*. Citeseer, 2012, pp. 423–434.

[146] A. Gupta, M. Agrawala, B. Curless, and M. Cohen, "Motionmontage: A system to annotate and combine motion takes for 3d animations," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2014, pp. 2017–2026.

[147] N. Saquib, R. H. Kazi, L.-Y. Wei, and W. Li, "Interactive body-driven graphics for augmented video performance," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 2019, p. 622.

[148] D. Vogt, S. Stepputtis, S. Grehl, B. Jung, and H. B. Amor, "A system for learning continuous human-robot interactions from human-human demonstrations," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2882–2889.

[149] H. B. Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 2831–2837.

[150] D. Porfirio, E. Fisher, A. Sauppé, A. Albarghouthi, and B. Mutlu, "Bodystorming human-robot interactions," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 2019.

[151] H. Ishii, "The tangible user interface and its evolution," *Communications of the ACM*, vol. 51, no. 6, pp. 32–36, 2008.

[152] J. Leong, F. Perteneder, H.-C. Jetter, and M. Haller, "What a life!: Building a framework for constructive assemblies." in *Tangible and Embedded Interaction*, 2017, pp. 57–66.

[153] "Hololens," 2017, https://www.microsoft.com/en-us/hololens.

[154] "Vuforia," 2017, https://www.vuforia.com/.

[155] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[156] P. P. Ray, "Internet of robotic things: Concept, technologies, and challenges." *IEEE Access*, vol. 4, pp. 9489–9500, 2016.

[157] O. Vermesan, A. Bröring, E. Tragos, M. Serrano, D. Bacciu, S. Chessa, C. Gallicchio, A. Micheli, M. Dragone, A. Saffiotti *et al.*, "Internet of robotic things: converging sensing/actuating, hypoconnectivity, artificial intelligence and iot platforms," 2017.

[158] "Ifttt," 2018, https://ifttt.com/.

[159] Y. Chuang, L.-L. Chen, and Y. Liu, "Design vocabulary for human–iot systems communication," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018, p. 274.

[160] "Roomba," 2018, https://en.wikipedia.org/wiki/Roomba.

[161] F. Ingrand and M. Ghallab, "Deliberation for autonomous robots: A survey," *Artificial Intelligence*, vol. 247, pp. 10–44, 2017.

[162] S. Alexandrova, Z. Tatlock, and M. Cakmak, "Roboflow: A flow-based visual programming language for mobile manipulation tasks," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5537–5544.

[163] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Transactions on automation science and engineering*, vol. 12, no. 2, pp. 398–409, 2015.

[164] A. Bauer, D. Wollherr, and M. Buss, "Human–robot collaboration: a survey," *International Journal of Humanoid Robotics*, vol. 5, no. 01, pp. 47–66, 2008.

[165] T. Ende, S. Haddadin, S. Parusel, T. Wüsthoff, M. Hassenzahl, and A. Albu-Schäffer, "A human-centered approach to robot gesture based communication within collaborative working processes," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 3367–3374.

[166] R. Stiefelhagen, C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel, "Natural human-robot interaction using speech, head pose and gestures," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2422–2427.

[167] S. Chernova and A. L. Thomaz, "Robot learning from human teachers," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 8, no. 3, pp. 1–121, 2014.

[168] G. Klein, P. J. Feltovich, J. M. Bradshaw, and D. D. Woods, "Common ground and coordination in joint activity," *Organizational simulation*, vol. 53, pp. 139–184, 2005.

[169] "Xsense," 2019, https://www.xsens.com/tags/motion-capture/.

[170] "Optitrack," 2019, https://optitrack.com/.

[171] "Oculus," 2019, https://www.oculus.com/.

[172] A. Jackson, B. D. Northcutt, and G. Sukthankar, "The benefits of teaching robots using vr demonstrations," in *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2018, pp. 129–130.

[173] R. Billon, A. Nedelec, and J. Tisseau, "Gesture recognition in flow based on pca analysis using multiagent system," in *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*. ACM, 2008, pp. 139–146.

[174] S. Sempena, N. U. Maulidevi, and P. R. Aryan, "Human action recognition using dynamic time warping," in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*. IEEE, 2011, pp. 1–5.

[175] H. Sakoe, S. Chiba, A. Waibel, and K. Lee, "Dynamic programming algorithm optimization for spoken word recognition," *Readings in speech recognition*, vol. 159, p. 224, 1990.

[176] "Robot operating system," 2019, http://www.ros.org/.

[177] "Razebo simulator," 2019, http://gazebosim.org/.

[178] "Rossharp," 2019, https://github.com/siemens/ros-sharp.

[179] M. Loskyll, I. Heck, J. Schlick, and M. Schwarz, "Context-based orchestration for control of resource-efficient manufacturing processes," *Future Internet*, vol. 4, no. 3, pp. 737–761, 2012.

[180] D. Gorecky, M. Schmitt, M. Loskyll, and D. Zühlke, "Human-machine-interaction in the industry 4.0 era," in *2014 12th IEEE international conference on industrial informatics (INDIN)*. Ieee, 2014, pp. 289–294.

[181] D. Gorecky, M. Khamis, and K. Mura, "Introduction and establishment of virtual training in the factory of the future," *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 1, pp. 182–190, 2017.

[182] L. Suchman, *Human-machine reconfigurations: Plans and situated actions*. Cambridge University Press, 2007.

[183] J.-M. Hoc, "Towards a cognitive approach to human–machine cooperation in dynamic situations," *International journal of human-computer studies*, vol. 54, no. 4, pp. 509–540, 2001.

