EXPLORATION, STUDY AND APPLICATION OF SPATIALLY AWARE INTERACTIONS SUPPORTING PERVASIVE AUGMENTED REALITY

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Ke Huo

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Karthik Ramani, Chair

School of Mechanical Engineering

Dr. Alex Quinn

School of Electrical and Computer Engineering

Dr. David Cappelleri

School of Mechanical Engineering

Dr. Song Zhang

School of Mechanical Engineering

Approved by:

Dr. Jay Gore

Head of the School Graduate Program

Dedicated to my family and friends.

ACKNOWLEDGMENTS

I would like to first express my gratitude to my advisor Professor Karthik Ramani for encouraging, supporting, and helping me to identify the fascinating research questions and investigate the answers scientifically. In particular, his generosity allowed me for plenty of time for a deep exploration on the HCI related areas before I settled down the research topic. Further, I would like to extend my gratitude to Professor Song Zhang, Professor David Cappelleri, and Professor Alex Quinn for serving as my doctoral committee member and the inspiring discussions on my research.

I sincerely thank all members (past and present) of the C-Design lab for their supports and helps on my work and my life. I am very grateful to all of my collaborators enriching my research experience within a wide scope and contributing to my thesis. Especially, the feedback from Dr. Yunbo Zhang, Dr. Vinayak, Dr. Wei Gao, Dr. Sang Ho Yoon has played important roles in shaping my research projects. I thank Bill, Tarun, Diogo for working together as TAs for ME 553.

Finally, I would like to thank my family for their unconditional love. To my parents, whose persistent supports encourage me through my entire graduate stage. To my brother and sister, for being taking care of our parents while I study abroad and their helps on my life.

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

ABSTRACT

Huo, Ke Ph.D., Purdue University, May 2019. Exploration, Study and Application of Spatially Aware Interactions Supporting Pervasive Augmented Reality. Major Professor: Karthik Ramani, School of Mechanical Engineering.

With rapidly increasing mobile computing devices and high speed networks, large amounts of digital information and intelligence from the surrounding environment have been introduced into our everyday life. However, much of the context and content is in textual and in 2D. To access the digital contents spontaneously, augmented reality (AR) has become a promising surrogate to bridge the physical with the digital world. Thanks to the vast improvement to the personal computing devices, AR technologies are emerging in realistic scenarios. Commercially available software development kits (SDKs) and hardware platforms have started to expose AR applications to a large population.

In a broader level, this thesis focuses on investigating suitable interactions metaphors for the evolving AR. In particular, this work leverages the spatial awareness in AR environment to enable spatially-aware interactions. This work explores (i) spatial inputs around AR devices using the local spatial relationship between the AR devices and the scene, (ii) spatial interactions within the surrounding environment exploiting the global spatial relationship among multiple users as well as between the users and the environment. In this work, I mainly study four spatially-aware AR interactions: (i) 3D tangible interactions by directly mapping input to the continuous and discrete volume around the device, (ii) 2D touch input in 3D context by projecting the screen input to the real world, (iii) location aware interactions which use the locations of the real/virtual objects in the AR scene as spatial references, and (iv) collaborative interactions referring to a commonly shared AR scene. This work further develop the enabling techniques including a magnetic sensing based 3D tracking of tangible devices relative to a handheld AR device, a projection based 3D sketching technique for in-situ AR contents creation, a localization method for spatially

mapping the smart devices into the AR scene, and a registration approach for resolving the transformations between multiple SLAM AR devices. Moreover, I build systems towards allowing pervasive AR experiences. Primarily, I develop applications for increasing the flexibility of AR contents manipulation, creation and authoring, intuitively interacting with the smart environment, and spontaneously collaborating within a co-located AR scene.

The main body of the research has contributed to multiple on-going collaborative projects. I briefly discuss the key results and visions from these projects including (i) autonomous robotic exploration and mapping of smart environment where the spatial relationship between the robot and the smart devices is resolved, and (ii) human-robot-interaction in AR where the spatial intelligence can be seamlessly exchanged between the human and the robot. Further, I suggest future research projects leveraging three critical features from AR, namely situatedness, mobility, and the capability to support spatial collaborations.

1. INTRODUCTION

The rapidly increasing mobile computing devices and the high speed networks lead to an increasing accessibility of digital information and intelligence which are usually hard to be detected or generated directly by human being's sense and mind. As augmented Reality (AR) 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, the 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 retains interactive in real-time [14].

To meet these requirements, the AR community has been primarily concerned with the enabling techniques such as tracking, calibration, registering, rendering, etc until recently [200]. Because of the vast improvement in the personal computing devices, now the AR technologies are beginning to become applicable to realistic scenarios. In particular, some commercially available products have started to expose AR applications to a large population within a wide range of contexts. For example, several popular software development kits (SDK) including Vuforia [180], Wikitude [190], ARCore [63], and ARKit [7] are available for developing AR applications for moderate mobile computing devices. Further, more hardware products such as Zenfone [13], Hololens [122], and Magic Leap [115] have began to empower AR system with better environmental perception by embedding a commodity depth sensor. Thus, with the past accumulated enabling techniques, studies towards AR interactions become significant for enhancing AR experiences. The works presented in this dissertation are largely driven by the following two broad research questions.

- What are the interaction metaphors suitable for the evolving AR, and the corresponding enabling techniques?

- How can we create seamless AR experience across different use scenarios?

1.1 Interactions for Augmented Reality

Various interaction techniques have been utilized for different AR applications. In 2D user interface design, the traditional "windows, icons, menus, pointer" (WIMP) metaphor has been proven to be capable of supporting complex interactions. Therefore adoption of such techniques and using them in AR applications is straightforward and helpful for the transition phase between WIMP and post-WIMP. Further, tangible user interfaces (TUI) focus on using graspable physical objects for manipulating the digital contents [92]. The TUI metaphor leverages the innate ability of users to manipulate the objects in the physical world thus increasing the intuitiveness of the interface. Particularly, since AR aims to blend the virtual contents and the physical world, tangible AR interfaces can further contribute to a seamless AR experience. More importantly, a mobile AR system allows users to spatially experience the scene with 3D graphical augmentations. The ideas from 3D user interface design are also critical in AR applications.

More recently, advances in new forms of capacitive sensing, acoustic, computer vision, depth sensing, and voice recognition lower the barrier of incorporating them in mobile computing devices. There have been several commercialization trials on head worn AR devices such as Google Glass [64] which uses touch input and voice input, Hololens [122] incorporates gaze and mid-air gesture as well as voice input, and so on. Numerous moderate mobile devices including smart phones and tablets serve as powerful platforms for mobile AR applications where touch inputs have been widely used. Also, marker based tracking using computer vision has enabled tangible inputs for AR.

However, there is still a gap from being able to seamlessly experience AR due to many aspects. For example, the social acceptance of obtrusive touch input with Google Glass is debatable, the mid-air interactions could cause potential fatigue and accuracy issues, the interactions on touch screen lack context awareness about the surrounding AR scene, and the AR interactions in single-user cases may require extra efforts to adapt to multiple-user

collaborative cases. In this work, I explore new AR interactions and the enabling techniques to extend and complement the existing interactions metaphors.

1.2 Spatial Awareness in AR

A mobile AR supports users to access the 3D augmentations in a physical spatial context. The nature of registering the virtual contents within the physical context while users moves around means the spatial awareness of the physical world in a *geometric* level, i.e., the right virtual contents being rendered at the right place spatially with changes of the viewers' positions. For this matter, the emerging vision based tracking technique, such as Simultaneous Localizing and Mapping (SLAM) [102] based tracking affords the mobile AR device spatial awareness of the physical world. Further, spatial awareness of the physical world in a *semantic* level generates the context of the user interface. Using the specific context while users moves around in an environment, more intelligence can be embedded into the AR user interfaces.

The local spatial relationship between the input device and the AR device enable new interaction methods to manipulate the 3D graphical augmentations. For example, a 3D tracked tangible device has been widely used for direct manipulation of 3D virtual objects in the scene [24]. On the other hand, the global spatial relationship between the mobile AR device or the user and the surrounding environment provides references for interacting with the physical world in the AR scene [30]. Moreover, the depth sensing capability provided by the AR devices enables 3D perception of the real world, and extends the AR interactions in a 3D context. Further, to coordinate multiple users in a collaborative AR scene, spatial awareness across collaborators is critical for sharing and communication.

In our works, I study the spatial awareness level of AR interactions across different dimensions: namely the scale of the interaction space and the intelligence of the spatial references. For example, I study local 3D input methods with a tangible tool, extend 2D touch inputs from a touchscreen using depth sensing, leverage the relative position and orientation between mobile AR device and the environment globally, and synchronizing

spatial frames for multiple co-located AR users. Further, the spatial references could be used simply for visual augmentation such as direct manipulation of virtual contents, be computationally interpreted for virtual contents creation and alignment, be connected to the surrounding smart environment, and be collaboratively exploited by multiple users.

1.3 Towards Pervasive Augmented Reality

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it. Driven by this vision from Mark Weiser [189], pervasive computing or ubiquitous computing is emerging in various forms by embedding intelligence to the surrounding environment and empowering the personal computing devices with perception of the smart environment. By incorporating the pervasive computing capability, the existing AR experience will be further enhanced with understanding of the contexts. Grubert et al., recently defined pervasive AR to emphasize on creating continuous and pervasive AR interfaces which adapts according to the usage contexts [65]. The primary feature for a pervasive AR is being able to sense the contexts based on the 3D spatial relationship between the user and the augmented scene. Further, the adaptive design of the interface and interaction requires considering smooth transitions between different contexts while users switch tasks.

As the mobile computing devices increase in computation power and decrease in size, self-contained mobile AR devices such as handheld personal computing device and head worn smart glasses are emerging. Further, the rapidly growing high-speed cellular network supports accessing the information from the surrounding environment. In this work, I primarily focus on explore interactions driven by an Ad-Hoc (e.g., handheld AR) or always-on (e.g., wearables) approach towards pervasive AR. Moreover, most of the existing AR applications focus on augmenting a specified local scene with pre-defined virtual contents, while pervasive AR allows more flexibility of the AR contents creation and authoring. I also consider such flexibility as our essential design goal for the interactions. When it

comes to collaborative AR, I further emphasize enabling spontaneous and spatial interactions instantly without prior efforts such as scanning the room or setup of fiducial markers.

1.4 Overview

The overarching goal of this thesis is to explore novel spatial aware AR interactions and investigate the enabling techniques towards supporting a pervasive seamless AR experience. The spatial awareness in modern AR systems are particularly important as (i) it supports proper rendering and arrangements of the virtual contents in a physical and spatial context; (ii) it leverages 3D user interface design principles in AR; (iii) it provides strong interpretations for understanding the users' context in different AR and smart environment; (iv) it enables multiple users coordination in a collaborative AR. A summary of the contributions is as follows.

- Explore local spatial relationship between the input device and the augmented scene and enable 3D input.
- Leverage spatial awareness of the physical world in both a geometric level and a semantic level for interactions.
- Coordinate multiple users in a co-located AR collaboration.
- Investigate enabling techniques including magnetic based 3D tracking, projection based sketching, and radio frequency (RF) + SLAM based 3D localization and registration.
- Develop systems using the proposed spatial aware interactions towards creating a seamless pervasive AR experience.
- Evaluate the usability and effectiveness of proposed AR systems and interactions.

Overall, this work included 4 phases of research along the following approach as shown in Figure 1.1: (i) we motivate our research with the spatial awareness in the AR environment; (ii) we leverage the spatial relationships to investigate spatially-aware interaction

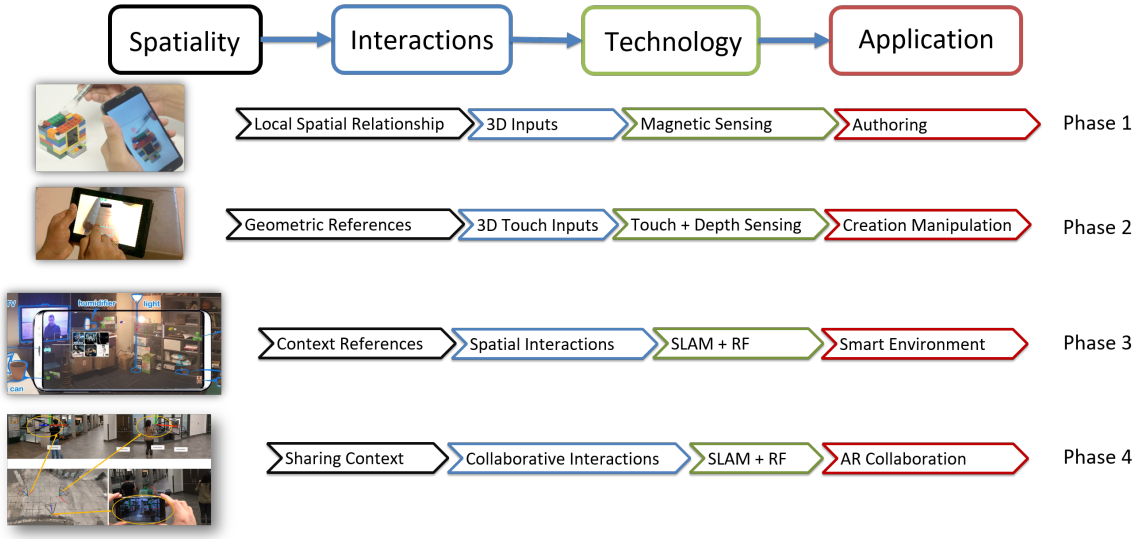

Figure 1.1. Overview of the research approach and four phases of the research.

metaphors; (iii) we then seek for technical solutions to enable the metaphors; (iv) and we finally develop systems and applications using the interaction metaphors and the enabling technologies towards providing more pervasive AR experiences.

Chapter 2 describes the related prior works, and compares our work and the state-of-the-art works. From Chapter 3 to Chapter 6, we discuss four phases of our research. In Chapter 3, I discuss enabling 3D input for a mobile hand held AR device using magnetic sensing techniques. In this work, we explored 3D spatial interactions with tangible proxies around the mobile device. Within local focused AR workspaces, we achieve physical interactions and authoring virtual contents in the scene. Chapter 4 describes our work in extending the 2D touch inputs from mobile devices to a 3D context by using the 3D perception capability provided by the depth sensing. We develop an AR virtual content creation system where users' sketches on the touch screen will be projected on the surfaces of objects in the physical world. This work focuses on studying leveraging captured geometric information of the real world for spatial interactions. In Chapter 5, we extend the spatial awareness of physical world in a geometric level into a semantic level, e.g., the AR interfaces are ware of the ubiquitous computing devices in the surrounding environment. We develop a RF

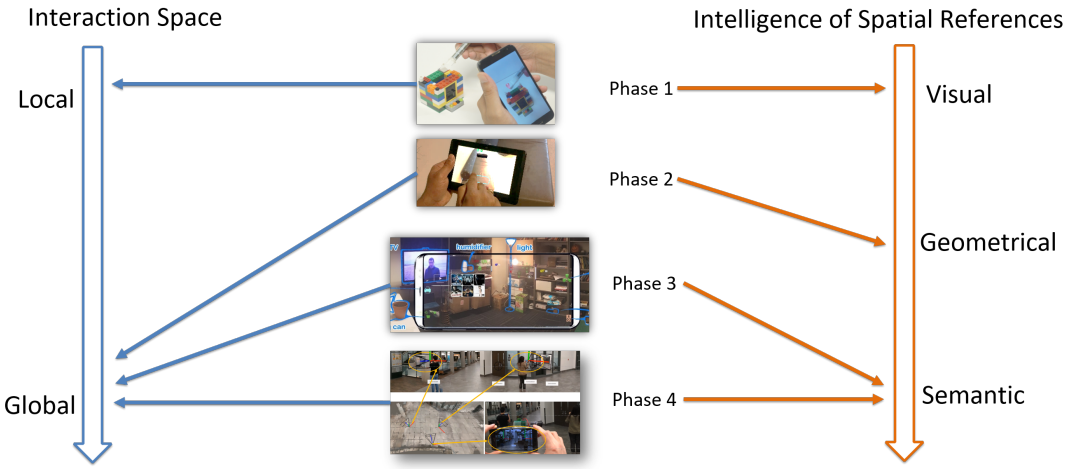

Figure 1.2. Distribution of four research phases on the spectrum of the spatial awareness in AR

+ SLAM based 3D localization method for the smart objects in the environment. In this work, we propose spatially aware interactions with the smart environment using the spatial relationship between the user and the smart object as a reference. Further, we explain coordinating multiple users in a co-located AR environment in Chapter 6. We develop a distance based indirect registration approach to resolve the transformations between separate SLAM devices without sharing maps or involving external tracking infrastructures. This approach allows for creating a spontaneous collaborative AR environment to spatially coordinate users' interactions.

Figure 1.2 illustrates the distribution of the four phases on the spectrum of the spatial awareness. For Chapter 3, we mainly focus on local interactions for AR. In Chapter 4, 5 and 6, we emphasize more on enabling mobile AR where users can freely walk around and interact within the environment. As for the intelligence level of the spatial references, Chapter 3 focused on applications which author basic visual augmentations onto the physical world. Further, Chapter 4 discusses methods to infer geometric information from the world for content creations. And finally, we leverage the spatial references at a semantic

level by connecting them to the smart environment (Chapter 5) and extend them to multiple user cases (Chapter 6).

Chapter 7 summarizes the major takeaways from the works in different phases and discussed the applications and expansions of the results in multiple on-going collaborative projects. For example, instead of interacting with IoT devices through AR, we explore equip a robot with the spatial intelligence from the IoT landmarks and perform autonomous exploration and mapping of a smart environment. On the other hand, we envision that in a human-robot-IoT ecology, spatial AR interactions contribute to an efficient in-situ authoring interface. Finally, in Chapter 8, I take a retrospective on the spatial awareness in AR and suggested three unique features offered by mobile AR: situatedness, mobility, and the capability supporting spatial collaboration. Exploiting these features, I suggest some future directions which may lead to potential killer applications of AR in real life: (i) human-in-the-loop simulation through AR, (ii) sharing context across heterogeneous agents, and (3) building transparent knowledge transferring interface with AR.

2. RELATED WORKS

Our research presented in this thesis is related to a range of research areas in human-computer interaction (HCI) including AR, spatial user interactions, embedded sensing, and ubiquitous computing. We will discuss the related work selectively for positioning our work in a broader background, and highlighting our contributions with respect to existing works.

2.1 Spatial User Interactions for AR

2.1.1 3D Input Around AR Device

For handheld smart devices, around-device interaction using 3D space has been explored with different set of sensing techniques while achieving equivalent performance with touch input [95]. Optical and vision based sensing techniques including depth cameras, IR proximity, and RGB camera are exploited to augment the general interaction with mobile device [37, 31, 169]. Also, for wearable displays, hand gesture recognition has been incorporated as a spatial input method [122]. However, these techniques require on the line-of-sight view of hand/interaction medium which limit the interaction space within the range of camera or optical sensor. In our work presented in Chapter 3, we adopt a magnetic sensing technique to encompass a full 3D volume around the mobile device. We develop spatial and physical interactions enabled by our technique to expand the interaction space by providing 3D tracking.

Around-device interaction with magnetic sensing has been investigated. Abracadabra and Nenya demonstrated 1D and 2D tracking techniques based on a single magnetometer to showcase the potential of magnet sensing as an input metaphor [12, 73]. In a similar manner, later works introduce the use of magnetic sensing to achieve delicate and rich mobile interactions [22, 35, 89, 100]. However, these works still focus on retrieving discrete

gesture inputs or 2D position tracking for symbolic interactions. We focus on embedding 3D tracking through which user's embodied motions are projected into intended interaction directly.

Magnetic sensing has been explored extensively for position tracking. Polhemus and Sixense both provide highly accurate 3D position tracking system in a large space [101, 150]. However, these approaches use an active magnetic source which requires the user to stay within the range of set-up space, thus not applicable for mobile usage scenario. Passive magnetic source has been adopted to accomplish 3D mobile input. GaussSense provides a magnet tracking system with 192 Hall-effect sensors embedded board [112]. However, the sensor board should be installed at the back of the device and only supports near-surface tracking (within 20mm). uTrack implements 3D position tracking of a permanent magnet using two magnetometers. It supports an accurate 3D inputs for wearables application [36]. As discussed by the authors, however, it still requires a desktop computation due to the extensive search algorithm. The heavy computation limits scalability and practicability as a stand-alone input technique for the mobile device. Our work presented in Chapter 3 provides a real-time tracking with a larger interaction volume solely based on the existing components of the mobile device.

2.1.2 Touch Input in 3D Context

Emerging smartphone, tablet and laptop equipped with smart styli enabling new input metaphors [121, 155]. Different aspects of the stylus have been studied including palm rejection [6], grip-based input [167], cross-device interaction [81], and high-resolution pressure sensing [131]. These approaches focus on either improving the digital pen experience more toward pen & paper interaction or enhancing the 2D user interface. In our work presented in Chapter 3, we directly extend the interaction volume of a tangible stylus from 2D on screen input into a 3D continuous input [197]. Respectively, we demonstrate AR applications where users perform discrete/continuous interactions in above/behind device spaces.

Touch input on handheld device supports intuitive and rich interactions which allows for complex 3D virtual content manipulation and creation. Previous explorations on 3D sketching using hand-held devices tended to separate the input devices from the display [153, 54]. Following a similar approach, Vinayak et al. [179] and Piya et al. [140] demonstrated systems that utilized smartphones as multi-touch controllers. Further, past works approached an immersive design environment by incorporating projection-based [98] or see-through head mounted displays [157]. Other works such as *(T)ether* [106] leveraged both touch inputs and mid-air gestures for modeling using a tracking infrastructure in conjunction with instrumented wearables and a tablet. *Paper3D* drew inspiration from papercraft and used multi-touch gestures for casual 3D modeling. In these works, the 3D shapes were created in either an empty physical space or in a virtual environment thus neither the dimensionality nor the visual appearance might pertain to their designated environment. While *Napkin Sketch* and *Second Surface* [96] started to merge the mobile AR technique, the physical reference in their works was limited to smaller working volumes and their outcomes were 3D wire or drawings. In Chapter 4, we concentrate on enabling the creation and editing of 3D models on arbitrary physical surfaces using a simple multi-touch interaction scheme [87].

2.2 Spatial and Geometric Information within AR

2.2.1 Creation and Authoring with Physical References

Virtual content creation using an AR-based system has also been explored. Previous works integrated instrumented tangible tools for operations such as modifying virtual models [9]. Nuernberger et al. [129] interpreted 2D drawing annotations using cues from 2D images and 3D geometry. *SnapToReality* [130] took a step further by extracting 3D edge and planar surface constraints from the environment and using them for precise alignment. Lau et al. [108] attached fiducial markers onto physical primitives and stamped the corresponding virtual shapes together to create shapes. Early work demonstrated virtually painting on physical objects [135] using a special designed brush. More recently, Mag-

nenat [117] et al. captured input drawings on normal papers and showed live texturing of 3D model in AR. Leveraging the merging mobile AR techniques, Xin et al. [194] presented *Napkin Sketch* as a system for creating 3D wire-sculptures on a napkin. Recently, *MixFab* [187] integrated scanned 3D model towards design using mid-air gestures in MR environments. While we draw inspirations from these approaches, in Chapter 4, our aim is to allow users to design directly on physical objects without being constrained by set-ups or being limited by the mobility.

Further, past works approached an immersive design environment by incorporating projection-based [98] or see-through head mounted displays [157]. Other works such as *(T)ether* [106] leveraged both touch inputs and mid-air gestures for modeling using a tracking infrastructure in conjunction with instrumented wearables and a tablet. *Paper3D* drew inspiration from papercraft and used multi-touch gestures for casual 3D modeling. In these works, the 3D shapes were created in either an empty physical space or in a virtual environment thus neither the dimensionality nor the visual appearance might pertain to their designated environment. While *Napkin Sketch* and *Second Surface* [96] started to merge the mobile AR technique, the physical reference in their works was limited to smaller working volumes and their outcomes were 3D wire or drawings. We concentrate on enabling the creation and editing of 3D models on arbitrary physical surfaces using a simple multi-touch interaction scheme.

2.2.2 Spatial Reference Based Interactions

With the relative spatial relationships between the user and the IoT device, we extracted three basic spatial elements: the *orientation* of users with respect to the IoT devices, the direct *distance* measurement between a user and an IoT device, the *approaching direction* in which users walks. Based on these three relationships, researchers have developed location aware interactions, such as distant pointing and proximity based control [109, 151]. Moreover, previous works demonstrated the visualization of the overlaid digital contents both inside and out side the view [66, 114]. These digital augmentations were rendered based

on the spatial locations of the corresponding physical objects in the environment. Further, incorporating multiple modalities for interacting with smart environment has been investigated. For examples, past works leveraged the spatial relationships to provide both visual and auditory augmentation [152], and context-awareness with voice command [136]. In Chapter 4, we utilize the depth sensing capability and projected the 2D touch input in the 3D context. Chapter 5 refers to our study on mapping the smart objects spatially in the AR scene and the enabled interactions [86].

2.3 AR in Pervasive Computing Environment

2.3.1 Context Awareness of Ubiquitous Computing Devices

Moderate pervasive computing devices such as mobiles and wearables, are able to discover the smart things connected to the same network and retrieve the corresponding interfaces effortlessly. Enabling context awareness of the surrounding smart environments on mobile devices has been the focus of ubiquitous computing community [60]. Previous works attempted to identify and select smart devices through a *touching* or a close proximity interaction [151], which means that the user needs to either physically contact or be present in close proximity (within 1m) to the target. Early works incorporated short-range RFID readers [185] or near field communication (NFC) chip [151] in mobile devices to link with a smart device. More recently, through leveraging the electromagnetic (EM) emissions from the smart devices, researchers have investigated using machine learning to recognize the EM signatures [107, 193, 184] without instrumenting the devices. To achieve selecting the device at a distance, various technical approaches have been considered including ultra-high frequency (UHF) RFID [176], infrared targeting [38], visible light sensing [159], visual fiducial tags [78, 68], and visual natural features [44]. To this extent, previous efforts primarily focused on local interaction with a device with prior knowledge of where the smart devices were located in the environment. The spatial knowledge about the smart devices played an important role in context awareness [30, 60, 109, 147]. In Chapter 5, we emphasize the discovery of absolute positions related to the environment using a wireless

localization method at a distance. We bring the location awareness to the smart devices and thus enable mobile spatial interactions, specifically within an AR scene.

2.3.2 Interacting with Smart Environment in AR

Recent works have showed great interest in leveraging mobile AR technology to interact with the smart environment [78, 113, 120, 159, 161]. However, the spatial relationship between the AR device and the smart device remains local, which means the augmentation only applies on one or multiple specific devices in the instant view. Yet, the awareness of the surrounding environment as a whole ecology in AR requires the mapping the devices using their 3D locations in the AR scenes. The SLAM based tracking technique, which brings the AR device awareness of the physical environment, has significantly matured in past years and has started to appear on commercialized product [13, 63, 122, 190]. However, the SLAM map itself has no semantic information. Recent researches has show progress in object detection [139] and pose estimation [154] working with visual SLAM. But it is still challenging to discover the 3D locations of all smart devices scattered in a cluttered scene by only using computer vision. Therefore, we propose the use of a wireless localization technique together with SLAM, which require no prior knowledge of the smart environment, to estimate the absolute positions of the smart devices instantly.

The wireless localization and mapping problem, especially in indoor environment, has been studied extensively [4]. We primarily consider the infrastructure-free localization techniques since we aim at instantly estimating the locations of the devices. We draw inspiration from the concept of wireless sensor network (WSN) localization which estimates nodes' positions as the smart devices naturally form a network. A common solution which deals with indoor environment is a distance-based localization method which derives the coordinates by measuring the distances across the nodes [5]. For examples, Multidimensional Scaling (MDS) and its variations are widely used distance based methods [26, 43]. Recent developed UWB technology which provides an accurate distance measurement($\sim$ 0.1m) leads to a highly accurate localization [49]. However, these ap-

proaches usually consider only stationary nodes. Hahnel et.al. proposed an idea using a mobile robot which was equipped with UHF RFID reader and two antennas to survey an environment and localize the RFID tags placed in the environment [69]. In the work discussed in Chapter 5, we merge this surveying idea into the WSN localization solutions. We leverage the mobility of our mobile AR device to survey the smart environment [86].

2.4 Collaborative AR Systems

2.4.1 Co-located AR Collaboration

The paradigm of AR has been introduced for both remote and co-located collaborations. Gauglitz et. al. leveraged SLAM technique to reconstruct a surface model of the local scene supporting virtual navigation in a video conference [59]. Oda et al. proposed to use virtual replicas to assist remote collaboration in AR [132]. Further, researchers investigated telepresence systems to enable life-size dynamic interactions between remote users. Room2Room [137] showed a projected augmented reality system, and Holoportation [134] utilized a head-mounted display (HMD). In a remote collaboration scenario, the interactions either stay loosely connected with the physical scene [134, 137] or constrained within a controlled small volume [132] since the local environment differs from the remote one. On the other hand, involving multiple users in a collaborative co-located environment requires synchronizing spatial frames across different users [20, 160]. This aspect is different from a single-user or remote collaborative AR application. Early explorations on co-located scenario such as Shared Space [25] and Studierstube [158] augmented face-to-face collaborative experience with AR. Vita [21] presented a 3D model visualization and manipulation system supporting multiple users. The interaction volume of the pioneer works were restricted by the external tracking setups, e.g., fiducial marker, electromagnetic, inertial, and multi-camera systems. Further, the cumbersome infrastructure counteracts the imperative mobility and immediacy of AR collaboration activities. In Chapter 6, we focus on constructing a shared augmented physical space instantly by synchronizing multiple users' local SLAM coordinates [88].

2.4.2 Synchronization of Spatial Frames

In a collaborative environment, common spatial references are crucial for communication and coordination [57, 160]. Registering multiple users together within a global frame using external vision based tracking systems have been used in previous works [67, 105, 109, 191]. Other works set up the global frames with different sensor based alternatives including GPS for outdoor environment [146], electromagnetic [141, 165], inertial [21], ultrasonic[60], RF based tracking for indoor scenarios [39]. Besides, registering users to a common anchor scene spatially also derives transformation between users for coordinations. Researchers have used fiducial markers [25] or pre-captured scene images [96] as anchors. Further, with the emerging SLAM techniques, a SLAM map of the shared scene which is offloaded to multiple users allows for flexible and mobile coordinations [10, 34, 124]. Moreover, collaborative SLAM supports multiple agents to share and build the map in real-time [55, 83, 123, 148]. In our work (Chapter 6), instead of sharing the SLAM maps, we emphasize on promoting spontaneous AR collaborations.

2.4.3 Peer-to-Peer Tracking and Localization

The advantages of utilizing the embedded camera on the SLAM based AR device to directly track the pose of the collaborator are obvious. Despite the convenience of avoiding introducing extra components, it has been challenging to accurately estimate the full pose of a wearable or hand-held AR device accurately from images where it is being operated by a user [170]. Other direct tracking alternatives such as electromagnetic sensing [84, 141, 165] are not applicable to mobile AR devices because of the high power consumption and bulky size of the base.

In contrast, the indirect approaches measure distances, angles-of-arrival or RSSI with RF based technologies and then derive the relative transformation between RF units. The indirect approaches have been widely used for wireless sensor network (WSN) localization [118]. Hazas et. al. applied ultrasonic based ranging for distance and angles-of-arrival

measurement to derive the 2D localizations of statically placed devices [75]. Gellerson et.al. explored spatial aware mobile user interfaces with similar method [60].

Our work in Chapter 5 demonstrated an approach combining SLAM based mobile AR with UWB units to localize multiple Internet-of-things (IoT) devices distributed in 3D space [86]. Comparing with ultrasonic based sensing, UWB provides much larger sensing ranges with high accuracy [46]. These works primarily focused on either multi-user collaboration in a static setup or a single-user interacting with static surrounding devices. Further, *SynchronizAR* from Chapter 6 contributes towards supporting spontaneous collaboration in general but highlights enabling spatial collaboration activities among freely moving users in AR. Besides, comparing with [86], *SynchronizAR* derives not only translational but also rotational transformation between users.

3. ENABLING 3D INPUT AROUND MOBILE HANDHELD AR DEVICES

Recent developments in smartphone displays and sensors have resulted in enhanced visual experiences such as mobile augmented (AR) and virtual reality (VR) [61, 180]. To support these 3D interfaces, previous study suggested on providing a natural correspondence like human motion in 3D space from the input device [71]. 3D input method also offers more intuitive and quicker way to interact with 3D interfaces [162]. To this extent, researchers have proposed an around-device mobile interaction [31]. It frees a physical boundary limited by mobile device screens and incorporates surrounding 3D space as an interaction space. Recent works employ 2D tracking [73] and event-based discrete inputs [89] in 3D space to enlarge the interaction space. Inspired from these works, we develop a real-time 3D position tracking technique, which enables rich spatial mobile input.

Acquiring input data from 3D mobile space has been investigated through vision and magnet-based techniques. Recent work shows mid-air gesture-based interaction using a depth camera [37]. Occlusion and lighting condition still limit the use of vision-based techniques in mobile environments. On the other hand, the magnetic sensing techniques which are free from occlusion and different light conditions have also been investigated [36, 112]. Although these works show high 2D/3D tracking accuracy in real-time operation, they still require either a desktop computation, or extensive modifications on the mobile device.

In our work, TMotion enables the mobile device to track a stylus embedded with a magnet and an IMU. Specifically, the algorithm calculates the magnet's position relative to the mobile based on the magnetic field vector and the orientation of the embedded magnet. We achieve a 3D position tracking rate greater than 30Hz possibly with mobile device. As a 3D mobile input, TMotion supports continuous/discontinuous interactions in above/behind device spaces. Our contributions include the following: (i) a novel sensing technique providing a real-time position tracking as 3D mobile input; (ii) an analysis of experiments

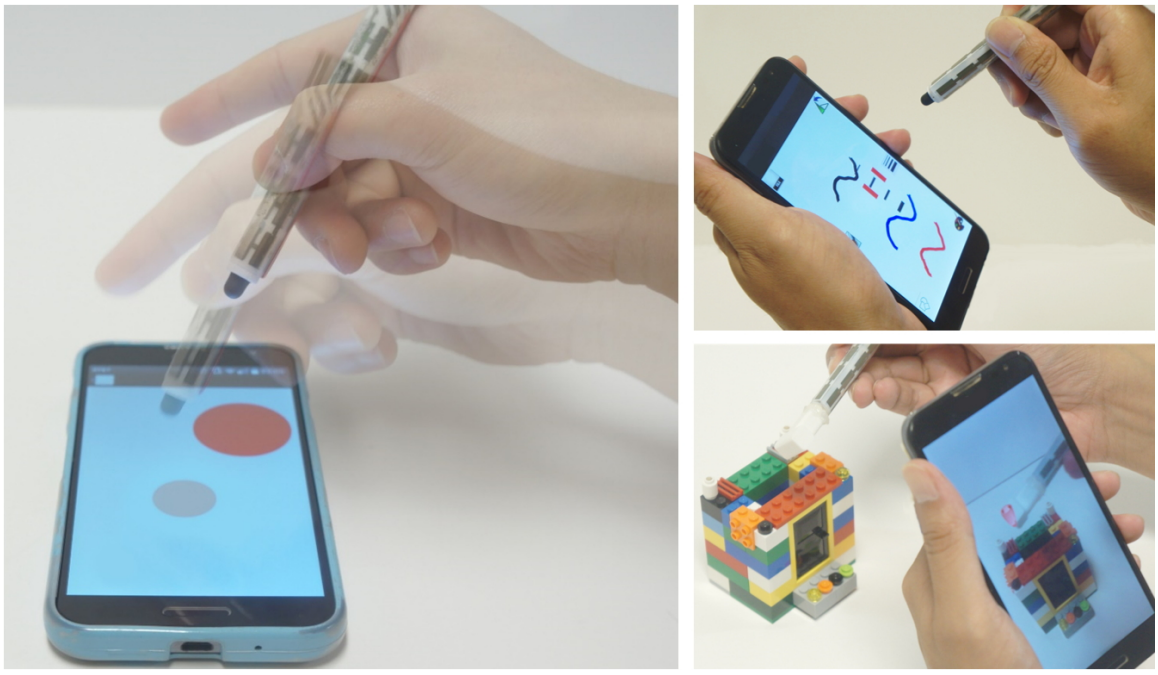

Figure 3.1. TMotion enables a real-time 3D position tracking using embedded permanent magnet and IMU with existing mobile device. TMotion provides interaction spaces above and behind the device while supporting discrete and continuous interactions.

and task evaluations including tracking and targeting accuracy using TMotion; (iii) demonstration of example applications exploring embedded continuous/discrete interactions in expanded spaces.

3.1 System

2D and 3D position tracking using multiple magnetic sensors have been explored [36, 70, 112]. However, they require either hardware modification or desktop computation. In this section, we introduce the background knowledge of the magnetic field sensing and our novel approach.

From the magnetism theory, 3D position of the permanent magnet in the magnetic sensor oriented space ($\mathcal{F}_{mobile}$) can be solved using the following equation

$$\mathbf{H}(\mathbf{r}) = \frac{K}{r^3}\left[\frac{3\mathbf{r}(\mathbf{m}\cdot\mathbf{r})}{r^2} - \mathbf{m}\right], r = |\mathbf{r}|, K = \frac{M}{4\pi} \tag{3.1}$$

Here, H refers to the magnetic field vectors, M denotes for the magnetic moment, **m** is the directional vector of the magnet, and **r** is the location vector of magnet relative to the sensor. With known **m**, M, and H, **r** can be solved.

We assume magnet is located at (x, y, z) resulting in **r** to be $(-x, -y, -z)$. The directional unit vector of magnet is ($\mathbf{M}_x$,$\mathbf{M}_y$,$\mathbf{M}_z$). We perform space transformation from IMU

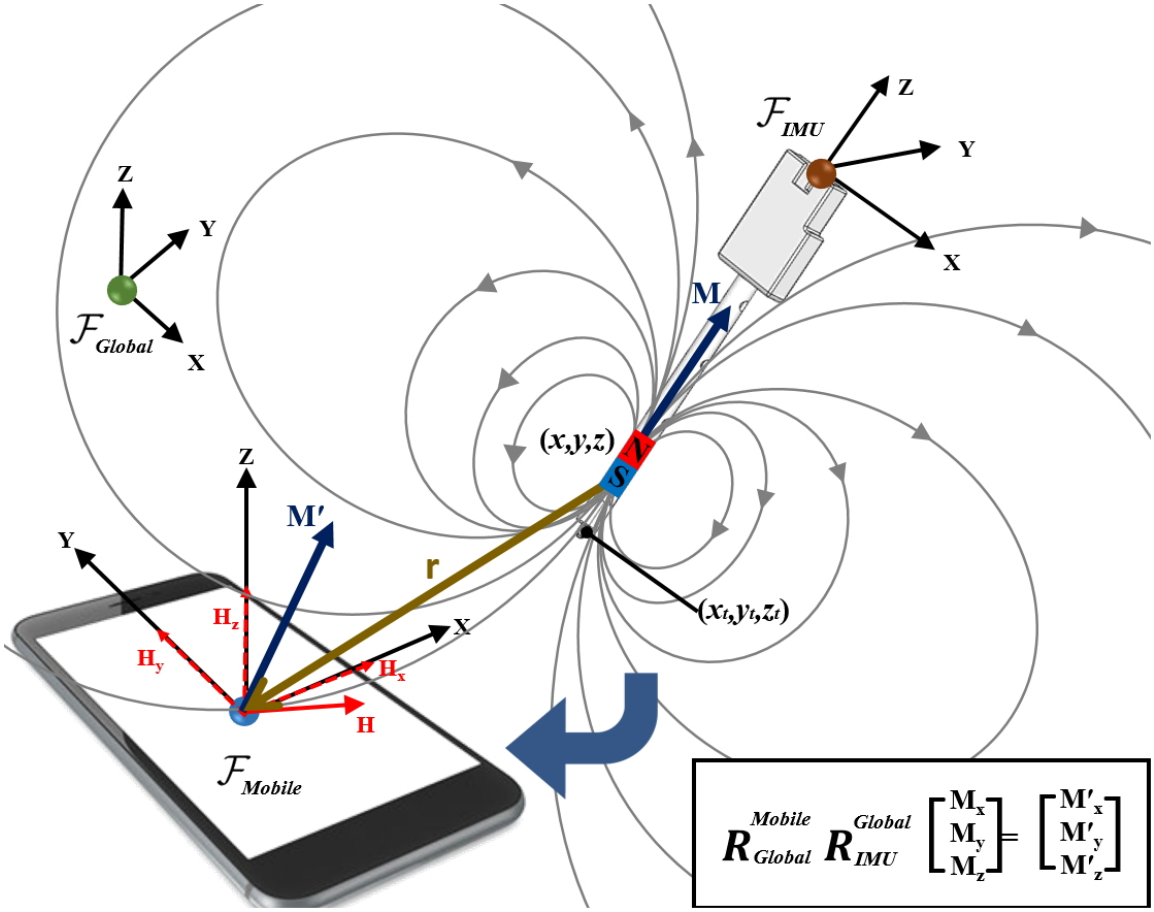

Figure 3.2. Magnetic vector (**H**) is generated by magnet. Magnetic directional vector from TMotion (M) is transformed to mobile's frame (M').

space ($\mathscr{F}_{IMU}$) to mobile space ($\mathscr{F}_{mobile}$). Figure 3.2 illustrates the transformation of the directional unit vectors (**M**) from TMotion to the mobile space (**M**′). Thus, Eq. (3.1) can be dissected into the following three scalar nonlinear equations.

$$\mathbf{H}_x = \frac{K}{(x^2+y^2+z^2)^{\frac{5}{2}}}\big[-3x(-M'_x x - M'_y y - M'_z z) - M'_x(x^2+y^2+z^2)\big] \tag{3.2}$$

$$\mathbf{H}_y = \frac{K}{(x^2+y^2+z^2)^{\frac{5}{2}}}\big[-3y(-M'_x x - M'_y y - M'_z z) - M'_y(x^2+y^2+z^2)\big] \tag{3.3}$$

$$\mathbf{H}_z = \frac{K}{(x^2+y^2+z^2)^{\frac{5}{2}}}\big[-3z(-M'_x x - M'_y y - M'_z z) - M'_z(x^2+y^2+z^2)\big] \tag{3.4}$$

$$[J(x^{(n)})v^{(n)}] = -F(x^{(n)}), \quad e = 10^{-7} \tag{3.5}$$

$$\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + R_{Global}^{Mobile} R_{IMU}^{Global} T_{Tip}^{Magnet} \tag{3.6}$$

By taking known orientations from attached IMU (**M**) and 3-axis magnetometer readings (**H**) from a mobile device as inputs, we employ Newton's method (Eq. (3.5)) to solve nonlinear Eq. (3.4). Figure 3.3 illustrates the system flow of our technique:

1. Input orientations from IMU (**M**) and magnetometer readings from phone's magnetometer (**H**) to the system.

2. Apply space transformation to calculate orientation (**M**′) in the mobile space ($\mathscr{F}_{mobile}$)

3. Apply Newton's Method to solve nonlinear equations (Eq. (3.5)). If it fails to converge ($e<10^{-7}$) within 15 iterations (i) or diverges ($e>10^{3}$) at any time, returns to the beginning to process new input signals.

4. On successful computation, updates an initial value with a new root (x,y,z) and apply transformation (Eq. (3.6)) to the root (x,y,z) for deriving the tip position (x_t,y_t,z_t).

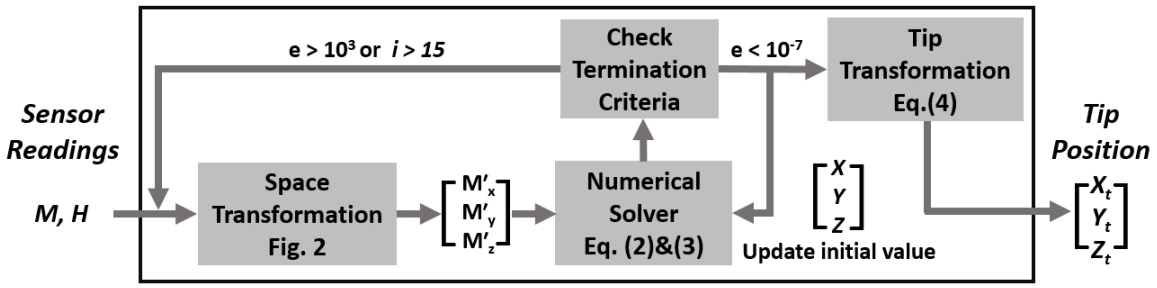

Figure 3.3. Tracking algorithm finds magnet's position through numerical solver and performs transformation to output the tip position. With known orientations, exhaustive search is not required.

Our approach enables a faster computation since we conduct the numerical solving once with known orientations from IMU. Whereas [36] requires multiple iterations of solving equations for the exhaustive searching. In preliminary work, we observe that the position tracking succeeds when the prototype operates within 160mm $\times$ 160mm $\times$ 200mm volume around the mobile device. The limited sensing range is due to the fact that the magnet strength is inversely proportional to the cubic distance to the magnetometer. Newton's method fails to converge occasionally due to mismatched pair of inputs (IMU orientation & mobile's magnetometer reading). The mismatches are potentially caused by the low signal to noise ratio when the permanent magnet locates at the tracking range borderline. To compensate this issue, we simply apply thresholding to pass valid sensor readings to the numerical solver. With the mitigation, we do not observe computation failure during continuous motion within the interaction volume.

In our work, we adopt a 9 degrees-of-freedom (DOF) IMU to disambiguate the unknown orientations which enables real-time mobile 3D tracking using a single magnetometer. Thus, we achieve a stand-alone mobile input which performs in a real-time and can be used with an unmodified mobile device. This approach distinguishes us from related works [36, 112].

3.2 Implementation

Figure 3.4 illustrates our prototype in detail. The diameter and the length of the prototype are 10mm and 170mm respectively. The prototype can hold multiple form factors to support embedded magnets of various orientation and size. While the stylus form is assumed to offer better comfort on and above device interaction, the wand design is considered to provide better comfort for behind device interaction. The conductive rubber is placed at the stylus tip to support conventional touch input. In our demonstration, we use a cylinder-shaped, N42 grade, neodymium magnet with 3.2x11mm in diameter and length respectively.

3.2.1 Hardware

For orientation, we use Sparkfun's 9DOF sensor stick which comprises of gyroscope (*Invensense ITG-3200*), magnetometer (*Honeywell HMC5883L*), and accelerometer (*Analog*

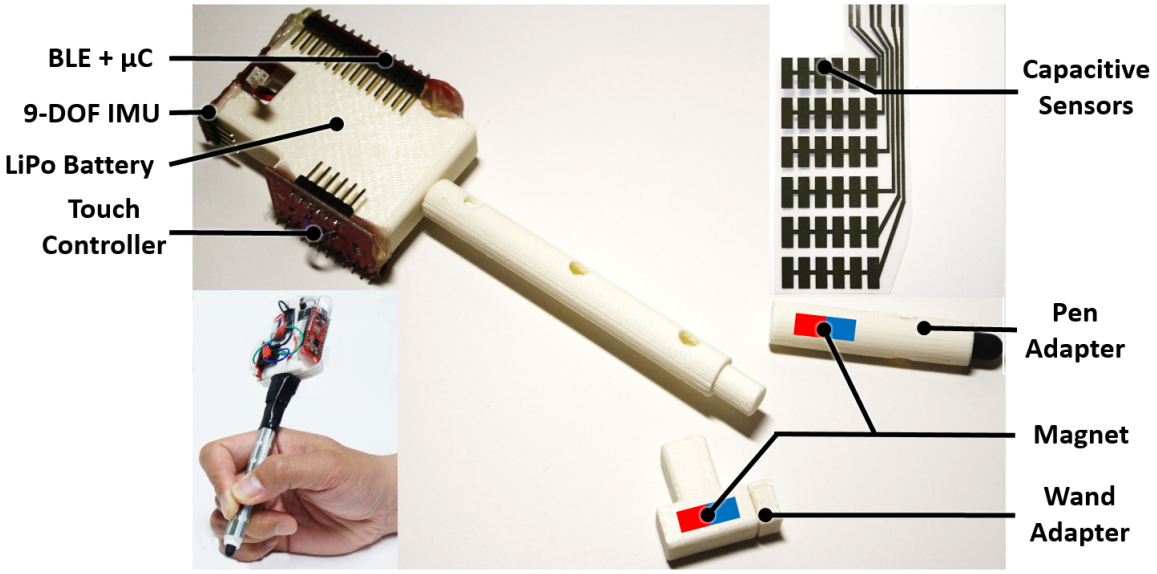

Figure 3.4. TMotion prototype and breakdown of its components. Permanent magnet and 9DOF-IMU are embedded for 3D position tracking.