[184] B. Thoravi Kumaravel, C. Nguyen, S. DiVerdi, and B. Hartmann, "Tutorivr: A video-based tutorial system for design applications in virtual reality," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 2019, p. 284.

[185] M. Goto, Y. Uematsu, H. Saito, S. Senda, and A. Iketani, "Task support system by displaying instructional video onto ar workspace," in *2010 IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2010, pp. 83–90.

[186] P.-Y. Chi, J. Liu, J. Linder, M. Dontcheva, W. Li, and B. Hartmann, "Democut: generating concise instructional videos for physical demonstrations," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 2013, pp. 141–150.

[187] P.-Y. Chi, S. Ahn, A. Ren, M. Dontcheva, W. Li, and B. Hartmann, "Mixt: automatic generation of step-by-step mixed media tutorials," in *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 2012, pp. 93–102.

[188] J. Kim, P. T. Nguyen, S. Weir, P. J. Guo, R. C. Miller, and K. Z. Gajos, "Crowdsourcing step-by-step information extraction to enhance existing how-to videos," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2014, pp. 4017–4026.

[189] G. Dini and M. Dalle Mura, "Application of augmented reality techniques in through-life engineering services," *Procedia Cirp*, vol. 38, pp. 14–23, 2015.

[190] VRChat, "Create and play in virtual worlds," 2019, retrieved September 1, 2019 from https://www.vrchat.net/.

[191] D. Preuveneers, "The ghosthands ux: telementoring with hands-on augmented reality instruction," in *Workshop Proceedings of the 11th International Conference on Intelligent Environments*, vol. 19. IOS Press, 2015, p. 236.

[192] T. Pejsa, J. Kantor, H. Benko, E. Ofek, and A. Wilson, "Room2room: Enabling life-size telepresence in a projected augmented reality environment," in *Proceedings of the 19th ACM conference on computer-supported cooperative work & social computing*. ACM, 2016, pp. 1716–1725.

[193] A. J. Fairchild, S. P. Campion, A. S. García, R. Wolff, T. Fernando, and D. J. Roberts, "A mixed reality telepresence system for collaborative space operation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 814–827, 2016.

[194] D. Damen, T. Leelasawassuk, and W. Mayol-Cuevas, "You-do, i-learn: Egocentric unsupervised discovery of objects and their modes of interaction towards video-based guidance," *Computer Vision and Image Understanding*, vol. 149, pp. 98–112, 2016.

[195] G. Sun, P. Muneesawang, M. Kyan, H. Li, L. Zhong, N. Dong, B. Elder, and L. Guan, "An advanced computational intelligence system for training of ballet dance in a cave virtual reality environment," in *2014 IEEE International Symposium on Multimedia*. IEEE, 2014, pp. 159–166.

[196] M. Oshita, T. Inao, T. Mukai, and S. Kuriyama, "Self-training system for tennis shots with motion feature assessment and visualization," in *2018 International Conference on Cyberworlds (CW)*. IEEE, 2018, pp. 82–89.

[197] P.-H. Han, Y.-S. Chen, Y. Zhong, H.-L. Wang, and Y.-P. Hung, "My tai-chi coaches: an augmented-learning tool for practicing tai-chi chuan," in *Proceedings of the 8th Augmented Human International Conference*. ACM, 2017, p. 25.

[198] Stereolabs, "Zed mini stereo camera - stereolabs," 2019, retrieved September 1, 2019 from https://www.stereolabs.com/zed-mini/.

[199] Noitom, "Perception neuron by noitom," 2019, retrieved September 1, 2019 from https://neuronmocap.com.

[200] Unity3D, "Unity3d," 2017, retrieved September 1, 2017 from https://unity3d.com/.

[201] FinalIK, "Finalik," 2019, retrieved September 1, 2019 from https://assetstore.unity.com/packages/tools/animation/final-ik-14290.

[202] J. Sauro, "10 things to know about the single ease question (seq)," 2012, retrieved September 1, 2019 from https://measuringu.com/seq10/.

[203] J. Sauro and J. S. Dumas, "Comparison of three one-question, post-task usability questionnaires," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2009, pp. 1599–1608.

[204] A. Bangor, P. T. Kortum, and J. T. Miller, "An empirical evaluation of the system usability scale," *Intl. Journal of Human–Computer Interaction*, vol. 24, no. 6, pp. 574–594, 2008.

[205] F. Biocca, C. Harms, and J. Gregg, "The networked minds measure of social presence: Pilot test of the factor structure and concurrent validity," *4th annual International Workshop on Presence, Philadelphia*, 01 2001.

[206] M.-P. Dubuisson and A. K. Jain, "A modified hausdorff distance for object matching," in *Proceedings of 12th international conference on pattern recognition*, vol. 1. IEEE, 1994, pp. 566–568.

[207] D. Lindlbauer, A. M. Feit, and O. Hilliges, "Context-aware online adaptation of mixed reality interfaces," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM, 2019, pp. 147–160.

[208] C. Murlowski, F. Daiber, F. Kosmalla, and A. Krüger, "Slackliner 2.0: Real-time training assistance through life-size feedback," in *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 2019, p. INT012.

[209] K. Kim, G. Bruder, and G. Welch, "Exploring the effects of observed physicality conflicts on real-virtual human interaction in augmented reality," in *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*. ACM, 2017, p. 31.

[210] R. Sodhi, H. Benko, and A. Wilson, "Lightguide: projected visualizations for hand movement guidance," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 179–188.