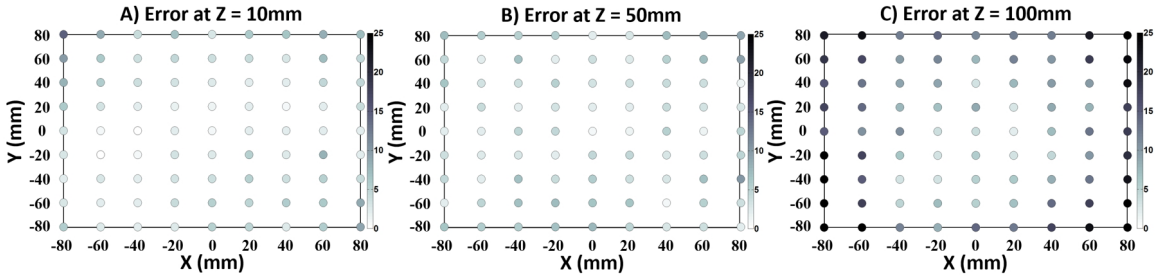

Figure 3.5. Tracking accuracy is tested in three heights a) 10mm, b) 50mm, and c) 100mm using TMotion. The grayscale indicates the Euclidean distance between the ground truth and our tracking. Origin of the graph represents the center of the magnetometer from mobile device's side. The mean error is 4.55mm in the volume that covers the 5.5″ smartphone, 80mm (x-axis) × 120mm (y-axis) × 100mm (z-axis)

Devices ADXL345). These sensors meet the technical requirement including sensing range and resolution. To avoid the magnetometer saturation, we configure the sensor stick and the embedded magnet in distinct locations (>5cm) in our prototype. Furthermore, we adopt an one-time calibration including scaling each axis value relative to the gravity (accelerometer), subtracting offset reading (gyroscope) and soft+hard iron calibration (magnetometer) [28]. The initial calibration process ensures the functionality of the IMU regardless of the embedded permanent magnet. The microcontroller integrated with a Bluetooth 4.0 Low Energy (BLE) module (ATmega32U4, Nordic nRF8001) captures and transmits analog readings from sensors to the smartphone wirelessly. We use a 110mAh battery which provides 6 hours of active operation with peak performance. For capacitive sensing, we inkjet-printed a sheet of electrodes using *AgIC* ink while processing capacitive proximity through *MPR121*.

We formed a self-contained setup using *LG Optimus G Pro* smartphone (1.7GHz quad-core with 2GB RAM). We were unable to retrieve the location of the embedded magnetometer from vendor's manual, and that necessitates an additional magnetometer attachment on the mobile. Here, we added a single *HMC5883L* with a microcontroller at the

back of the phone using On-The-Go cable. Evidently, given the accurate sensor placement information of the mobile device, we need no such modification.

3.2.2 Software

The orientation of the prototype is computed using Direction Cosine Matrix algorithm for fast and stable performance during dynamic motion. The microcontroller streams calculated Euler angles and capacitive touch sensor values (15 bytes in total) through the BLE module (45$\sim$50Hz). With the streamed mobile's magnetometer data (75Hz), we update the tip position from the latest computation. In our test setup, each numerical computation takes between 1$\sim$8ms (3ms in average), which results in overall tracking rate of $>$30Hz. In the example applications, we adopted an exponential filter to smooth the raw data. For capacitive sensing, we set threshold value to detect the tap gesture. The system requires an initial calibration to compensate noises from the geomagnetic field. We subtract average magnetometer readings before the prototype gets into the interaction volume.

3.3 Evaluation

3.3.1 System Evaluation

To find out the tracking performance of TMotion, we have conducted three experiments: tracking accuracy in different 1) heights, 2) orientations, and 3) tracings. We measured accuracy performance by comparing Euclidean distance between a physical ground truth and computed positions. We set a plastic shelf (160mm$\times$160mm) covered with a grid paper (20mm space in both x and y directions). We adjusted the height of the shelf with a set of blocks to test the prototype in heights of 10, 50, and 100mm above the mobile device. We placed the prototype's tip on each grid intersection point with normal usage orientations (0$\sim$60°) and recorded 100 readings at each point. The overall testing volume was 160mm (x-axis), 160mm (y-axis), and 10$\sim$100mm (z-axis) about the magnetometer's center, with a total number of 24300 data points (100 readings x 81 intersections x 3 heights).

To further investigate the effect of the different orientations, we rotated the prototype around a set of fixed points for (1) normal usage range ($<60^\circ$) and (2) steeper tilt angles ($60\sim90^\circ$). A total of 5000 data points were captured at five fixed points [0,0],[-50,-50],[-50,50],[50,-50],[50,50] with z=50mm. At last, we traced the printed shapes on the testing jig which were assumed to be our ground truth. We repeatedly traced each shape for 40s and captured more than 1000 data points.

Results

Figure 3.5 illustrates the Euclidean distance between our readings and the ground truth at each point. In a total volume with 160mm(W)$\times$160mm(H)$\times$100mm(D), an average error is 6.27mm (σ = 4.56mm). The errors are mainly caused by the environmental magnetic field noises as the prototype moves away from the sensor similar to previous works [36, 73]. If we narrow down to an interaction space of 80mm(W)$\times$120mm(H)$\times$100mm(D) which still encapsulates the 5.5$''$ smartphone, the error significantly reduces to the 4.55mm (σ = 2.6mm). It is also noticeable that the tracking shows more errors near the center at z = 10mm than at z = 50mm. Such inconsistency is caused by the saturated magnetometer readings when the magnet approaches the center at z = 10mm due to the strong magnet strength. For later task evaluations and applications, we adopt a height range of 10$\sim$100mm as our interaction space.

We carried out experiments to test performance variations during different orientations and dynamic tracings. For normal usage orientation ($<60^\circ$), the mean error was μ = 5.66mm (σ = 3.33mm). The steeper orientation ($60^\circ\sim90^\circ$) showed no significant increase in the mean error (μ = 6.13mm, σ = 3.05mm). Thus, our tracking technique performs uniformly regardless of tilt angles. For tracing, we came up with a visual inspection of traced data points to confirm the dynamic performance of our tracking. As shown in Figure 3.6, the tracking performance does not degrade significantly comparing with previous results. The tracing results still form a shape similar to the ground truth and the z-direction tracking deviates within $\pm$1.5mm.

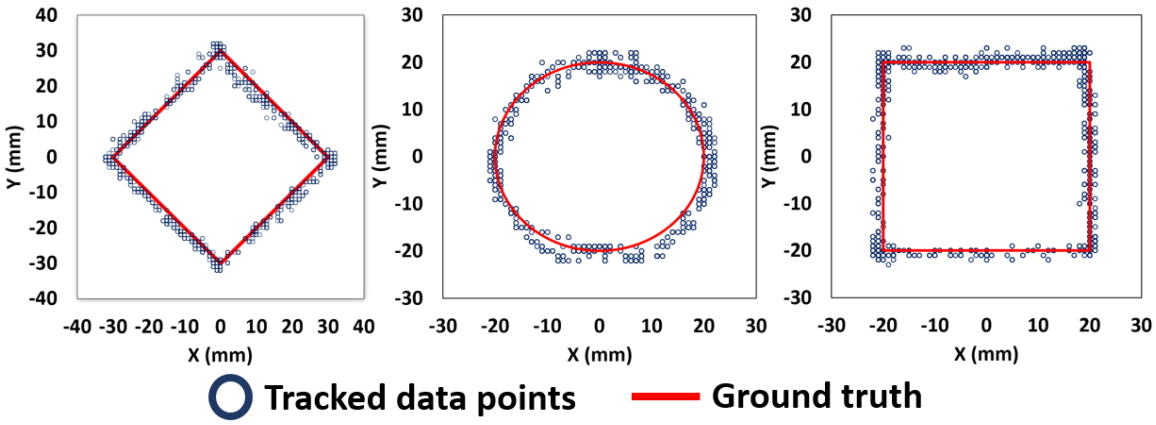

Figure 3.6. Visualization of the shape tracing at 50mm above the device.

3.3.2 Enabling Spatial Interactions

To demonstrate the usage scenarios of our technique in around device interactions, we develop four applications. Enabled by the occlusion free 3D position tracking with TMotion, we are capable of expanding the interaction space to both above and behind device. On the other hand, TMotion delivers a wide range of interaction types such as hovering, tracing, and pointing. As illustrated in Figure 3.7, we categorize the provided interactions into continuous spatial tracking and discrete spatial zoning. We consider the spatial tracking as a continuous relationship tailored to the user intents expressed by natural motions. And we characterize the spatial zoning as a dissection of physical volume around the mobile device or the real object into several zones to embed discrete information.

In above device interaction space, we demonstrate the spatial tracking feature with an example that associates user movement with the measurement of object's dimensions. The multi-level menu interface shows how we use above device spatial zones to embed discrete information. For behind device, we leverage the back camera on the smartphone, and construct applications in AR environment. Through this set up, we show direct manipulation and registration of digital contents within the augmented scene using continuous and discrete interactions respectively.

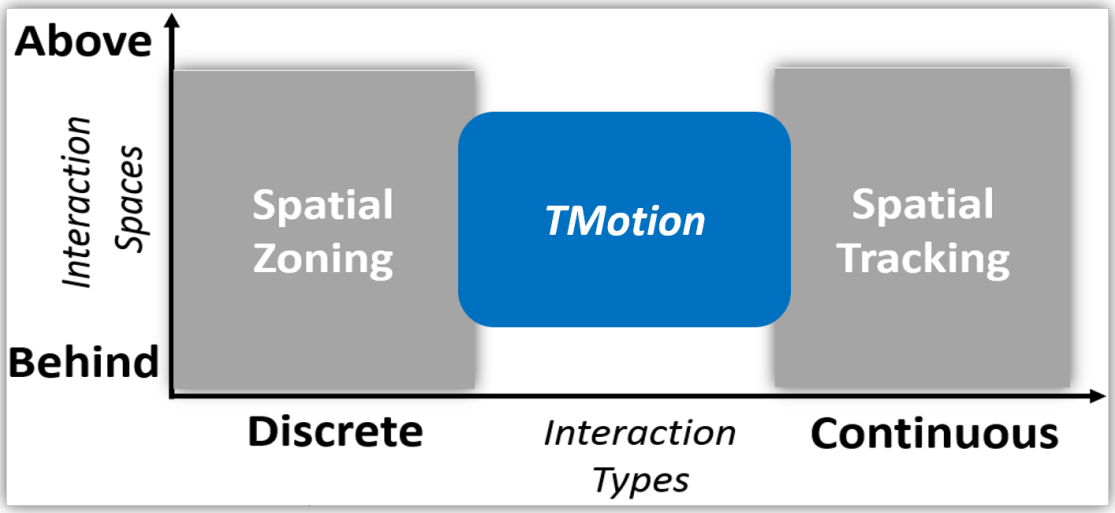

Figure 3.7. The spatial interactions enabled by TMotion

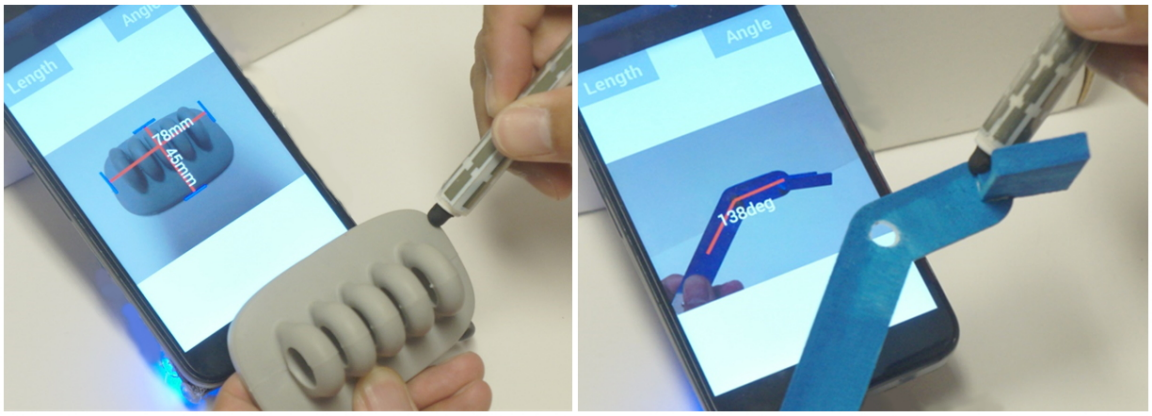

Figure 3.8. 3D position tracking guided by tap gesture enables physical measurements of length (Left) and angle (Right) above the device.

Above Device Interaction

Spatial tangible measurements: With a spatial tracking above the device, application designers are encouraged to utilize the mid-air interaction space. As described in SPATA [186], the measurement is one of a key element for fabrication-aware context, es-

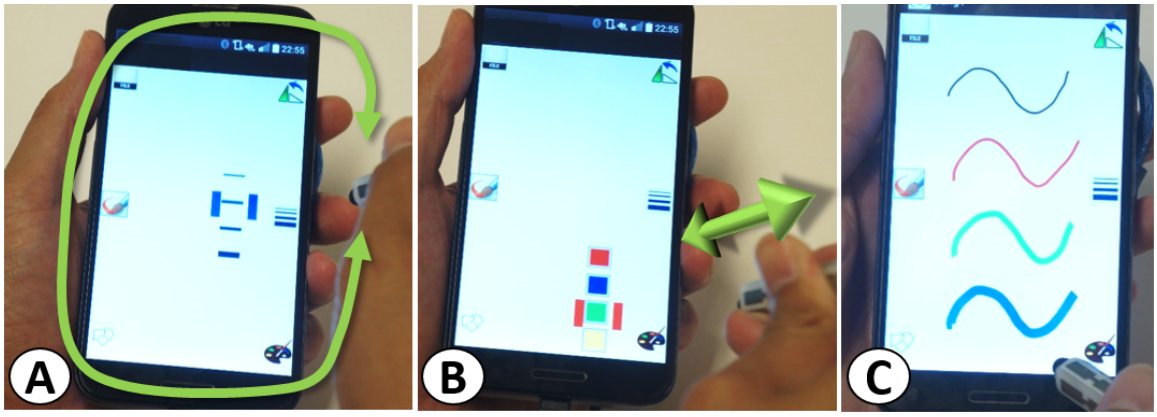

Figure 3.9. TMotion enables a mid-air multi-level menu control offering (A) hovering around the device to open option lists and (B) depth control and tapping for option selection (C) in a drawing application.

pecially for designers. Here, we develop an application which measures dimensions of real objects. First, users take picture of the target object. Then, users pre-annotate the measurements that will be taken. Subsequently, users place the stylus tip on the interesting points and tap pen body to complete the measurements. For length and angle measurements, 2-points and 3-points selection are required respectively. Upon completing the physical measurement, results will be displayed on the pre-selected annotation label (Figure 3.8). This illustrates TMotion's capability to achieve the user-guided spatial tracking above the device.

Multi-level menu interface: Previously, single menu control using around device interaction has been demonstrated based on 2D tracking [89, 112]. Using 3D position information offered by TMotion, we implement richer interactions through 3D spatial zones formed around the mobile device. We constructed a drawing application embedded with a mid-air controlled multi-level menu interface. While hovering around the displayed icons, user pops up a first-level menu. Then, the user moves along the z-axis to hover the option list and taps to confirm selection. This showcases richer interactions using discrete spatial zones around the mobile device.

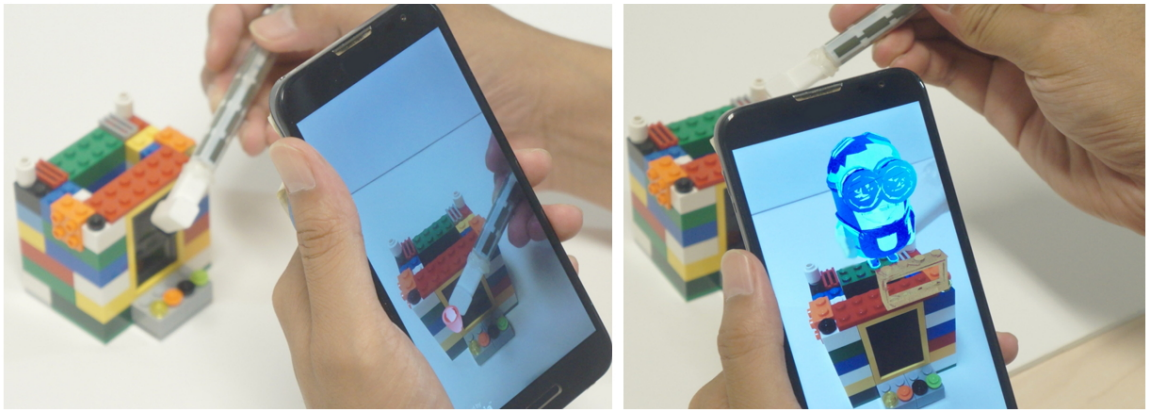

Figure 3.10. TMotion is aligned with virtual model in the augmented scene (Left). The system enables manipulating virtual blocks with respect to the physical object (Right).

Behind Device Interaction

In our AR applications, we use Vuforia^{TM} SDK for tracking in physical environment. In both demos, pre-built LEGO blocks are used as world frame reference. The natural feature points of the LEGO blocks are first captured and stored for object tracking and recognition purpose. Furthermore, we align the physical pen tip with corresponding virtual contents within an augmented scene.

In-situ building blocks: The early tangible AR manipluation which is based on monocular vision tracking suffers from occlusion and bulky size of the marker [200]. On the other hand, TMotion enables a low profile 3D input device in mobile AR application by providing full 3D tracking capability. Here, we apply TMotion to manipulate the virtual contents directly. Users place and drop virtual models onto the existing LEGO construction within the augmented scene. The virtual creations are superimposed onto the designated locations. Then, users conduct visual inspections from different points of view by moving the mobile device. This example showcases the use of continuous interactions behind the device.

Digital contents overlay: The mobile AR setup also suffers from the limited alternatives to interact with the physical environment. Vuforia SDK provides a virtual button

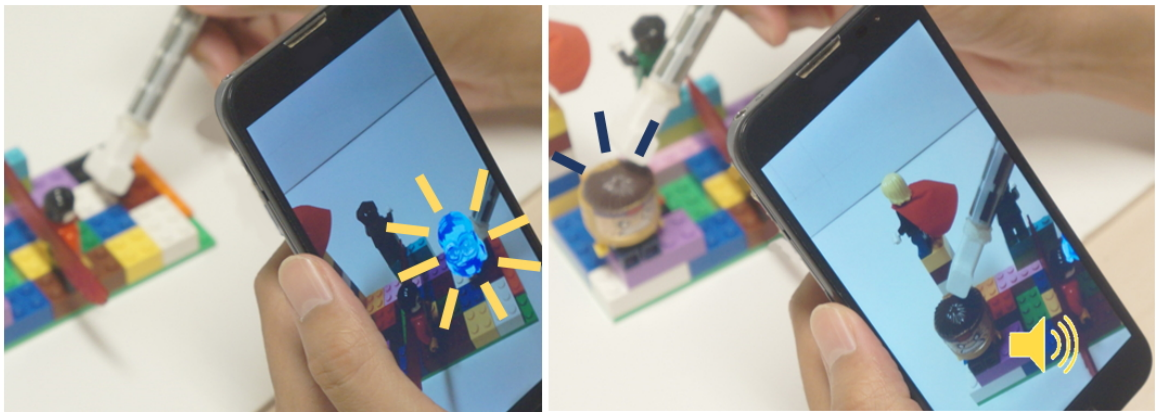

Figure 3.11. TMotion interacts with the spatially embedded digital contents around the real objects such as discovering hidden virtual character (Left) and playing sounds for different characters (Right).

solution which is triggered by blocking the line-of-sight view. Such solution requires users to block the printed buttons on a marker sheet to trigger them. However, 3D tracking using TMotion allows us utilizing the discrete spatial zoning feature. We successfully embed the virtual contents including sounds and virtual characters into the dissected space around the physical LEGO blocks. To access the contents, user can hover or tap in the specific regions in the physical world. This demonstrates TMotion's capability of providing discrete interactions behind the device.

3.4 Discussion

In this work, we show that TMotion achieves a real-time 3D position tracking with a deeper understanding of user intents in 3D mobile space. Our work represents human's natural motion with physical input device as an embedded 3D interaction. Demonstrated applications show a potential to offer new interaction metaphors which cannot be provided by previous 2D tracking or gesture based discrete inputs. Here, we discuss design implications, limitations and future work.

Coarse Interaction Strata for Discrete Input: We observe that the performance of targeting in the mid-air using a physical input device becomes worse under 20mm layer thickness. Multiple factors other than the system performance comes into play such as fatigue from the users during the mid-air interaction. This implies that even if the system supports better accuracy, the users still have limited discrete controllability above the device. Aligned with previous study [181], our finding also suggests to use coarse interaction strata for above device interaction with 3D mobile input to provide an acceptable discrete input controllability.

3D Mobile Input as Spatial Tangible Interaction: For spatial tangible interaction, the tracking accuracy during physical interaction decides the overall performance. From our task evaluation with users, we noticed that the tracking accuracy with the physical object improved from experimental results due to the user's tendency of interacting near the mobile device. We presume users prefer the near-surface interaction in order to maintain the visibility of the mobile screen. This implies that the 3D mobile input offered by TMotion has a potential to provide spatial inputs for tangible interaction.

Real-time Registration in Augmented Scene: Registration of the virtual contents to the physical input device in the augmented scene is particularly important to seamlessly connect virtual and physical worlds. In this work, we successfully register the virtual and the physical pen tips by translating the tracked pen tip from the magnetometer's frame to camera frame and scaling the interaction volume to fit into the video scene. Furthermore, we use the camera's pose estimation to superimpose the virtual contents to the physical environment. Through examples, we successfully showcase using the physical 3D input device freely manipulates virtual contents in AR environment. This implies that TMotion could potentially serve as an interaction medium to support upcoming mobile AR interface.

There are several limitations to the current version of TMotion. First, our approach requires subsequent maintenance of the device's orientation after an initial calibration. We plan to solve this issue with orientation estimation using either extended Kalman filter or magnetic dip angle detection where both methods work even under magnetic perturbation [195].

The interaction volume is still limited under 100mm above and behind the device. Simply increasing the strength of the magnet does not enlarge the interaction space proportionally. We tested our prototype with a stronger magnet (ϕ = 6mm, 15T), but it created a large saturation near the sensor due to the strong magnetic field and lost the dipole characteristics from the length of the magnet. However, this would be remedied by using upcoming magnetometers that have higher magnetic field sensing resolution and range.

Future work will include further expansion of applications into both AR and VR fields. We are in process of enhancing the prototype to be compatible with different size of mobile devices including tablet and smartwatch. Extensive user studies with real applications using proposed technique are also within our interest. These works will explore how users perform and perceive 3D mobile input for upcoming interfaces.

3.5 Conclusions

In this Chapter, we present TMotion, an embedded 3D mobile input using magnetic sensing technique. With the known orientations from 9DOF-IMU, we explicitly solve the position of the embedded magnet through numerical solver. In our experiments, we have shown that TMotion achieves a real-time and accurate 3D tracking with an existing mobile device. We also verify that TMotion maintains tracking and targeting accuracy with real users. Example applications showcase the continuous/discrete interactions in expanded spaces. As 3D mobile interfaces develop, the needs for better method to handle and exploit richer user inputs also increase. We demonstrate that TMotion potentially fulfills these requirements by presenting a real-time 3D mobile input.

In Chapter 3, we mainly investigated a 3D input method which leverages the local spatial relationship between the tangible device and the mobile AR device. We adopted a magnetic sensing based enabling technology for 3D tracking of the input device relative to the AR device. In a later work, we further improved the magnetic sensing in other wearable form factors for interacting with the environment [198].

4. EXTENDING 2D TOUCH INPUTS TO 3D CONTEXT FOR AR CONTENT CREATION

Recent works have demonstrated that *see-through* MR can play a vital role in *in-situ* geometric design. However, most of these approaches use the physical environment mainly as a dormant container of digital artifacts rather than a source of inspiration for facilitating quick digital prototyping for design ideation. The physical environment often serves as a means for inspiring, contextualizing, and guiding the designer's thought process for expressing creative ideas through early design objects are frequently used as references to explore the space of novel designs [168, 182, 72]. Recent works [166, 187, 201] have shown that *through-the-screen* AR/MR can play a vital role in bridging the gap between the physical and digital worlds for creative expression of ideas. However, most of these approaches use the physical environment mainly as a dormant container of digital artifacts rather than a source of inspiration for facilitating quick digital prototyping for design ideation. The key potential value that AR/MR systems bring to design, is the integration of *reflection-in-action* [168] (creating on the physical world), *design inspiration* [77] (borrowing from the physical world) and *reflection-on action* [85] (looking at the physical world). In this paper, we explore this value through re-purposing the physical environment as a reference, context, and source of inspiration for quick idea generation in early design.

We present *Window-Shaping*, an approach that integrates sketch- and image-based [125, 133] 3D modeling approaches within a mixed-reality interface to develop a new design workflow(Figure 4.1). Using the Google Tango device, *Window-Shaping* leverages the RGB-XYZ (i.e. image and point-cloud) representation of a scene allowing users to create planar curves on physical surfaces and *inflate* them into 3D shapes. Using *Window-shaping* we demonstrate design scenarios include the use of everyday objects and low-fidelity mock-ups as design references, and exploration of novel designs by combining physical references from multiple sources. *Window-Shaping* both complements and extends existing

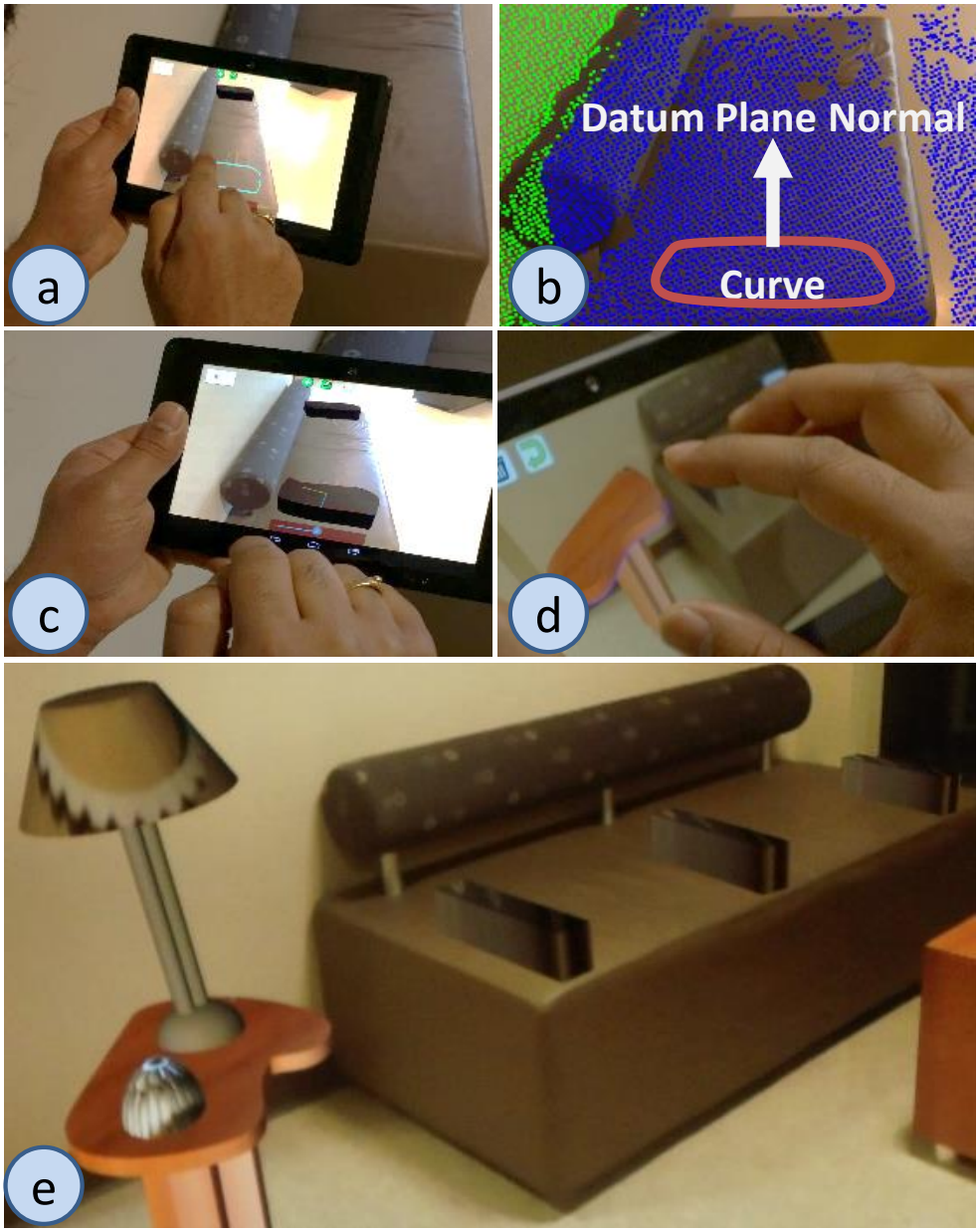

Figure 4.1. User simply draws a curve on the screen (a), that is mapped to a 3D planar curve using the point cloud (b). The 3D curve is inflated into a 3D model (c). Users manipulate the shapes through a multi-touch interaction scheme (d). *Window-Shaping* enables quick creation of virtual artifacts for augmenting the physical environment by borrowing dimensional and textural attributes from objects (e).

approaches [187, 108, 194] by exploring a new interaction metaphor wherein a new virtual 3D object is created as an extension of its physical context without the need for reconstructing the 3D model of the physical scene. We make two contributions:

Tangible In-Situ Design Workflow: We offer a novel combination of an existing modeling scheme with the synchronized RGB-XYZ information to enable creative design exploration with the physical environment in context. Our approach enables the creation, editing, and inspection of virtual objects directly at the desired position in the physical space. Further, our MR-based design workflow lends itself to a natural means to precisely edit shapes by simply moving the hand-held mobile device closer to a physical object.

Dimensionally consistent and visually coherent design: We offer the capability of creating and visualizing 3D shapes directly on the surface of any object with the desired dimensions and locations. Further, by mapping the background texture of the user's sketch inputs, we allow users to re-purpose existing textures in new creations.

Use cases: Using *Window-shaping* we demonstrate design scenarios include the use of everyday objects and low-fidelity mock-ups as design references, and exploration of novel designs by combining physical references from multiple sources.

4.1 System

The *Window-Shaping* interface comprises a hand-held Google Tango device, that serves as a local interface between the physical environment and the user. The simultaneous localization and mapping (SLAM) algorithm available with the Tango API allows for the acquisition of a point-cloud of the scene with respect to the global coordinate system. The resulting RGB-XYZ data allows users to implicitly define planes on any physical surface by simply drawing on the physical scene. Any touch input on the device screen can be *unprojected* on the physical environment to obtain a 3D point along with its normal in the global (world) coordinate system (Figure 4.1). This essentially helps users define a plane at any recognized point on a physical surface. Below, we describe the design goals, modeling metaphor, and user interactions.

4.1.1 Design Choices

We build towards the broad theme of "*effortless integration of physical objects into the design process*" [188]. The primary objective of our work is to support quick design ideation by allowing users to (a) quickly create 3D geometry in reference to physical artifacts, (b) borrow shape and appearance from physical artifacts to re-purpose them for design exploration, and (c) inspect the virtual artifacts in the physical context from different views in order to make design modifications. Our interface design choices are described below.

Appropriate Use of Interaction Modality: The use of Google Tango tablet allows for both spatial (3D) and multi-touch (2D) interactions. While touch-based interactions allow for precise control and 2D sketching operations, spatial mobility allows for *reflection-on-action* by enabling users to inspect their creations from multiple views with respect to the physical environment. Thus, we use well-established multi-touch interactions for enabling content creation, editing, and rigid transformations. For inspection, we make use of the natural spatial movement. With the augmentation from the see-through video and captured RGB-XYZ information, the traditional 2D interactions go beyond planes, and are enhanced with a third dimension. Further, this three dimensional extension of touch interactions provides a tangible and immersive experience for the design ideation.

Consistent dimension and appearance: The appearance of designers' creations is an important factor in reflecting their intent. Our metaphor enables users to both use physical objects as contextual references as well as re-purpose them at different physical locations. Further, our approach allows copying the texture of physical references for a consistent rendering of the newly created 3D shapes. Further, the editing operations are designed such that the changes maintain the consistency of visual appearance.

Geometric Modeling Scheme: We aim for flexibility in terms of the expressive power of the modeling scheme while retaining the simple interactions for shape creation. Multi-touch inputs naturally allow for 2D curve input. Thus, we employ a *sketch-and-inflate* modeling scheme in *Window-Shaping*. First, a user sketches the silhouette (and holes) of

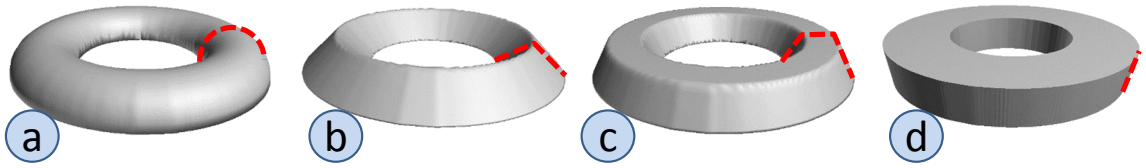

Figure 4.2. Geometric primitives with different inflation functions in *Window-Shaping* include: a. Circular, b. Conical, c. Tapered, and d. Linear.

the shape with which a closed mesh is generated. Then we inflate the mesh using a distance transform function [133] to obtain an inflated 3D shape. We provide users with four inflation functions (primitives) for expressive shape creation(Figure 4.2). The main advantages of this approach are that it: (a) has been demonstrated to be particularly simple for novice users [90], (b) allows for creation of complex topological structures with a simple set of interactions, and (c) has a simple and natural 2D parametrization that allows for texture mapping.

4.1.2 User Interactions

Shape Creation and Editing

Projective Sketching: *Window-Shaping* allows for direct one finger drawing on the tablet screen. Once finalized, the sketched curve is mapped on the physical scene and is converted and rendered in the scene as a 3D inflated mesh (Figure 4.3(a, b)). The first curve drawn by the user is by default the boundary curve. Multiple *hole* curves can then be drawn inside the boundary.

Placing Curve Template: As an alternative to direct drawing, we also provide a set of curve templates(Figure 4.3(c,d)). Users can simply place the selected curve on any surfaces of physical or virtual objects using a single-tap gesture. The curve is placed on a fitted 3D plane around the single-tapped location. The curve template feature allows for quick exploration of complex ideas with minimal interaction effort.

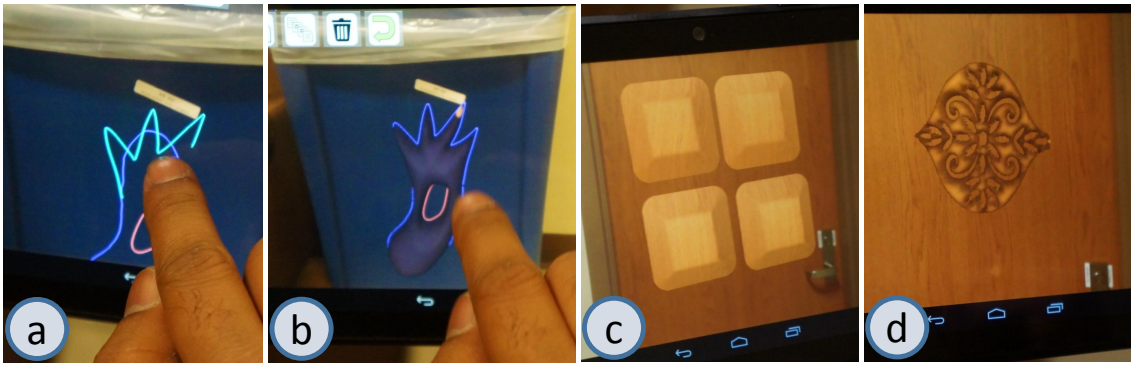

Figure 4.3. User sketches boundary and hole curves on a physical object, edit the sketched curves to add local details(b), and obtain inflated circular shape(b). The user creates a tapered inflated shape with a template, inflates it, and patterns it (c), explores complex feature(d).

Capturing Outlines: *Window-Shaping* also allows users to extract the outline of the object from the scene in *the image space*. Users draw a region of interest (ROI) which is automatically converted into a contour using the GrabCut algorithm [149]. This enables users to directly use the visual representation (outline shape and texture) of a physical object and re-purposing it in 3D form in their own designs.

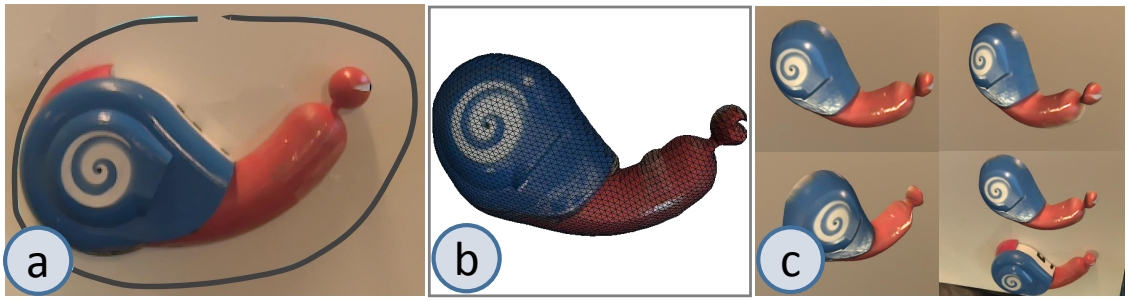

Figure 4.4. User use capture outline and texture of the snail shape: user draw ROI around the physical object(a), a circular inflated shape using captured outline(b), viewing the mesh together with real object(c).

Editing Curves: Using the over-sketching interaction technique [19], we provide intuitive and quick curve editing(Figure 4.3(a)) allowing users to add details and improve the appearance. Moving the tablet closer to a desired region allows for precise virtual operations in screen-space. On the other hand, moving away from a physical surface allows for better overview that is valuable for coarse operations such as placing shapes and curve templates on desired locations.

Inflating and Deflating: We implemented a three-finger gesture for inflating or deflating a 3D mesh. Here, pinching (bringing fingers closer) effects in pulling the shape out of the screen, and spreading (moving fingers apart) results in pushing the shape into the screen.

Manipulating Shapes

Rotations & Scaling: Two-finger rotate and pinch/spread are used for rotating and scaling the shape respectively. These gestures can be applied either directly to the 3D shape or to the underlying curve of the shape. The two-finger interaction constrains all rigid transformation the plane of the curve.

Translation: The in-plane translation is performed by dragging a shape using one finger. This allows for precise placement of the shape on the plane defined by its underlying curve. In order to provide consistent dimensional perception, we project the finger movement onto the underlying plane instead of using constant mapping between pixel space and physical space.

Placement: Shape placement allows users to directly transfer a selected 3D shape to any point in the scene by using a one-finger tap gesture. Here the 3D shape is both translated to the specified point and re-oriented along the normal at this point. Here users can place a new virtual object on the physical scene as well as on an existing virtual object. This maintains a perceptual depth consistency during interactions.

Auxiliary Operations: In addition to geometric operations, we provide operations such as copying/patterning and deleting a shape. Users can select and make copy/pattern of the shape by using the single-tap gesture.

Appearance Control

During the over-sketching operation, we automatically update the texture image to maintain the visual consistency. We also provide the option to explicitly update the texture during rigid transformations. This is helpful when users are experimenting with different backgrounds for the same shape.

4.2 Implementation

4.2.1 Hardware & Software

Our hardware comprises a Google Tango 7 inch tablet with the NVIDIA Tegra K1 processor and 4GB RAM, running Android 4.4 KitKat OS. The tablet captures the RGB image ($60Hz$) and depth data ($5Hz$) from the built in 4MP color camera and depth sensor respectively. The Tango SDK [62] provides functionality for synchronizing these two cameras, allowing us to compute a point cloud (XYZ) of the scene such that each point is mapped to a unique pixel in the RGB image. We prototyped our metaphor using the Android SDK and the geometric modeling methods in C++ using the Android NDK (JNI) with OpenGL Shading Language for rendering.

4.2.2 2D Curve Processing

We require all curves to be closed, oriented, and preferably smooth while preserving the features. To meet these requirements, we first apply an exponential smoothing filter [171] to each point on the curve as the user is drawing them. We then check the curve for closure based on the distance between the end-points, discarding an open curve as an invalid input. For a closed curve, we perform an equidistant curve re-sampling [103] and orient

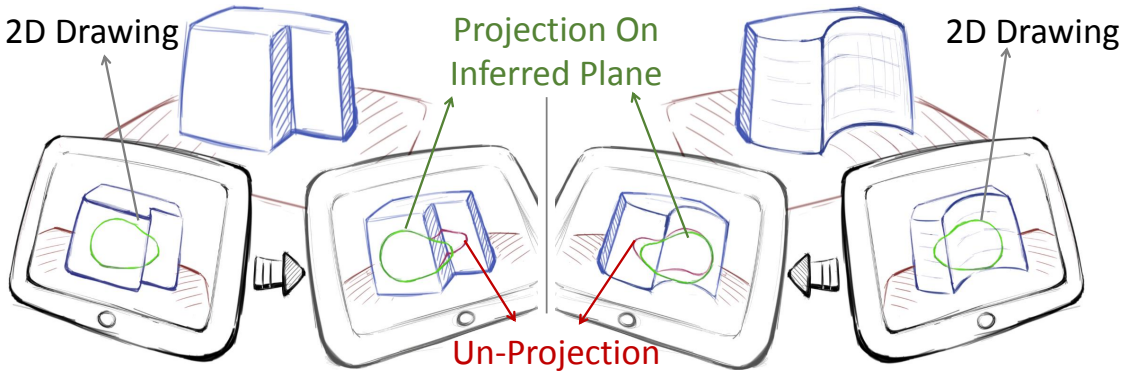

Figure 4.5. Illustration of plane inference: Un-projection of a 2D drawing results in a discontinuous curve (red) and a projection on the inferred plane(green)

the boundary curve counter-clockwise (i.e. positive area) with the holes oriented clockwise (negative area).

4.2.3 3D Planar Curve Computation

Given the processed curve on the screen, we first query the 3D points corresponding to each curve-point. For each 3D point we compute its normal by fitting a plane using its neighborhood. Based on the standard deviation of the distances between adjacent points on the 3D curve, we categorize the curve as either continuous or discontinuous. For a continuous curve, we estimate its plane by averaging the position and normals of these points. This, however, results in unpredictable planes for discontinuous curves (Figure 4.5). In this case, we first divide the curve into segments belonging to the same plane using euclidean distances and normal differences and then select the largest segment to identify the plane.

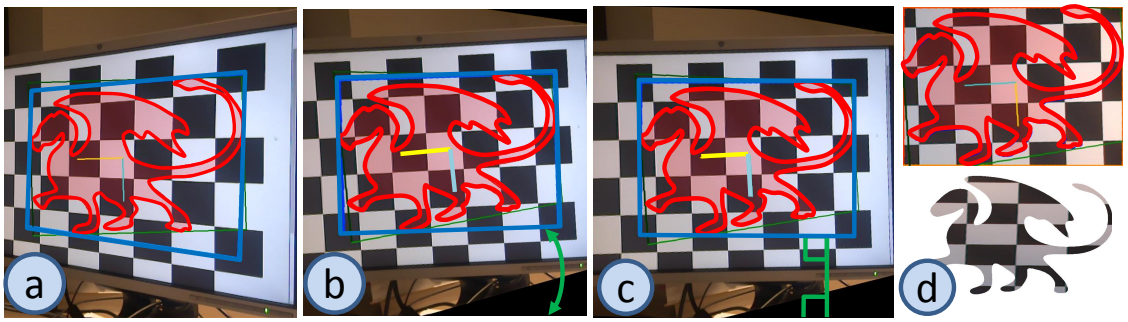

Figure 4.6. Creating a texture: (a) projecting the bounding rectangle (blue) of the 3D planar curve (red), (b) image skewing, (c) rotation correction, and (d) image cropping.

4.2.4 Mesh Generation

Given the processed boundary and hole curves, the mesh generation is performed in three steps: (a) computing two symmetrically aligned open (*half*) meshes bounded by curves (boundary and holes) through constrained delaunay triangulation (CDT), (b) topologically stitching these two open meshes to create a closed mesh, and (c) inflating the *top half mesh* using the distance transform function [133]. We implement CDT using the poly2tri library [51]. For the round, conical and tapered primitives, we sample the interior region of the curve with a uniform equilateral point configuration and add the sample as Steiner points to obtain a regularly sampled triangulation. Further, this modeling scheme has a simple and natural 2D parametrization which allows for texture mapping.

4.2.5 Texture Computation

We implement texture generation using openCV in four steps (Figure 4.6). We first compute the bounding rectangle of the 3D planar curve and project the bounding box on the image space. Then, we apply a skew transformation on this image with the constraint that the projected bounding rectangle is axis-aligned with the image. We rotate the skewed

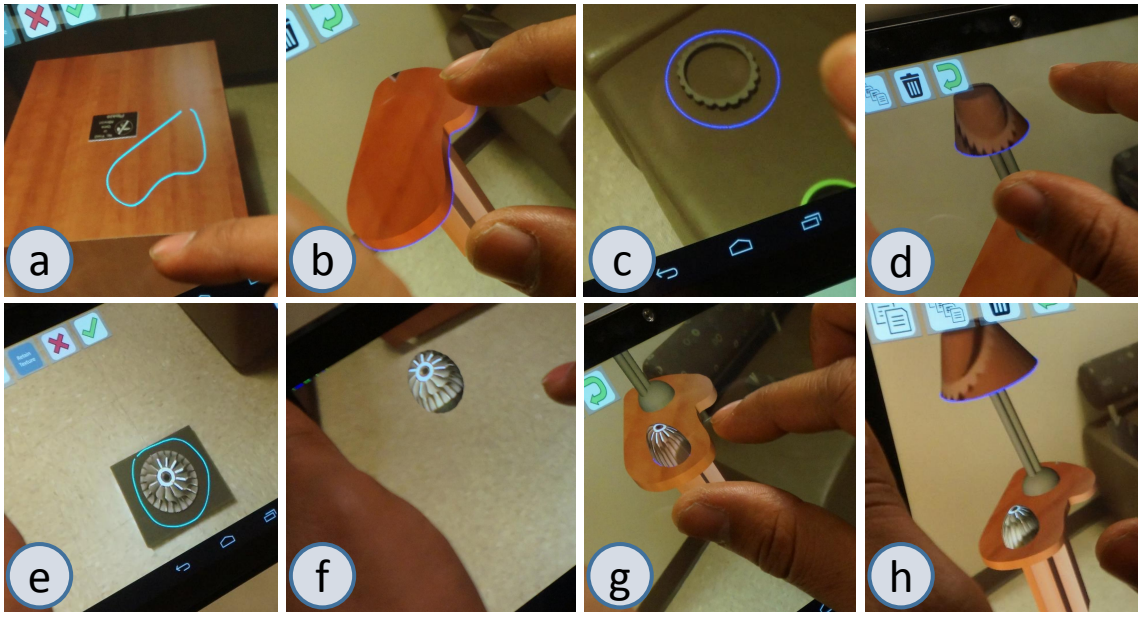

Figure 4.7. Furniture design: A virtual side-table is created by borrowing the texture from a physical table (a, b). The surrounding objects are then used to explore the lamp design (c, d) and *GrabCut* is applied to capture the outline and texture(e, f) to form a decorative object(g, h)

image to correct the residual angle between the rectangle and the image. Finally, we crop the image using this projected bounding rectangle to obtain the texture image.

4.3 Use Cases

The design work flow and interactions in Window-Shaping, can potentially be adapted to different kinds of design contexts. Below, we identify four such design patterns.

Designing on Physical Objects: The most important design capability offered by *Window-Shaping* is creating new geometric features on existing objects. These existing objects can be both physical and virtual objects. For instance, in an interior design scenario, a user could add complementary features to a piece of furniture (Figure 4.1, Figure

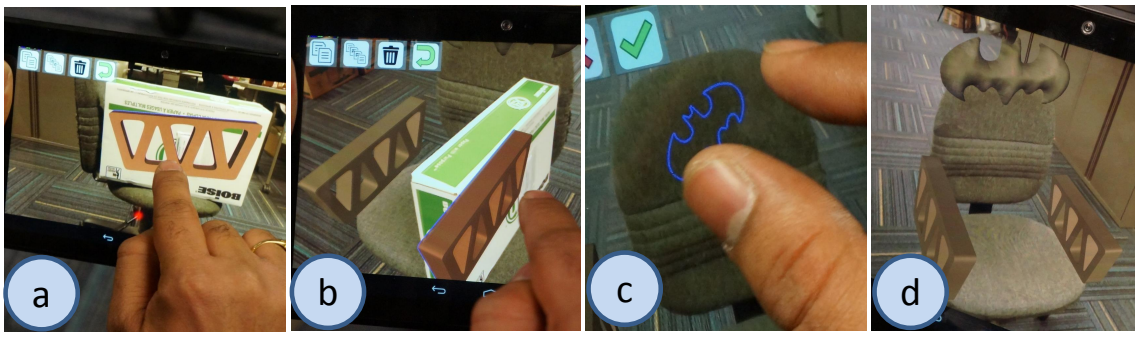

Figure 4.8. Chair armrest design: A *truss*-like shape is created on a metal shelf and placed as an armrest on the sides of a chair (a). A box is used here to appropriately position and orient the armrest with respect to the seat (b). Using a template (c), the back-rest is re-designed (d)

4.7(a,b)) and also create virtual additions to the scene by adding new assemblies to the surrounding area (Figure 4.7(g, h)).

Re-purposing Physical Objects: By re-purposing, we mean the use of both the shape and the appearance of a physical object to create a new design feature. The aforementioned GrabCut algorithm allows users to capture outlines of existing objects from the scene for shape creation. The captured outline shape and texture serve as design inspiration for direct use in an existing mixed-reality scene(Figure 4.7(e, f)).

Physical Objects as Spatial References: In situations where users desire to *fill in a blank space* to augment a physical product, it can be helpful to use a physical object to define a reference plane (Figure 4.8). Using objects as references enables a tangible and spatially coherent way of designing in context.

Physical Objects as Visual References: The appearance can serve both aesthetic and functional purposes (such as material specification). In *Window-Shaping*, users can experiment with the appearance of a 3D model. Such experiments can be performed either by transferring the virtual shape to a new location and re-texturing or by simply changing the background texture of a sketched curve (Figure 4.9).

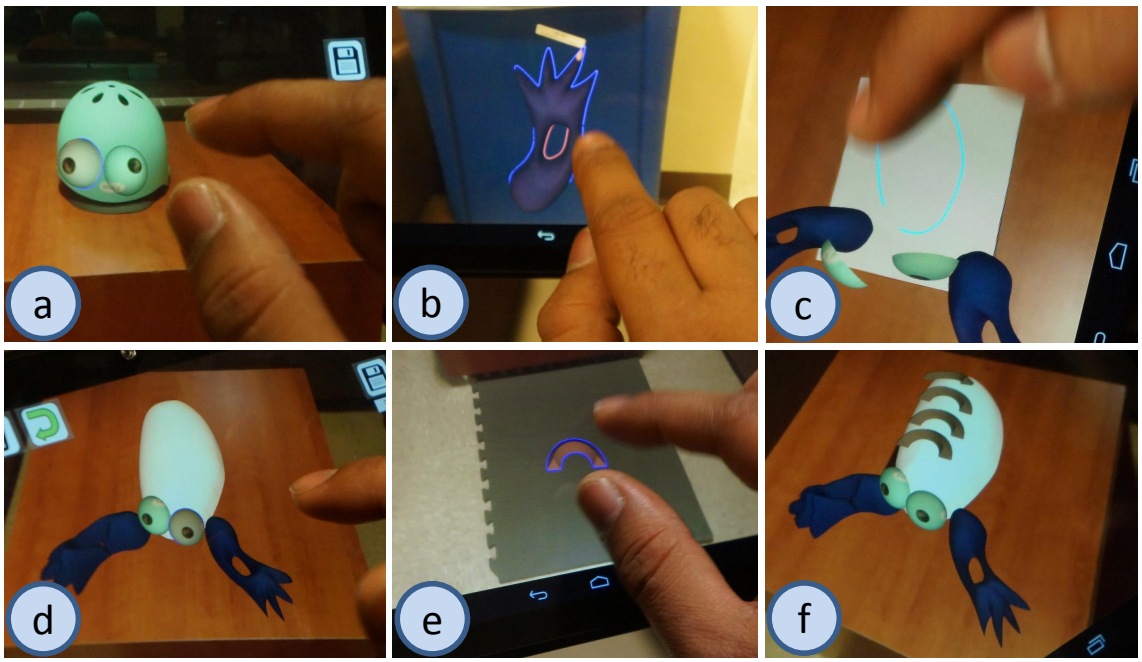

Figure 4.9. Creature design: The eyes (a), limbs (b), and body (c) are created using a helmet, a trash can, and a piece of white paper as visual references respectively. They are then assembled on a table (d) and the details (the scales of the creature) are created from a mat sheet and patterned on the body (e, f).

4.4 Evaluation

We conducted a preliminary evaluation for eliciting user feedback regarding the utility, experience, potential, and limitations of our workflow in creative design activity.

4.4.1 Participants

We recruited 8 (2 female, 6 male) participants ($26-30$ years old) with 5 engineering students and 3 students from non-engineering fields (science, management, etc). Of these, 4 participants had no prior knowledge of AR or MR interfaces, 3 were familiar with the concept of AR/MR, and 1 had used AR interfaces for gaming. Three participants had no prior knowledge of computer-aided design (CAD) or 3D modeling.

4.4.2 Procedure

We conducted user trials ($45-60$ min. per trial) with the *Window-Shaping* prototype application. We first introduced participants to the broader idea behind *Window-Shaping* and demonstrated the user interactions in *Window-Shaping* through practical use-cases (10 min.). Following this, we invited the users to perform four design tasks. We gave the users the option to either use a capacitive stylus or direct finger touch for tablet interactions.

(T1) Designing with physical mock-ups where the participants were given a cubiodal box and were asked to create the face of a creature. **(T2) Re-purposing objects** through which we introduced the grab-cut feature to the participants and asked them to create a 3D new part using *GrabCut* and add it to the face. **(T3) Using an object as a spatial reference** the participants were asked to design the handle of a chair using a cubiodal for placing the handle on the sides of the chair. **(T4) Using objects as visual references**, the users added details to the back rest and seat and explored the texture of the handles.

The participants were allowed to move around in the surrounding environment, create their feature on an arbitrary object and then transfer the feature back on the design. While we guided the participants through the interactions, we encouraged them to define their own strategy for completing the design tasks. At the end of the tasks, they were asked, through answering a questionaire, about their experience in using the interface (Figure4.10(a, b, c)). We also asked them to explain their reasons along with the Likert scales. Moreover, we asked three open-ended questions regarding the usability, potential use scenarios, and desirable capabilities.

4.4.3 Findings

Although we constrained the design tasks, we found that the resulting creations (Figure 4.10(d)) had reasonable diversity across users. Most of the users were able to quickly understand the modeling mechanism and successfully perform the trial tasks within the given time. Below, we discuss the main insights we gained from our observation and the feedback from the users.

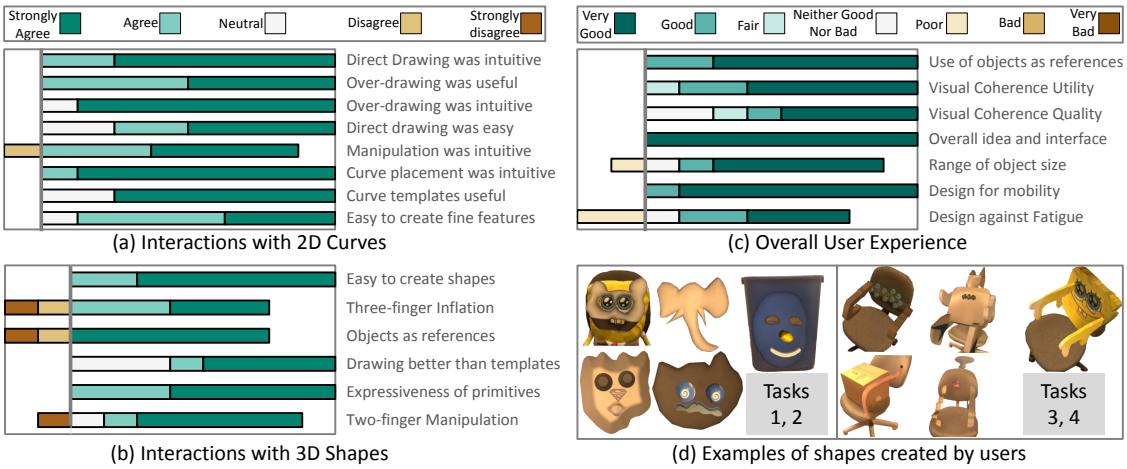

Figure 4.10. User feedback on interactions (a, b) and overall experience (c), and designs generated in trial tasks (d).

Validation of Interactions

All participants responded favorably to the ease of use offered by direct drawing and creating fine details and a majority preferred drawing more than using available curve templates (Figure 4.10(a)). Interestingly, there was a distinction between the engineering and non-engineering students in terms of the curve creation. Engineering students expressed a need for interactions such as mirroring, symmetric curve creation and editing. A user suggested: "*to enter some numeric values of the extrusion lengths*". In contrast, one non-engineering user commented on the curve templates provided: "*I did not use these features. It was intuitive to create something by myself*".

Most users found the interactions for placing (one finger tap) and manipulating (two finger rotate-scale) the curves and shapes to be intuitive. A user noted: "*Very similar to existing gestures on smartphones and tablets, so very intuitive*". Regarding shape inflation, while 6 out of 8 users agreed on the intuitiveness of the 3-finger gesture, 2 users expressed negative response to it. According to one user, "*It was intuitive, but the direction could be inverted*". This was contrary to our assumption in which pinching increased the infla-

tion and spreading decreased the inflation magnitude similar to in the physical action of deforming clay.

Mobility & Fatigue

One of the most important aspects enjoyed by all users was the capability to move around in physical space during the design process. This mobility offered the flexibility for users to borrow geometric and visual features from various locations (user comment: "*this help(s) you also to create shapes everywhere at any time*"). During the trial tasks, we observed that the users walked frequently within our constrained space to look for existing objects as references, to perform spatial inspection from different angles, and to design new shapes on reference planes. Further, some of the users naturally moved the device close to the object while creating detailed features on the curves. One user said: "*Bringing it [the tablet] closer did help a lot.*" This strongly suggests a positive outcome of our approach towards enhancing design ideation by increasing the reflection in and on the action.

While we expected this outcome, we were also interested in knowing how this mobility in our interface affected user fatigue. Six users agreed that using our system did not cause physical stress or fatigue that could adversely affect the design task itself. We believe this to be a result of the balance between the size ($119.77 \times 196.33 \times 15.36$ *mm*) and the weight (0.82 *lbs*) of the Tango tablet [62]. However, we did notice that our system caused fatigue specifically in the process of adding details to the curves for some users. A female user mentioned: "*While getting precise details for a long time it may cause the arm to be a bit tired.*". Last, although we offered the users two choices between using a stylus and their fingers for touch inputs, we got no particular comments on this through the survey.

Design Workflow

Users responded favorably in terms of the expressiveness, engagement, and enjoyment provided by *Window-Shaping*. In particular, the user feedback strongly validated our primary goal – quick design ideation through dimensionally consistent and visually coherent

shape modeling. In the given design contexts, all users except one agreed that our geometric primitives were expressive and allowed for sufficient scope for creative exploration. As expected, users generally favored the fact that our system allows capturing the background textures and can be used in real time. A user reported "*[without using our system] it will be difficult specially for getting textures or for placing the features on the desired position.*" Also, users tended to inspect the dimensionality of the creations from different perspectives. One user stated: "*The scale of the object with respect to the surroundings could be easily found.*"

Utility & Potential

Users confirmed *Window-Shaping's* utility as a quick ideation tool both for individuals and design teams. As one user said: "*The tool is definitely very useful for tinkering and showing new ideas to others.*" In particular, users with prior CAD experience appreciated the value of designing with a real scene in context. One user commented: "*You can visualize the components in the real world much more clearly than any CAD software.*" Another user pointed to the potential for collaborative design: "*I can see this being very useful in design collaboration where several people work on it at the same time. Very useful in modifying existing products and brainstorming what features to add.*". Participants also proposed some novel applications such as "*It's useful for topography when we need to study the landforms of some region*" and "*design your garden before planting*".

4.5 Discussions

User Interaction: Drawing while holding the device resulted in a lack of precise control. The participants found the curve editing to be more challenging in comparison with tasks such as placement. This primarily affected the addition of curve details. We believe this can be addressed through filtering out hand jitters by using the device IMU data and the camera image. Long term usage of the device may cause potential fatigue. Nonetheless, depth sensing technologies are emerging with more light-weight mobiles, such as *Lenovo Phab*

2 Pro [111]. Further, the texture capturing could be better implemented by allowing users to control the object segmentation. While users liked the idea of using physical objects as references, they pointed out a need for a mechanism to explicitly define a reference plane. This issue can be easily addressed by introducing *virtual plane* and *3D widget* elements into our interface. While previous work has shown better drawing accuracy for styli in comparison to fingers [15], it will be worthwhile to conduct more controlled experiments with a mobile system such as ours in terms of accuracy, fatigue, and stimulation.

Advanced AR Environment: While the robustness of the tracking provided by the Tango was impressive, our current implementation did not allow for object awareness. This resulted in shifting of the users' creations due to the frequent device movement. Introducing object tracking will reduce the tracking artifact when dealing with featureless scenes. Even though our interface offers mobility in a large indoor environment, it is currently difficult to create or borrow geometric and textural features from smaller objects. At this stage, the point cloud quality is limited by the low accuracy of the depth camera. Another issue in our current implementation is occlusion management with real objects. To address this issue, usually a fine model of the object and a high resolution of the depth data are required. With constant hardware improvements to the depth camera and mobile computational power in the future, we believe this issue will be easier to resolve.

Modeling Scheme: We restricted our implementation to the use of planar curves for inflating shapes. Although the current modeling scheme allows for reasonable expressive capabilities, there is a need to investigate the usage of 3D curves as well as different shape representations such as swept primitives, skeleton-based models, and volumetric representations. Towards a more complete and refined application, additional standard operations such as undo and redo will certainly enhance the modeling work-flow. Further, improvements in point-cloud acquisition will also allow us to extract and use manifold constraints (e.g. corners, edges, curvature) from the environment. The assembly mating relationships and kinematic constraints could then be leveraged for designing functional objects.

4.6 Conclusions

Window-Shaping reveals an untapped design space that emerges from the combination of multi-touch interactions, sketch-based geometric design, and mixed-reality interface towards bridging the gap between physical and digital space in early phase design exploration. Given the users' positive reactions, we believe that the proposed concept has potential towards a richer space of MR-based design work-flows for advanced in-situ modeling, collaborative idea generation, and fabrication-aware design. Although through the preliminary evaluations, we obtain overall positive feedback on the interface, the limitations of the current implementation need to be addressed before *Window-Shaping* reaches its broader potential. Based on the findings from our preliminary evaluations, we plan to improve the interaction metaphors against the jitter, accuracy and fatigue issues, and adding advanced geometric features into the existing modeling approach. Further, we plan to study how experience, performance, and creative outcomes will change with respect to different user groups such as artists, engineering designers, and young participants. Finally, it will be worthwhile to find how the interactions behind *Window-Shaping* could be extended to applications in domains such as architecture, education, animation, and engineering design and analysis.

In this chapter, we mainly developed an projection based approach to map 2D touch input from mobile AR device into 3D context. The AR application demonstrated here allows users to create virtual contents directly in the AR scene with high flexibility. We envision in pervasive AR, authoring easily-customizable AR contents plays an important role. We now extended the interaction volume from a local and small level as in Chapter 3, to a global and mobile level.

5. MAPPING SMART OBJECTS IN AR AND INTERACTING WITH THE SMART ENVIRONMENT

The ecology of connected smart devices is being rapidly interwoven with people's daily lives and work environments. People's vision of their surrounding physical world will largely be enhanced with the digital intelligence that comes through ubiquitous computing [138]. However, accessing and interacting with the Internet of Things (IoT) remains challenging due to the increasing diversity and complexity of the connected devices [18]. Traditionally, the digital interfaces of the interactive devices have been realized with a self-equipped touch screen display which has a limited adaptability. But now, contemporary IoT devices allow users to access full functionalities remotely by using an offloaded or duplicated interface on a smartphone. Still, in order to discover and access the devices, users need to browse through a specific webpage on-line or search for the corresponding applications. To alleviate the cumbersome processes, we leverage the spatial information of the devices relative to the environment and propose a physical browsing approach with AR.

As a novel interface which bridges the real and the digital, Augmented Reality (AR) has become a promising surrogate for interacting with the proliferating smart things [78, 113, 120, 159]. By superimposing the graphical digital interfaces on the physical world, users are exposed to the functionalities of the devices together with their physical affordance. This way, users are able to directly and intuitively access the smart environment. Moreover, the emerging visual SLAM technique allows a mobile AR device spatial awareness within the surrounding environment. Further spatial references based interaction metaphors can be realized in AR [60, 66, 109].

To this end, the key part of the workflow for interacting with the smart environment in mobile AR is mapping of the smart objects globally, i.e., knowing where the smart things are located in the AR scene. Simple scene augmentation has been achieved by detecting

Figure 5.1. *Scenariot* is a method for discovering and localizing IoT devices with a SLAM-based AR device. We embed UWB distance measurement units on the controllers of each IoT device. We register the discovered devices spatially in the AR scene to enable new spatial aware interactions.

the objects in the view of a camera. More recent works have shown progresses in multi-view object detection [139] and pose estimation [154] during consecutive movements of the camera. But, computer vision approaches largely rely on keeping the object of interest in the camera's view locally, which implies users already being aware of the identities and locations of the devices.

In contrast, we primarily aim at enabling AR interactions with the surrounding smart environment a whole ecology which requires discovering and localizing the smart things globally without prior location information of the devices. Wireless techniques such as Bluetooth, Zigbee, and WiFi allow for automatic discovery of the connected devices in an area network. Yet, a received signal strength indication (RSSI) based localization with the above technology suffers from low accuracy (from only a few meters) [4]. An accurate alternative utilizing Ultra-wide Bandwidth (UWB) based RF technology has been advanced and made accessible recently. Therefore, we develop a distance based localization method which integrates UWB based localization with SLAM to achieve quick mapping of smart devices spatially in the AR scene.

In this Chapter, we present *Scenariot*, an AR system which provides fast estimation of the 3D locations of smart things and exploits the spatial relationships discovered for

location aware interactions. To achieve this, we equip the IoT controllers and the SLAM based AR device with distance measurement units. The user carries the distant sensing capable AR device and surveys the surrounding environment while moving. We develop a distance based localization algorithm to estimate the positions of the IoT devices. By mapping the IoT devices into the coordinate system of the AR environment, *Scenariot* enables spatial context aware interactions instantly, including distant pointing, proximity based control, and visual navigation. Following is a list of the contributions:

- An approach to estimating the 3D locations of distributed smart things using a SLAM based AR device;
- Implementation and evaluation of hardware and software systems allowing users to rapidly map the smart things and interact with them in AR scenes; and
- Example applications demonstrating a wide range of usage of the proposed localization method and the enabled interaction metaphors.

5.1 System

We embed UWB units on IoT controllers and mobile AR devices. The distributed smart devices in the surrounding environment together with the AR device form a UWB network as shown in Figure 5.2. Unlike the conventional localization in wireless sensor networks where all nodes are stationary, we incorporate a dynamically moving node (the mobile AR device), along with a group of stationary nodes (distributed smart things). We are interested in finding the positions of the *stationary* nodes relative to the *dynamic* one. Due to the visual SLAM built in the mobile AR device which creates and updates a global map of the surrounding environment, the dynamic node is capable of real-time *self-localizing* on the map. We leverage the mobility and treat the *dynamic* node as a mobile *surveying* platform. Along the moving path of the dynamic node, we collect the distance measurements between dynamic node and each of the stationary nodes together with the positions at every measuring instance. We then employ the MDS technique and derive the 3D coordinates of

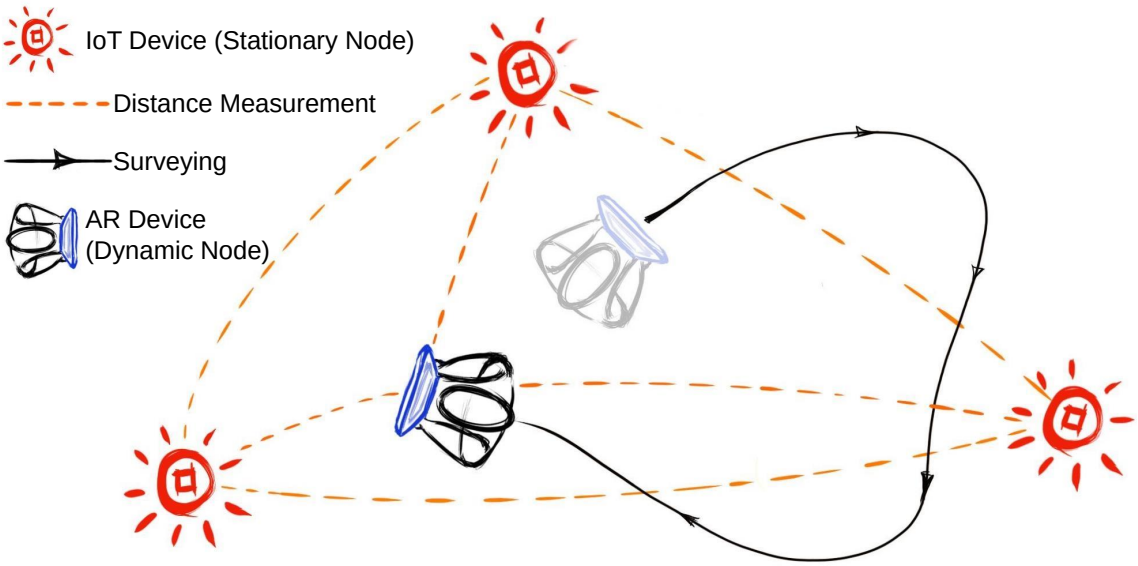

Figure 5.2. *Scenariot* localization principle.

the nodes in the coordinate system of the built SLAM map. Note that, in order to achieve 3D localization, we require 3D movements instead of planar ones from the dynamic node.

To this end, we discover and map the smart devices spatially in the AR scenes. It is worth noting that, the surveying movement only needs to be conducted once for an unknown environment. We store the 3D locations of the devices as well as the created SLAM map of the scene so that when users revisit the same region, the spatial registration is retained as long as the smart devices remain at the same locations and the environment has not changed much. We can render an augmented reality scene with the digital representation of the smart devices superimposed at the physical objects' locations instantly. By exploiting the spatial relationship between the user and the connected devices, e.g., distance, orientation, and movement, we further enable context aware in situ AR interactions.

5.1.1 Reviewing MDS Localization Principles

We first describe a traditional localization problem in a wireless network solved with MDS. MDS is a general technique which recovers the coordinates of a collection of nodes

by minimizing the mismatch between the measured distances and the distances calculated from the estimated coordinates [50]. Consider that we have N nodes to be localized in a fully connected network, in which the the Euclidean distance matrix across all N nodes is complete. We denote the coordinates as $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T \in \mathbb{R}^{N\times 3}$. The MDS algorithm estimates the relative coordinates of the nodes by minimizing the stress function $S(\mathbf{X})$:

$$\min_{\mathbf{X}} \mathrm{S}(\mathbf{X}) = \min_{\mathbf{X}} \sum_{i\leq j\leq N} \omega_{ij}(\hat{d}_{ij} - d_{ij}(\mathbf{X}))^2, \tag{5.1}$$

where $\hat{d}_{ij}$ is the distance measurement, $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$, and the weight ω_{ij} is defined based on the quality of the measurements. We denote a weight matrix $\mathbf{W}$ with the size of $N \times N$ which includes ω_{ij} as an element.

To solve this optimizing problem, an iterative method called "Scaling by MAjorizing a COmplicated function" (SMACOF) has been widely used with high guarantees and speeds of convergence [45]. We introduce a majorizing function as $\mathrm{T}(\mathbf{X},\mathbf{Z}) \geq \mathrm{S}(\mathbf{X})$ which bounds S from the above and touches the surface of S at $\mathbf{Z} \in \mathbb{R}^{N\times 3}$:

$$\mathrm{S}(\mathbf{X}) \leq \mathrm{T}(\mathbf{X},\mathbf{Z}) = C + \mathrm{tr}(\mathbf{X}^T\mathbf{V}\mathbf{X}) - 2\mathrm{tr}(\mathbf{X}^T\mathbf{B}(\mathbf{Z})\mathbf{Z}) \tag{5.2}$$

where the matrix element of $\mathbf{V}$ and $\mathbf{B}(\mathbf{Z})$ are defined as follows:

$$v_{ij} = \begin{cases} \sum\limits_{k=1,k\neq j} & -\omega_{kj} \quad if i \neq j, \\ \sum\limits_{k=1,k\neq j} & v_{kj} \qquad if i = j, \end{cases}$$

$$b_{ij} = \begin{cases} \sum\limits_{k=1,k\neq j} & \omega_{kj}\frac{\hat{d}_{ij}}{d_{ij}(\mathbf{Z})} \quad if i \neq j, \\ \sum\limits_{k=1,k\neq j} & -b_{kj} \qquad if i = j, \end{cases}$$

$\mathrm{T}(\mathbf{X},\mathbf{Z})$ is a quadratic and thus convex function [45]. Further, we compute the minimum of the function as:

$$\mathbf{X} = \min_{\mathbf{X}} \mathrm{T}(\mathbf{X},\mathbf{Z}) = \mathbf{V}^{-1}\mathbf{B}(\mathbf{Z})\mathbf{Z} \tag{5.3}$$

The SMACOF as summarized in Algorithm 1 [50], iteratively minimizes the majorizing function $\mathrm{T}(\mathbf{X},\mathbf{Z})$. After solving the MDS localization using SMACOF, we obtain the relative coordinates of the nodes. However, the absolute positions of the nodes are lost when

we rely only on the distance information. In order to recover the absolute positions fully, a set of at least 4 non-coplanar nodes (anchors) need to be localized a priori [50]. One common way to estimate the rigid body transformation, i.e., rotation-translation, between estimated coordinates of the anchors and the actual coordinates is by conducting a *Procrustes Analysis* [50].

Algorithm 1 SMACOF

1: SMACOF$\mathbf{X}^{(0)}, \mathbf{W}$
2: calculate $\mathrm{S}(\mathbf{X}^0)$
3: **while** $\delta \geq \varepsilon$ **do**
4: $\quad \mathbf{Z} = \mathbf{X}^{k-1}$
5: $\quad \mathbf{X}^k \leftarrow \min_{\mathbf{X}} \mathrm{T}(\mathbf{X}, \mathbf{Z})$
6: $\quad \delta = \mathrm{S}(\mathbf{X}^{k-1}) - \mathrm{S}(\mathbf{X}^k)$
7: **end while**
8: **return** $\mathbf{X}^k$

5.1.2 SMACOF with Mobile Anchors

Our problem formulation differs from the above traditional approach with in three ways: (i) our network incorporates stationary nodes and a dynamic node, namely the smart devices and the mobile AR device; (ii) no prior location information on the stationary nodes, i.e., no physical anchors available from infrastructure; (iii) we are interested in recovering the absolute positions of the stationary nodes using location information of the AR device in the SLAM map. Consider we have n stationary nodes in the network to be localized and m measurement instances to be sampled during the surveying using the dynamic node. We tackle these problems as follows.

- Due to the self-localizing capability of the dynamic node, we remove the dynamic node, meanwhile insert a group of mobile ”anchors” with known positions over a period of time to the network. We reinterpret this problem as a localization problem

in a fully connected network with a total number of $N = n + m$ stationary nodes: Given the positions of the m nodes and the full Euclidean distance matrix, we localize the unknown positions of n nodes.

- Leveraging the mobility of the AR device, we can introduce an arbitrary number ($m \geq 4$) of "anchors" with diverse configuration into the network. Basically, we eliminate the requirement for the fixed and previously localized physical anchors by incorporating a self-localizing dynamic node.

- A straightforward way of estimating the absolute positions is performing a full SMACOF in N dimension followed by a *Procrustes Analysis* with the anchors. However, we observed two coupled drawbacks: (a) the distances across m anchors should not contribute to the stress function; (b) the search space in SMACOF increases from a dimension of $\mathbb{R}^{n\times 3}$ to $\mathbb{R}^{N\times 3}$ unnecessarily. We incorporate the idea of partitioning [48] to resolve these issues.

We now explain the specifics of the modified SMACOF. We separate the set of nodes into "unknown" ($\mathbf{X}_u$) and "anchors" ($\mathbf{X}_a$) partitions:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_a \\ \mathbf{X}_u \end{bmatrix}, \quad \mathbf{Z} = \begin{bmatrix} \mathbf{Z}_a \\ \mathbf{Z}_u \end{bmatrix},$$

with,

$$\begin{aligned} \mathbf{X}_a &= [\mathbf{x}_1, \cdots, \mathbf{x}_n]^T \in \mathbb{R}^{n\times 3} \\ \mathbf{X}_u &= [\mathbf{x}_{n+1}, \cdots, \mathbf{x}_n + m]^T \in \mathbb{R}^{m\times 3} \\ \mathbf{Z}_a &= [\mathbf{Z}_1, \cdots, \mathbf{Z}_n]^T \in \mathbb{R}^{n\times 3} \\ \mathbf{Z}_u &= [\mathbf{Z}_{n+1}, \cdots, \mathbf{Z}_n + m]^T \in \mathbb{R}^{m\times 3} \end{aligned}$$

Similarly, we partition the weight matrix $\mathbf{W}$, as follows:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} \\ \mathbf{W}_{21} & \mathbf{W}_{22}. \end{bmatrix},$$

where block matrices $\mathbf{W}_{11}$ is of size $n \times\ n$, $\mathbf{W}_{12} = \mathbf{W}_{21}^T$ is $n \times\ m$, $\mathbf{W}_{22}$ is $m \times\ m$. We then simplify $\mathrm{S}(\mathbf{X})$ by reducing $\mathbf{W}_{22}$ to $\mathbf{0}$ because distances among the anchors contribute

nothing to the stress, followed by updating $\mathbf{V}$ and $\mathbf{B}(\mathbf{Z})$ accordingly. In the same way, we partition the auxiliary matrices V and B into block matrices. Further, we derive the partitioned $\mathrm{T}(\mathbf{X},\mathbf{Z})$, and differentiate it to solve the minimum of $\mathrm{T}(\mathbf{X},\mathbf{Z})$ [48]. Now we only account the nodes with unknown positions in the optimization procedure:

$$\mathbf{X}_u = \mathbf{V}_{22}^{-1}(\mathbf{B}_{22}\mathbf{Z}_u + \mathbf{B}_{12}^T\mathbf{Z}_a - \mathbf{V}_{12}^T\mathbf{X}_a). \tag{5.4}$$

We revise the Algorithm 1 with Eq. (5.4). Further, we lower the computation complexity by splitting the matrices and reducing the dimensions. This is important for us, because (i) we need to deploy the algorithm on mobile devices; (ii) in our formulation, the number of the mobile anchors (m) can be arbitrarily large. Moreover, this way allows us to estimate the absolute positions in a single step manner by incorporating the anchors' absolute positions directly in the SMACOF procedure.

5.2 Implementation

Our prototype is composed of IoT controller modules, AR devices, firmware running on the microcontrollers (MCUs), and applications installed on the AR device. The AR device works as a host to handle the localization algorithm and interface with IoT devices. As shown in Figure 5.3, IoT controllers are deployed to smart things as well as to the AR device. All of the devices connect to a network through WiFi. Moreover, each IoT device is capable of measuring distances to the others. Note that we use off-the-shelf components and design the hardware as a development board for prototyping purposes. We believe the package size can be greatly improved after iteration. We developed the firmware and the mobile application with reliability as our primary goal at this stage. Thus, there is a lot of room for improvement in the efficiency.

5.2.1 Hardware

As shown in Figure 5.3, the overall size of the board is $100mm \times 100mm \times 20mm$ with the units installed in position. This board is designed to process distance measurements,

deliver basic IoT functions, such as collecting sensor data and control appliances, and connecting with the smart environment network and the AR device over WiFi. The main MCU (Teensy 3.6) communicates with the DecaWave DWM1000 UWB module using SPI bus. Further it handles the WiFi communication by connecting a ESP8266 WiFi module (NodeMCU E12) via UART. The board also incorporates a set of general docking ports to interface with different IoT components such as sensors, and power relays. The board provides both 5V (1A max output) and 3.3V (1A max output) output from a rechargeable Li-ion battery (9V, 600mAh) using a dual regulator set. The battery lasts for $\sim$ 1.5 hours with a continuous two-way WiFi communication and a UWB ranging. Our localization method works on mobile devices supporting a SLAM based AR environment. For our prototype, we adopted ZenFone AR (ZS571KL, Snapdrago^{TM} 821 processor, Adreno^{TM} 530 processor, 6GB RAM) which is embedded with Google Tango technology [13]. We attach one of the self-contained boards on the back of the phone. Together, they serve as the dynamic node in the wireless network.

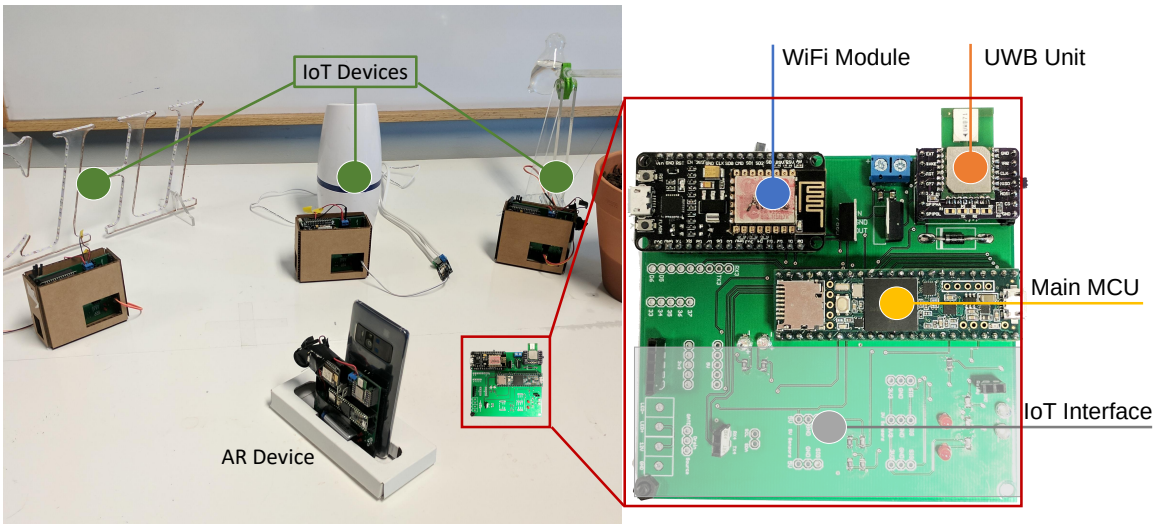

Figure 5.3. Overview of the *Scenariot* hardware. Deploy IoT controller board (right) to IoT devices and AR device (left).

5.2.2 Firmware

The firmware for the MCU is developed with the Teensyduino library and runs on an ARM Cortex-M4 chip (CPU speed 180MHz) that comes with the Teensy 3.6 board. The firmware mainly accomplishes the following tasks: (i) ranging to all available modules; (ii) connecting to a local area network through the WiFi module; (iii) communicating with the host AR device regarding localization and IoT function related messaging. Each MCU runs asynchronously with a tick function called from its own main loop and updates its state machine locally according to the tasks. We run a simple parsing and forwarding code on the NodeMCU chip after shaking hands with the main MCU. We support transmitting the distance data using User Datagram Protocol (UDP) via WiFi for high speed. We also support Transmission Control Protocol (TCP) if any IoT functionality requires large a file transmission.

5.2.3 Distance Measurements

We employ an asymmetrical double-sided two-way ranging scheme for time-of-flight ranging measurements between the IoT controller modules. This scheme is well known for correcting clock drift by exchanging two round-trip messages[94]. Although this approach is simple to implement, it works best for a small number of devices because it involves time-division multiplexing to range with multiple devices. We tune the tick timer in a conservative manner, which leads to an approximate upper bound of update rate $1000/(80+21n)$Hz for performing a *one to n ranging*. For our current prototype, we reach a ranging rate of ~ 3.7Hz when localizing a total number of 8 IoT devices at the same time. In this extreme condition, this update rate still allows users to move at a normal pace ($\sim$ 1m/s) without introducing many ranging errors.

5.2.4 Localization within AR

Our proposed localization method requires two groups of distance measurements: constant distances across all n stationary nodes, and continuous measurements from the dynamic node to the stationary nodes. We first poll a total number of $n(n-1)/2$ distance measurements across the IoT devices alternatively. Specifically, we perform $n-1$ times *one to i ranging*, where $i \in \{1, \cdots, n-1\}$. Then, during the surveying movements, the IoT module attached on the AR device collects the measurement instances and updates to the AR device using UDP. On Zenfone, the acquisition of the device position in the SLAM map is provide by the Google Tango API. We collect the position of the device when receiving a valid measurement instance. When the surveying ends, we launch the adapted SMACOF algorithm in a separate thread. Then after the algorithm terminates, we store the 3D location information of each connected device. For run time applications, we implement the proposed method on Zenfone. We developed the application within Unity3D [175] using C#. We employ an open source C# library Math.NET Numerics [142] to perform matrix calculations. To balance the computation resources and the localization accuracy, we empirically choose the number of samples from the surveying to be 100, the maximum iteration limit in SMACOF to be 500, and $\varepsilon = 1e-12$. This way, users spend less than 30s on the surveying. And running 500 iterations with 100 samples takes $\leq$ 10s to finish.

5.3 Technical Evaluation

To analyze the performance of our localization method in terms of accuracy, we chose to evaluate our method under several possible surveying conditions. We illustrate the setup in Figure 5.4. We divided the surveying conditions into two levels. The primary conditions including surveying distance (r), i.e., the distance from the center of the surveying space to the devices to be located, and the number of devices (n) to be located, The secondary conditions included the surveying space, number of samples (m) collected in surveying, etc. We defined the surveying space using the axes (x, y, z) aligned bounding box ($l \times w \times h$) of the sample points, which is centered at the origin. Note, the origin of the SLAM map

coordinate system was located at the point where the application launched. We launched the application with the phone being placed at a fixated location with a height of 1.5m (comparable to height of human body) above the floor.

In order to collect the data in a systematic manner, we decided to vary the primary conditions and fixate the secondary conditions when collecting the data. As shown in Figure 5.4, we conducted 9 surveyings to collect the surveying data with $r \in \{2,3,5\}$m and $n \in \{1,2,4\}$. For each surveying, we covered a sufficiently large survey space ($3 \times 3 \times$ 2m) and collected 3000 samples. To achieve a uniform sampling as much as possible, we held the device at different heights and walked within the surveying region with different directions. Since the AR device is equipped with a depth camera, we manually tagged the center of the IoT module as ground truth locations. We recorded the position of the IoT modules relative to the instant AR device location and transformed it to the SLAM map coordinate system.

We first studied the effect of the secondary surveying conditions on the accuracy, by fixating the primary conditions. Further based on the findings of the secondary conditions, we then evaluated the primary conditions and, designed studies with the suggested secondary conditions. For each studying test, we subsampled the dataset based on different conditions and fed the drawn samples to the localization algorithm. For evaluation purposes, we implemented the same algorithm with MATLAB and ran the algorithm on a desktop with a configuration of $\varepsilon = 1e-12$ and 500 maximum iterations. for all the experiments in this section. We used Root Mean Square Error (RMSE) between the localization results and the ground truth positions to indicate the accuracy.

5.3.1 Sampling Space

In order to gauge out the effect of the sampling space over the localization accuracy, we first assumed $l = w = h$, i.e., the surveying happening in a cube. We indicated the worst primary conditions as $r = 5$m, and $n = 4$, and the secondary condition as ($m = 100$). We varied $l = w = h = 1, 1.2, 1.4, 1.6, 1.8, 2$m to study the effect of the surveying space size on

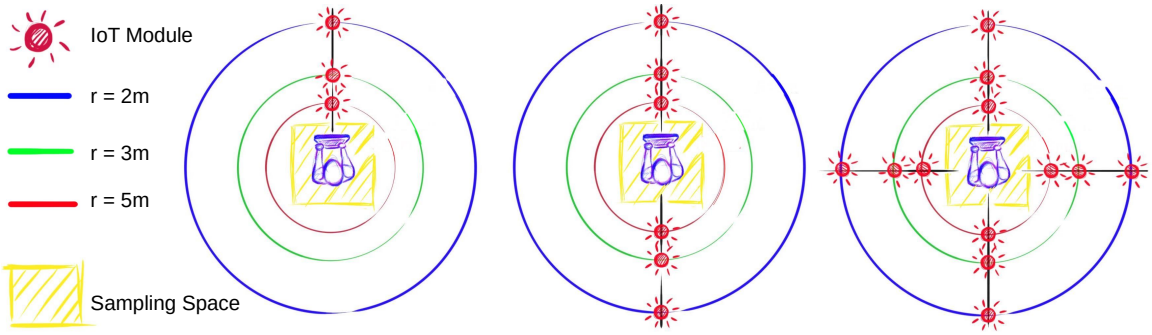

Figure 5.4. Technical evaluation setup. We varied the surveying distances (r) by distributing the IoT modules such that they were located on a circle with different radius (r) and a fixed height ($\sim$ 1.5m).

the accuracy of the localization. Then we randomly subsampled m points within the surveying space ($l \times w \times h$) from the overall dataset and fed them into the localization algorithm. We repeated the subsampling and localization 100 times for each of the variations. Then we took the average error of all 4 devices for the analysis. Last, we conducted a one-way univariate ANOVA and post hoc pairwise comparisons with Bonferroni correction. Overall, we found a significant difference across different survey space sizes ($p < 0.05$). Yet, within the set of $\{1.6, 1.8, 2\}$m, no significant difference was found ($p > 0.05$). As shown in Figure 5.6 (left), the mean error for $\{1.6, 1.8, 2\}$ was less than 0.5m which was less than 10% of the sampling distance $r = 5$m.

Second, reaching up to a large height limit involves an awkward motion. Taking into account a practical range of the arm motion without extra effort, we needed to study the effect of h on the accuracy with fixed l and w. Here, we varied $h = 0.8, 1.0, 1.2, 1.4, 1.6$m while fixating $l = w = 1.6$m. Other conditions remained the same as the first part. With the ANOVA test result, we found that there were significant differences across different height ranges, yet there were no significant differences across each other within the set of $\{1.2, 1.4, 1.6\}$m. From Figure 5.5 (right), we observed a mean error of 0.4m (SD = 0.1m) with a height range of 1.2m. To reach the range limit, an adult needs to fully stretch his/her

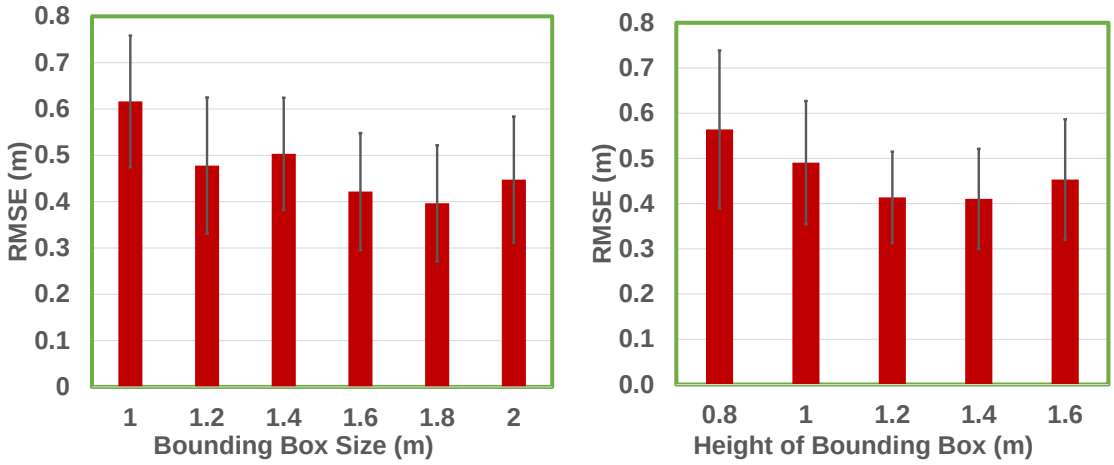

Figure 5.5. Effect of Sampling Space on the Localization Accuracy: assume a cubic volume (left), varying h and set $l = w = 1.6$(m)

arm up and down. On the other hand, even if there existed a degradation when $h \leq 1$m, we still observed a mean error ≤ 0.6m given the 5m sampling distance.

5.3.2 Sampling Number

We design the experiments in a similar way to the sampling space. We chose the primary condition as $r = 5$m, and $n = 4$, and the secondary condition as $l = w = h = 1.6$m, and vary the sampling numbers ($m = 20, 50, 100, 200, 300$) on the accuracy. From a one-way univariate ANOVA and post hoc pairwise comparisons, we concluded among $m = 100, 200, 300$ that there was no significant difference ($p > 0.05$), yet $m = 20, 50$ both showed significant differences with $m = 100, 200, 300$. From a computation efficiency point of view, we suggested to surveying with a sampling points number of 100.

5.3.3 Sampling Distance and Number of Devices

Based on studies on the secondary conditions, we set $l = w = h = 1.6$m and $m = 100$ to study the effect of sampling distance r and number of devices n. We designed this

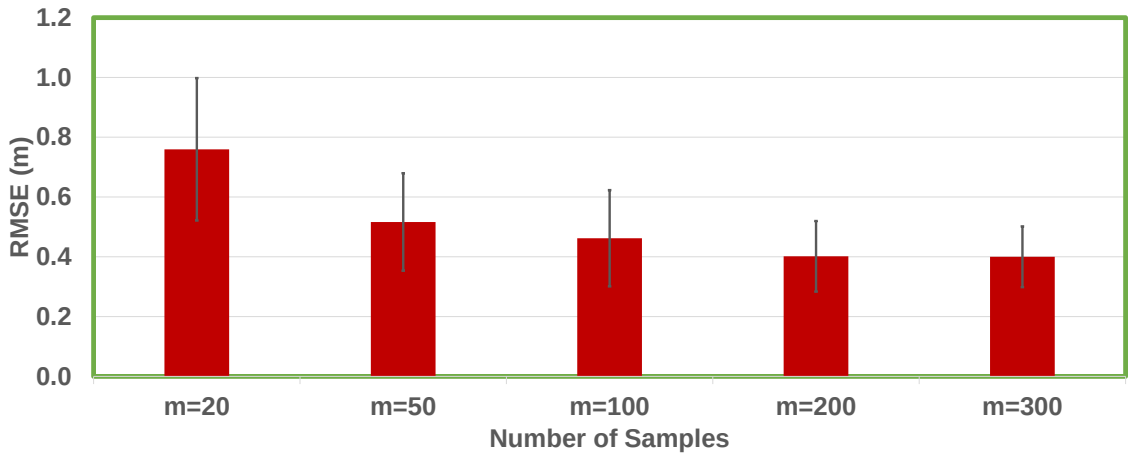

Figure 5.6. Effect of Sampling Number (m) on the Localization Accuracy.

study with variations of $r = 2,3,5$m and $n = 1,2,4$. We calculated the average errors for the conditions with $n > 1$ and used them for the tests. We conducted a two-way univariate ANOVA followed by post hoc pairwise comparisons. Overall, the ANOVA results indicated that both r and n were statistically significant ($p < 0.05$) over the accuracy. By examining the pairwise comparisons, we found out $n = 2$ and $n = 4$ showed no significant difference ($p > 0.05$) and that both of them were significantly different from $n = 1$. As shown in Figure 5.7, with the condition of $r = 5$m, $n = 2$ and $n = 4$ presented a larger error (> 0.3m). We observed that there was no significant difference between $r = 2$m and $r = 3$m, yet $r = 5$m yielded a significant difference from the others. From the figure, we confirmed that the mean errors of localizations at $r = 5$m increased but still remained < 0.4m.

5.3.4 Guidelines

From the study results, we summarized the following preliminary guidelines on utilizing the localization: (i) the surveying space should be sufficiently large ($l \geq 1.6$m, $w \geq 1.6$m, $h \geq 1.2$m); (ii) enough data should be sampled during surveying ($m \geq 100$); (iii) localization of multiple devices is feasible but likely to introduce more errors; (iv) the localization error increases as the IoT devices are located further from the survey re-

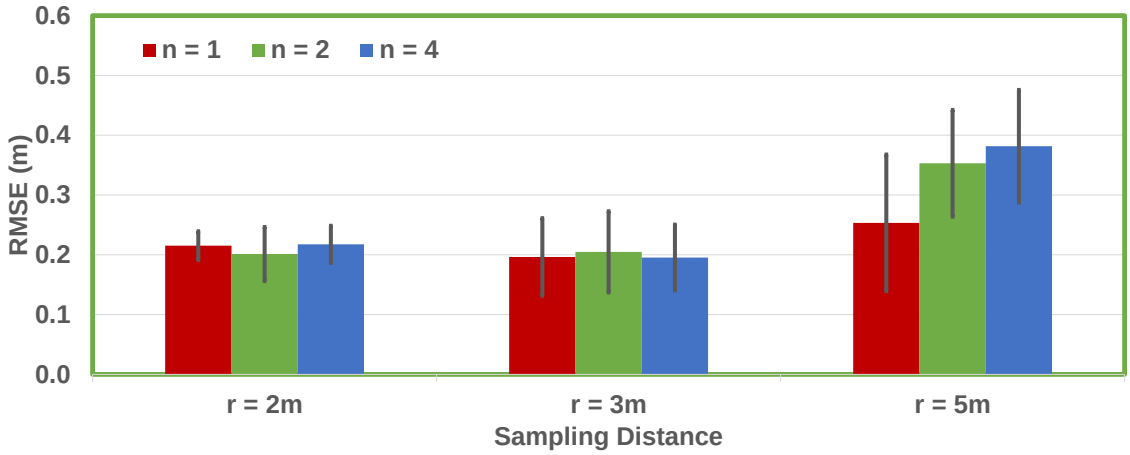

Figure 5.7. Effect of Sampling Distances (r) and Number of Devices (n) on the Localization Accuracy.

gion;(v)within a room of normal size($< 10 \times 10$m), surveying at the center of the room should localize the scattered devices with an average error at the level of $0.4m$ or less. With these guidelines, we further designed task evaluations and demonstration applications to verify our proposed method. Note that this technical evaluation was conducted with limited resources and so maybe less conclusive. This is why we suggest these guidelines conservatively.

Further, with the relative spatial relationships between the user and the IoT device, we extracted three basic spatial elements: the *orientation* of users with respect to the IoT devices, the direct *distance* measurement between a user and an IoT device, the *approaching direction* in which users walks. Based on these three relationships, we design and implement two location aware interactions, namely, distant pointing and proximity based control [109, 151]. In Task Evaluation section, we study the performance of these two widely accepted spatial interactions with users using our localization method.

5.4 Task Evaluation

Through the task evaluation, we expected to: (i) verify the localization performance with real users in a realistic scene; (ii) examine whether the localization performance meets the requirements of the spatial interactions in AR. We deployed 8 IoT controller modules onto 8 physical appliances in a cluttered office environment as illustrated in Figure 5.8. They were distributed within a region with a footprint of $\sim 10 \times 8$m at various heights. We also kept the existing common furnitures such as desks, shelves, and chairs in the testing area. Within this setup environment, we tested the localization accuracy by asking users to perform the surveying. After the surveying and the localization, we asked users to conduct these interactions. We then evaluated the performance in terms of targeting accuracy and completion time.

We recruited 11 participants with an average age of 25 for our study. Each user was asked to conduct a two-session study regarding the distant pointing and proximity based control respectively. Each session included 3 subtasks, where users first performed surveying movements then acted the designated interactions. Prior to the trial tests, we offered users a practice session to familiarize them with the system. We gave users a 5 minutes break between each session.

5.4.1 Localization Accuracy

For all 6 subtasks, users were asked to first perform surveying movements around the center area of the setup environment. We collected 6 sets of surveying movement trajectories and runtime localization results from each user, which resulted in 66 trials in total. After the surveying, the author manually tagged the ground truth positions of each IoT module as in the experiments in *Technical Evaluation* section. We displayed 4 progress bars on the screen to indicate the sampling number collected in the survey, as well as the expansion of the surveying space on 3 axes (x, y, z). We asked users to reach a minimum expansion of $l = w = 1.6$m, and $h = 1.2$m which was suggested by the technical evalua-

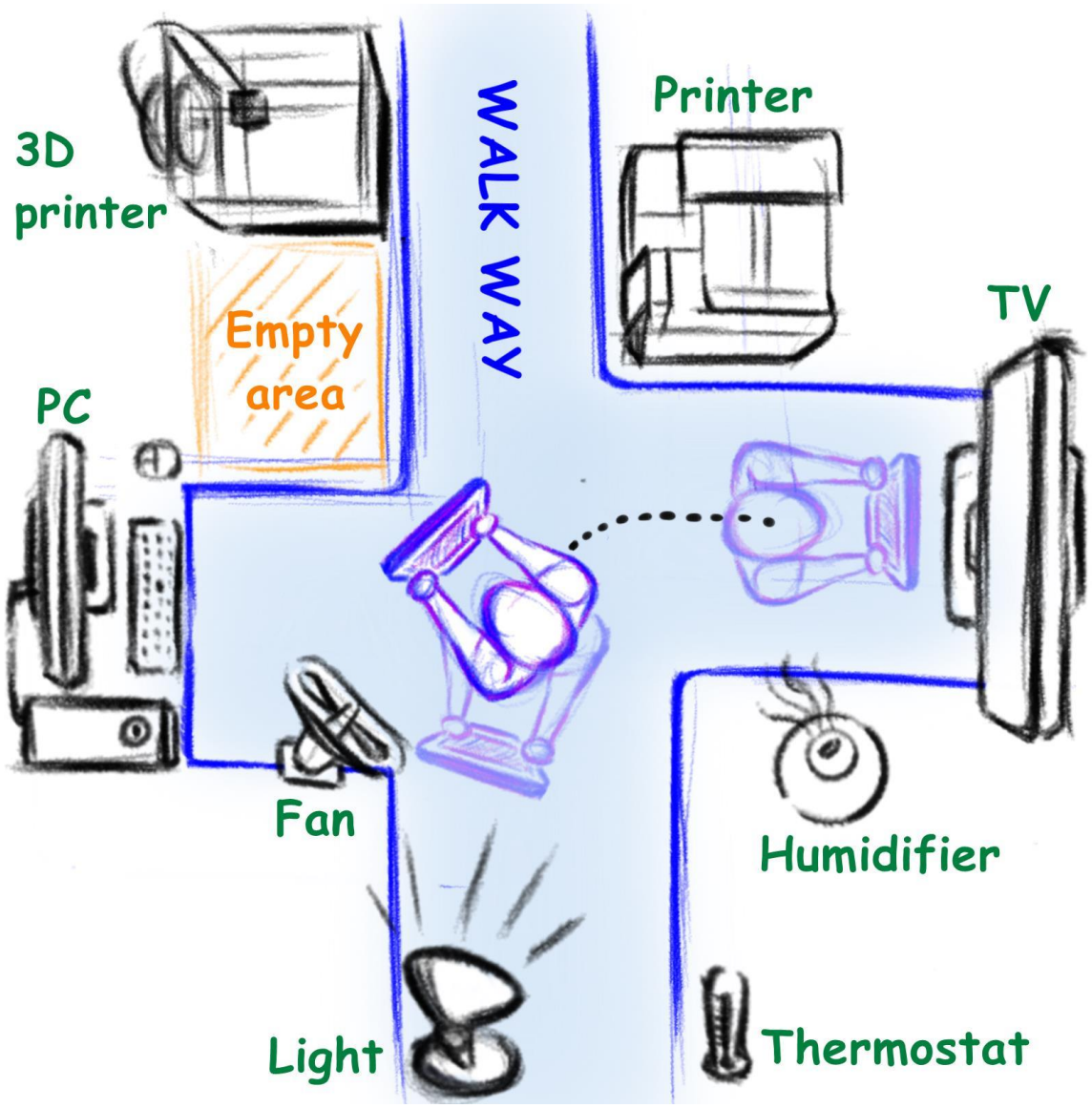

Figure 5.8. Task evaluation setup with 8 IoT devices distributed in an office environment.

tions. We did not ask the users to follow any specific trajectory as we tried to find out the possible performance degradations in the realistic scenarios.

Result As shown in the Figure 5.9, the average of the localization error over all 8 devices yielded 0.41m (SD=0.24m). We expected this result based on the technical evaluation results. We ran a one-way ANOVA to find out if the localization accuracy was similar

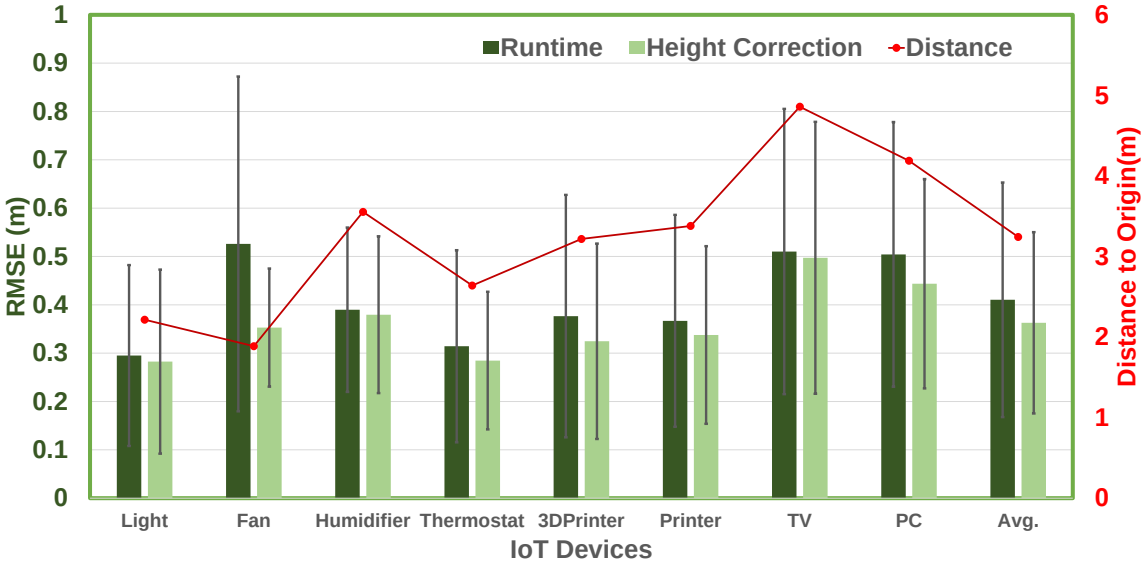

Figure 5.9. Localization Accuracy with Users. Runtime: runtime localization result. Height Correction: results with height correction. Distance: the distances of IoT devices to the center of the surveying space.

across different devices, and the result indicated the existence of significant differences. Further a post hoc pairwise test showed that the accuracies over Fan, TV, and PC were significantly different from those of the others. The TV and PC were placed at an extreme distance from the setup environment resulting in their being $\sim$ 5m and 4m away from the surveying region. As the distance increases, the localization performance may go down. Although the Fan was placed near the center ($\sim$ 2m), we deliberately left it on the floor under a desk. We suspect that the possible occlusion caused by the placement affected the localization performance.

We recalled that in the technical evaluations, the 100 samples were uniformly subsampled from the dataset. However, in real trials, we observed that users tended not to move much on the z axis. Instead of reminding the users, we let users freely perform the surveying. Sometimes if most of the sampled points largely lay approximately on a plane (horizontal), the flip ambiguity became more severe [16]. We inspected the collected data and found that flipping about an approximately horizontal plane happened occasionally

which introduced an error on the z axis mainly. We observed that the average localization error on z (0.32m) was larger than on the other two axes (0.13, 0.15m).

One way to compensate for the error caused by flipping was to incorporate some meta information about the IoT device. Here, we quickly implemented a heuristic leveraging of a rough prior knowledge about the devices. We dissected 3 levels on the z axis with respect to the floor, namely, lower ($0 \leq z \leq 1$m), middle ($1 < z \leq 2$m), and upper ($z > 2$m). We designated the IoT devices in this way: lower (Fan), middle (Humidifer, 3D Printer, Printer, and PC), and upper(Light, Thermostat, and TV). Compared with the runtime results in Figure 5.9, the overall average error decreased to 0.36m (SD = 0.19m). The T-Test showed that there was a significant difference between the runtime result and that the one with the heuristic (p ¡ 0.05) and thus indicated a decreasing trend on the localization errors over all 8 IoT devices.

5.4.2 Distant Pointing

Distant pointing leverages the orientation of the AR camera and detects if the object of interest is located in the center of the view window. We placed a virtual spherical collider at the location of the IoT module. Next, we dissected the spherical colliders by diameters (d) into three groups: small ($d = 0.5$m), medium ($d = 1$m), and large ($d = 1.5$m). Then we categorize the corresponding 8 physical devices based on their physical sizes: the PC, and TV as large, the 3D Printer, and Printer as medium, and the rest as small. We implemented a pointing scheme which performs AABB collision tests with a viewfinder frustum (8 degrees [8]) over the colliders.

Within each subtask, we generated a randomized sequence, where each IoT device appeared twice in the sequence. We randomly assigned the ground truth position or the runtime localization result to the colliders. For each trial, user oriented the device towards the object which was hinted at by its name according to the sequence. We suggested the users that they perform the distant pointing around the center of the setup environment though we do not limit their movements. We asked the user to place the whole physical

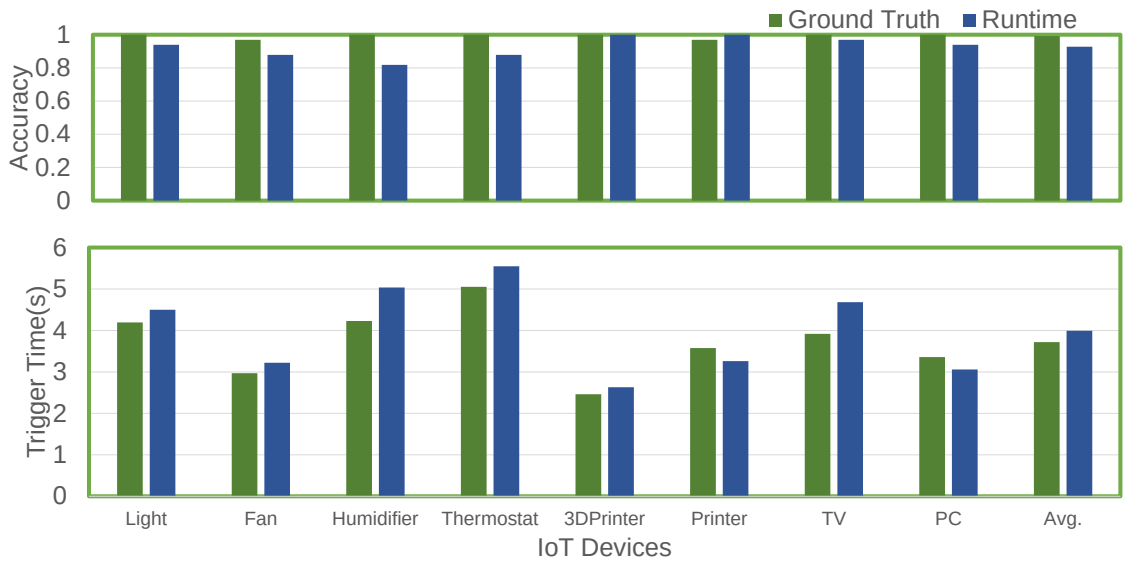

Figure 5.10. Distant pointing accuracy and completion time.

device at the center of the view as we applied an offset to compensate for the deployment displacement. We counted a device as being triggered if user pointed to the correct device within 10s and dwelled for over 1s. We counted a negative trigger if during the dwelling time, any other device mis-triggered as well. After each trial, we asked the user to fully disengage with all of the objects and point to some empty space as shown in Figure 5.8. In total, we collected $8 \times 2 \times 3$ distant pointing trials from each user, which resulted in 528 trials across all users.

Result As shown in Figure 5.10, we observed an average of 0.99 pointing accuracy with ground truth. We achieved an average of 0.93 accuracy with run time result, within which the Fan, Humidifier, and Thermostat had an accuracy less than 0.9. Compared to the localization accuracy shown in Figure 5.9, we suspect that the accuracy degradation was not just caused by the localization accuracy. We conjectured some potential reasons without verification: awkward installment positions, extra cognitive load from the cluttered scene, and selection ambiguities. For examples, the Fan was placed on the floor and the Thermostat was hanging around the ceiling, and the white Humidifier was hidden in a

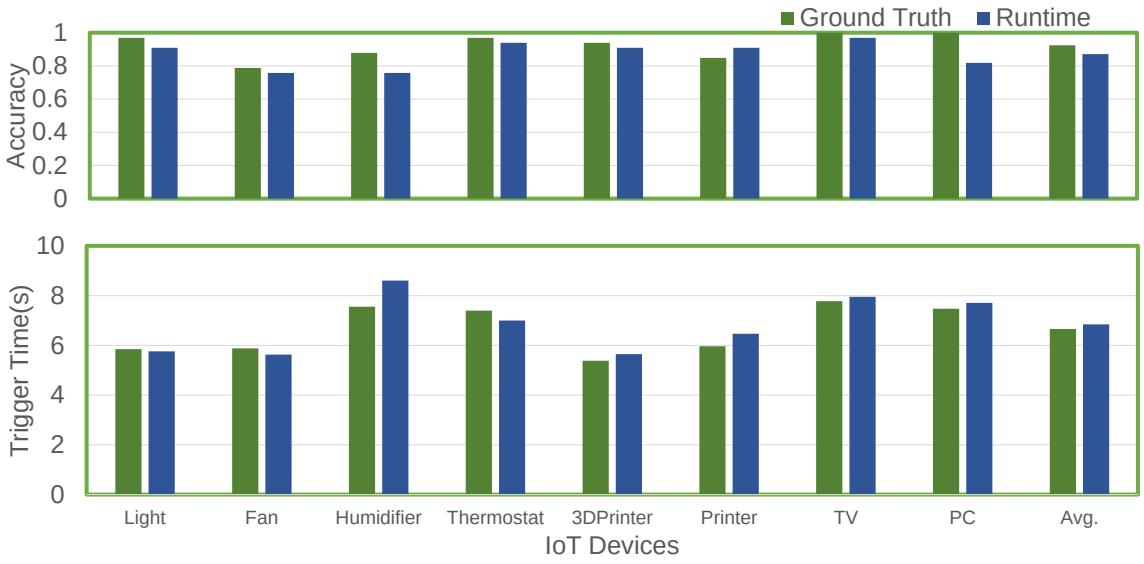

Figure 5.11. Proximity based control accuracy and completion time.

cluttered scene. In terms of completion time, we only counted the successful trials, and a T-Test showed there was no significant difference ($p > 0.05$).

5.4.3 Proximity based Control

Based on the proximic control framework proposed in a previous paper [109], we used three spatial elements for a proximity based interaction: *orientation*, *distance*, and *approaching direction*. The triggering conditions included facing towards, approaching towards and reaching into the proximity region of the hinted IoT device. The trial procedure was similar to that of Distant Pointing. We set a timeout limit of 15s, and we asked the users to return to approximately the same position to disengage from all of the objects.

Result The analysis showed an overall triggering accuracy of 0.92 with ground truth while 0.87 with runtime result. The ground truth accuracy suggested that we need to improve the interaction scheme. A paired T-Test on the accuracy between these two conditions indicated no significant difference($p > 0.05$). The accuracy with the results on both the Fan and the Humidifier were worse than others (< 0.8). We observed that users had unnatural

motions, including bending towards the Fan and detouring before approaching the Humidifier. Therefore, we need to adjust the interaction design according to the possible obstacles in the way of the target and the height of the object located there, i.e., it was too high or too hard to reach. For the completion time, the T-Test showed no significant difference between the ground truth and the runtime conditions ($p > 0.05$).

5.5 Example Use Cases

Based on the localization result, we register the IoT devices spatially in the AR scene which empowers the IoT devices to have the spatial awareness of the physical world. We foresee a wide range of flexibility and applicability using *Scenariot*. Here we selectively deployed *Scenariot* in 4 use cases.

5.5.1 Discoverable World

When a user enters a new environment, the AR device broadcasts a discovery message to the network then all connected devices send an acknowledgement and register with their identities. After the user localizes the IoT devices, the digital interfaces will be relocated to the discovered 3D positions. Users can simply browse the digitally enhanced world within the augmented scene. Inspired by previous works [66, 114], we further deliver a spatial aware picture-in-picture (PiP) effect. As shown in Figure 5.1 and Figure 5.12, we not only visualize the digital interfaces when the corresponding physical object is located inside the view, but also the ones outside. To achieve this effect, we parameterize the outside view space using spherical coordinates and shift the outside locations to the peripheral region of the view frustum. This way, we preserve the spatial information of the outside view devices.

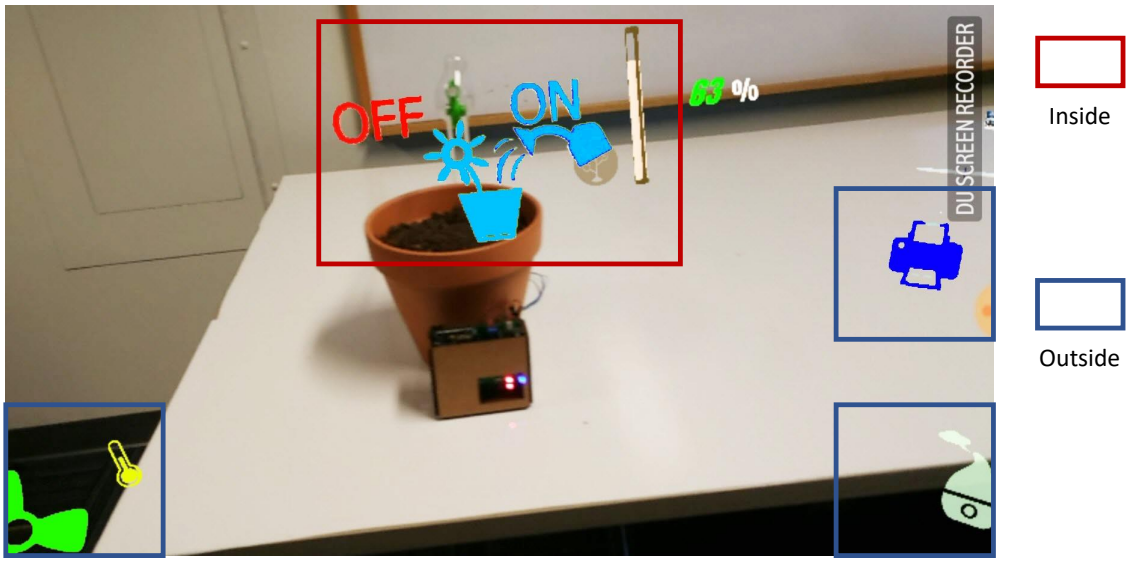

Figure 5.12. Discoverable World. The digital representations of the discovered IoT devices are visualized within the AR scene with spatial PiPs.

5.5.2 Proximity Based Control

This interaction scheme has been studied in the Task Evaluation section also. We here demonstrate *Scenariot* being used for fabrication machine inspections as shown in Figure 5.13. Users approach the machine to examine the status or operate it through the AR interface. As users move closer to the target, the digital interface adjusts according to the distances for different levels of engagement [109].

5.5.3 Monitoring Assets and Navigation

By attaching our IoT module to assets, we store the 3D locations of the assets together with the map created by the AR device for later revisit usages as in Figure 5.14 (a). When the user reenter one of the discovered scenes, we check the distance measurements across the IoT devices and/or between the IoT device and the AR device. If they do not match with the calculations based on the last location records, we consider the IoT devices as having shifted from their original locations. We provide suggestions for the user to conduct a

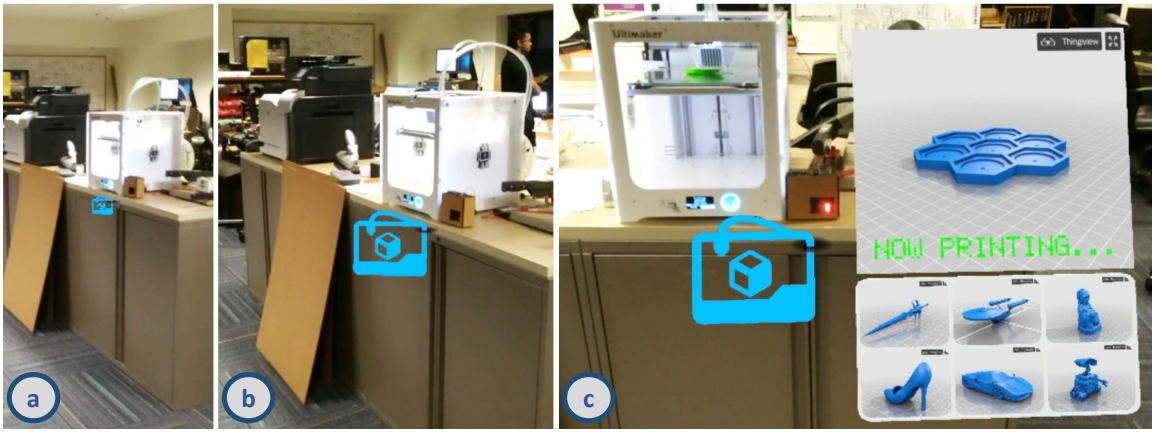

Figure 5.13. Proximity based Control. While users move closer to the machine (a, b, c), the level of engagement is adjusted accordingly.

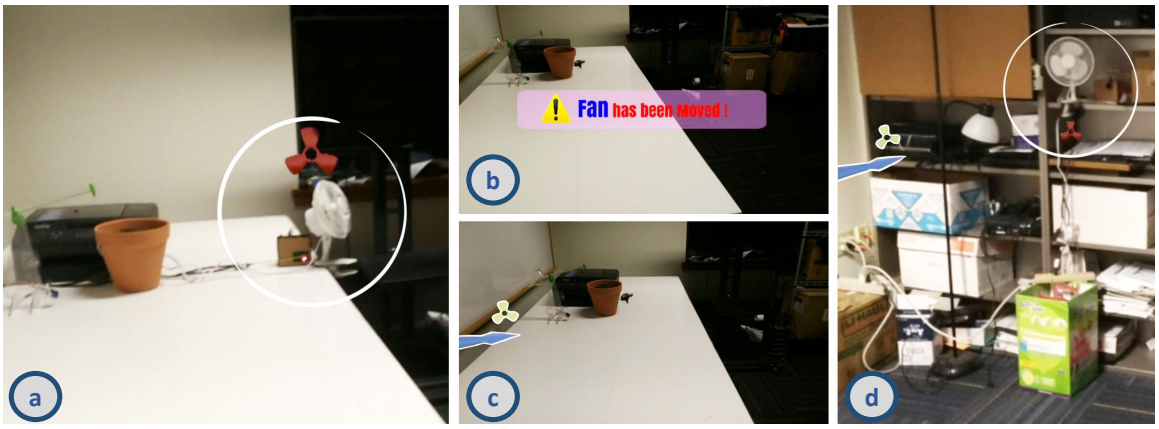

Figure 5.14. Monitoring the IoT assets (a, b) and navigating the user towards the assets by visualizing the direction on the screen(c, d).

new surveying(Figure 5.14 (b)). After the new locations are discovered, we navigate users towards the new position by showing them an direction indicator on the AR device (Figure 5.14 (c, d)).

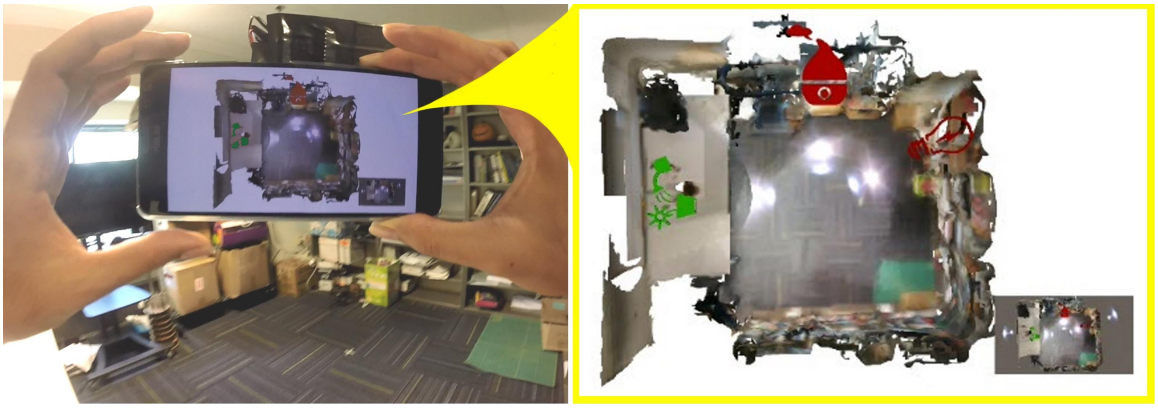

Figure 5.15. User creates a miniature world of the physical environment enhanced by the digital interfaces of IoT devices.

5.5.4 Miniature World

We consider another spatially aware interaction scheme for remote interaction with IoT devices, namely a miniature world [27]. For example, with the depth camera equipped with Zenfone, we allow users to scan and reconstruct the mesh model of the surrounding environment. We can combine the surveying stage with scanning movements. With the discovered 3D location information on the IoT devices, we superimpose the digital interface onto the virtual model. To this end, we develop an IoT-device-enhanced miniature world. Users can further interact with the miniature world to control the physical world.

5.6 Discussion and Future Work

More Spatial Interaction Metaphors With *Scenariot*, we can explore more spatial mobile interactions. We envision advanced inter-devices interactions can be realized with the given spatial information. Currently, we are only considering a single AR device to multiple stationary devices. In the future, we will consider to including multiple AR devices and dynamic IoT devices such as a service robot. Moreover, with the discovered locations, we can form an infrastructure based tracking by opportunistically referring the

IoT devices (more than three) as anchors [143]. Further, we plan to investigate incorporating multiple modalities for interacting with smart environment. For examples, we can leverage the spatial relationships to provide both visual and auditory augmentation [152], and context-awareness with voice command [136].

Improving Localization Algorithm With our current method, we assume a fully connected network, which means that the distances across all devices are available. However, in larger scale, this assumption may be valid. In the future, we need to further evaluate the effect of missing distance measurements on the localization accuracy. Although UWB provides a high distance sensing accuracy, heavy non-line-of-sight (NLOS) situations such as crossing walls needs to be identified and properly compensated [177] for a better accuracy. Moreover, we developed a heuristic for the flip disambiguation, yet we need a more general solution to resolve this problem [16].

Scalability In the Task Evaluation Section, we have evaluated the localization accuracy when deploying 8 IoT controllers to the environment. One essential bottleneck is the sampling rate of the distance sensing. We currently employ an asynchronized manner using 2 round-trip communications which is easy to implement but suffers from scalability. We are considering to employ synchronized manner which only needs 1 message for distance measurement to increase the sampling rate [4]. Also, the current centralized localization approach may suffer high computation costs for a larger scale deployment (e.g.,factories). We are also considering implementing our method in a distributed way for large scale adoption. Technically, visual SLAM and UWB should both work in outdoor environment, yet for this work, we only test in indoor environment. We would like to expand the study to outdoor setup in the future. Moreover, in real use scenarios, we need to address the heterogeneous interfaces with different IoT devices.

Form factor of AR device For prototyping purposes, we use Google Tango devices which are specially designed and embedded with SLAM, yet our localization method can be deployed to any moderate smartphones/tablets which are compatible with the third party SLAM based AR SDK (e.g., Wikitude [190], ARCore [63], etc). Further, integrating *Scenariot* with the emerging head mounted display based AR devices, e.g., Hololens [122] is

another alternative. Moreover, we received some feedback on the fatigue issue during the task evaluation. We expect to develop a compact version with optimized packaging.

5.7 Conclusions

We are building towards the broad goal of empowering users with the ability to quickly discover and intuitively interact with the connected smart things within the surrounding environment. We propose *Scenariot* as an approach to discovery and localization of the surrounding smart things along with spatially registering them with a SLAM based mobile AR system. By leveraging the spatial registration, in-situ AR interaction with the IoT devices is enabled. Through our experiments and user studies, we verified our method is capable of providing object level localization accuracy ~ 0.4m with multiple devices distributed in a cluttered scene with a normal size($\sim 10 \times 10$m). Therefore, we believe this work can bring spatial awareness to the IoT devices within an AR scene and further inspire advanced interaction designs.

In this chapter, we discuss an approach to discover the spatial locations of the smart devices in the environment and map them into the AR scene. This way, we enable spatial aware interactions with the smart environment. We demonstrate several user interfaces which adapt according to the spatial relationship between the user and the AR scene. To this end, we extend the intelligence of the spatial reference from a geometric level to a semantic level. The contextual awareness brought in the AR interfaces then contribute to realize the vision of pervasive AR.

6. INSTANT SYNCHRONIZATION FOR SPATIAL COLLABORATIONS IN AR

Emerging mobile technologies allow augmented reality (AR) applications to become pervasive [82]. Especially, the advancing simultaneous localizing and mapping (SLAM) technique extends the interaction volume into a highly spatial space by providing highly accurate tracking. With SLAM, a mobile AR device is capable of instant self-localizing with respect to the surrounding environment without external tracking setups and prior maps [63, 122].

Involving multiple users in a collaborative co-located environment requires synchronizing spatial frames across different users [20, 160]. This aspect is different from a single-user AR application. To overcome this challenge, researchers often introduce an external tracking system [158, 21] to establish a global shared frame. However, the cumbersome infrastructure counteracts the imperative mobility and immediacy of AR collaboration activities.

A contemporary approach leverages SLAM to create a map of the environment in-situ and share it across users either off-line or through a cloud service [10, 34, 124, 174]. Although this approach alleviates the restriction on mobility, it suffers from a laborious global map building process notably in a large space. Recently, researchers have proposed collaborative SLAM methods which automatically share the map in real-time as it expands [55, 83, 123, 148]. Yet, these methods require the users to start roughly at the same position with common views to synchronize the maps initially. This assumption markedly prevents a spontaneous collaboration as it requires specific positions and orientations to start the registration.

The state-of-the-art cloud based AR synchronization solutions essentially rely on a centralized data structure, i.e., a SLAM map contains one or multiple anchors or the full scan of the environment. Instead, we focus on instantly registering multiple SLAM based mobile

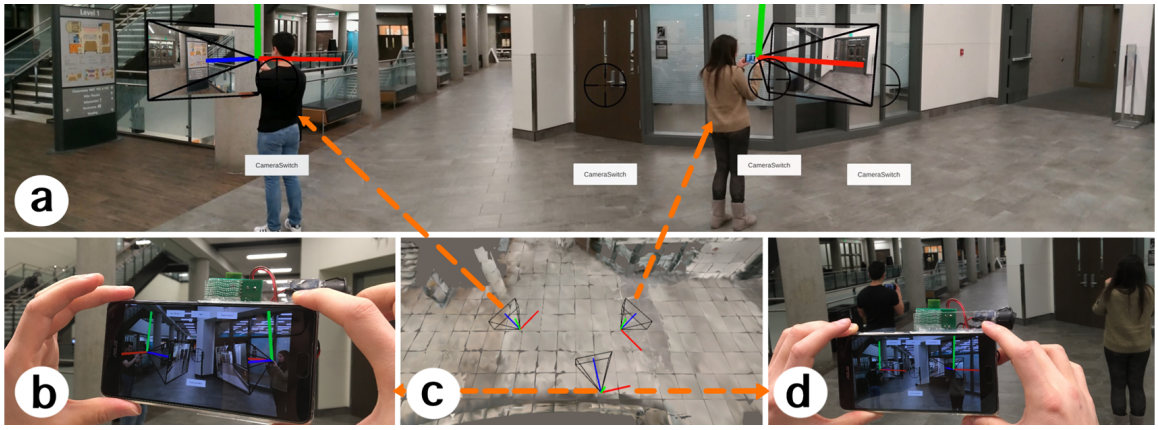

Figure 6.1. *SynchronizAR* allows for instant spatial registration among multiple users' mobile AR devices. Three SLAM based AR devices are registered with respect to each other (a, b, d). We enable AR collaboration activities such as spatial aware screen sharing (a) and miniature world navigation (c).

AR devices without sharing maps or using external tracking setups to support spontaneous AR collaborations in this work.

A direct approach to resolve the peer-to-peer 6 degree-of-freedom (DOF) transformation requires tracking the collaborator's device and estimating its full pose from the local device. One straightforward method is applying vision-based tracking using the embedded camera SLAM device. Unlike the traditional fiducial marker based tracking [11], recent learning based methods have achieved remarkable successes on human/object pose estimations [29, 139, 173]. Yet the vision-based approaches still rely on keeping the collaborator within the local camera view to estimate the pose and derive the transformation. Furthermore, the wearable or hand-held form factors of AR devices demands segmenting them out from images which involve human-device interactions [104, 170]. An electromagnetic based alternative suffers from bulky size and sensitivity to the magnetic distortion in the environment [17, 141, 165].

We present *SynchronizAR*, an indirect synchronization approach which leverages local SLAM results and radio-frequency (RF) based distance measures among the SLAM devices. While the multiple SLAM devices move on independent paths, the distance mea-

suring instances corresponds to the time varying positions in their local SLAM coordinate systems (Figure 6.1). Then we formulate a distance based registration to resolve the transformation across different local SLAM frames. In specific, we adopt the UWB based time-of-flight distance measuring, as it outperforms existing received signal strength indication (RSSI) based technique using Wifi or Bluetooth in terms of accuracy [118].

In summary, our registration follows a non-central approach by leveraging a self-contained hardware module (i.e., UWB). Comparing with the cloud-based synchronization, we better supports in-situ spontaneous AR collaborations.

1. More flexibility against a dynamic environment (e.g., lighting conditions, objects being moved) and zero cost when shifting to a new environment.
2. Less constraints on users' working zone as no "re-localization" is required.
3. Less dependence on cloud and network especially when Internet accessing is limited.
4. More compatibility across devices which normally don't share the same perception hardware, SLAM algorithms and map files.
5. Better supports on privacy control when the map contains sensitive information.

Here we list the main contributions of this work as follows.

- An approach to resolving the relative translation and rotation between SLAM based mobile AR devices utilizing UWB distance measurement units.
- Implementation of a spontaneous collaborative AR system enabled by the instant registration and evaluation of the system performance.
- Exploration and demonstration of enabled co-located collaborative AR activities with our prototypes.

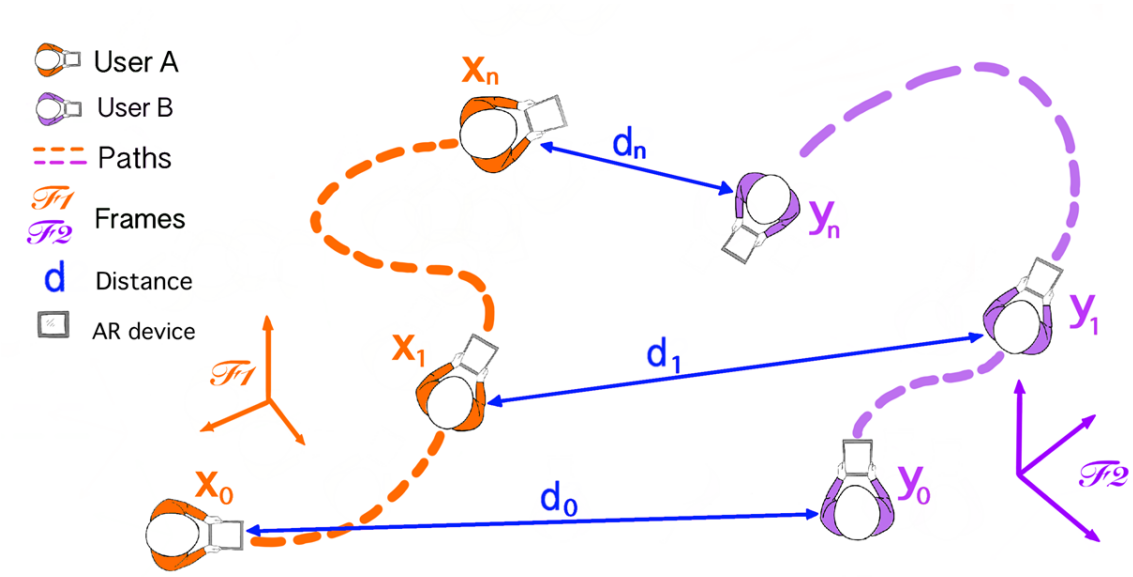

Figure 6.2. Registration between two users with *SynchronizAR*.

6.1 System

We introduce *SyncrhonizAR*, an approach to instantly register co-located multiple SLAM devices spatially with respect to a shared environment. It is an enabling registration technique which can be used to coordinate the collaborative AR interactions. We attach an UWB unit on each mobile AR device which is capable of self-localizing with respect to the environment using SLAM. During the registration, the AR devices move on different paths correspondingly, and the UWB units measure the distances among the devices as shown in Figure 6.2. We then derive the relative transformations by solving a distance based optimization problem. In this section, we first describe the general formulation to solve 6 DOF registration between two device. Then we adapt the method according to our realistic requirements.

6.1.1 General Formulation

In Figure 6.2, each user holds a SLAM based mobile AR device which is equipped with a UWB unit. As we are not sharing the SLAM map, the devices (A and B) yields two independent coordinate systems, i.e., $\mathscr{F}_1$, $\mathscr{F}_2$ respectively. Without loss of generality, the registration essentially resolves the translational ($\mathbf{T}_2^1$) and rotational ($\mathbf{R}_2^1$) transformation from $\mathscr{F}_2$ back to $\mathscr{F}_1$, e.g., ${}^1\mathbf{x}_i = \mathbf{R}_2^1\,{}^2\mathbf{x}_i + \mathbf{T}_2^1$. As the users are moving, ${}^1\mathbf{x}_i, {}^2\mathbf{y}_i \in \mathbb{R}^3$ denotes positions of A and B at time $t = i$ in their corresponding frames, i.e., $\mathscr{F}_1$ and $\mathscr{F}_2$. The distance between A and B at each time instance while they move on their paths is derived as follows.

$$\begin{aligned} d_i &= \|{}^1\mathbf{x}_i - {}^1\mathbf{y}_i\| = \|{}^2\mathbf{x}_i - {}^2\mathbf{y}_i\| \\ &= \|{}^1\mathbf{x}_i - \mathbf{R}_2^1\,{}^2\mathbf{y}_i - \mathbf{T}_2^1\| \end{aligned}$$

Within the time period $t \in \{1,\ldots,N\}$, we collect the local positions, ${}^1\mathbf{X} = [{}^1\mathbf{x}_1,\ldots,{}^1\mathbf{x}_N]^T \in \mathbb{R}^{N\times 3}$ and ${}^2\mathbf{Y} = [{}^2\mathbf{y}_1,\ldots,{}^2\mathbf{y}_N]^T \in \mathbb{R}^{N\times 3}$ for A and B respectively. At the same time, the UWB units measure the distances $\hat{d}_i$. Because of the distance errors introduced by the measurements, we formulate an optimization to estimate the transformations as follows.

$$\min_{\mathbf{R}_2^1,\mathbf{T}_2^1} \mathrm{S}({}^1\mathbf{X},{}^1\mathbf{Y},\mathbf{R}_2^1,\mathbf{T}_2^1) = \min_{\mathbf{R}_2^1,\mathbf{T}_2^1} \sum_{i\le N} \omega_i(\hat{d}_i - d_i({}^1\mathbf{X},{}^1\mathbf{Y},\mathbf{R}_2^1,\mathbf{T}_2^1))^2 \tag{6.1}$$

where the weight ω_i is defined based on the quality of the measurements. Note, in our current implementation, we simply set the weights equally to be 1.

6.1.2 Optimization with Reduced Dimensions

The general formulation of the problem requires to search solutions in a 6 dimensional space, as our unknown transformation has 6 DOF, i.e., 3 translational and 3 rotational DOF. However, with a close look at the SLAM system, we reduce the rotational DOF down to 1. Modern SLAM implementations on the of-the-shelf devices such as Google Tango and Hololens often leverage the built in inertial measurement unit (IMU). Such a visual-initial approach achieves a robust and accurate motion tracking.

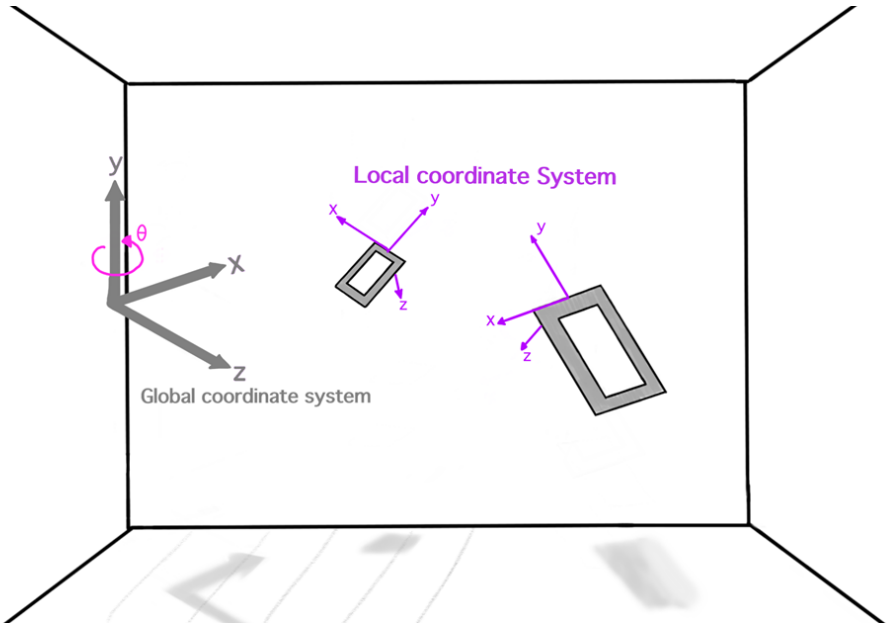

Figure 6.3. Coordinate system of a SLAM device.

As shown in Figure 6.3, when the device initializes SLAM, a world coordinate system will be created with an origin at the instant position. Also, the orientation of the coordinate system will be compensated by the IMU measurements at the moment so that the $x-z$ plane remains horizontal. That said, we only need to consider the rotation angle θ about y axis. Then we reduce the search dimension from 6 to 4.

Furthermore, we employ a heuristic to constrain the search space with boundaries on the translational y axis. First we observed a simple fact that when a user interacts with an AR device, the translational movements along y axis are limited considering ergonomic factors such as arm lengths and fatigues. Further, comparing with the movements on x and z axes which can easily reach to dozens of meters, the range on y axis appears a relative small level ($\sim$ 1 m). Besides, for a HMD, moving along y axis is obtrusive and unnatural.

However, for the distance based optimization problems, the flip ambiguity arises easily when the sample positions roughly appear on a plane which implies a irregular distribution, i.e. not a uniform distribution in 3D space [16, 86]. Our heuristic tackles these problems by taking the following steps: (i) initializing the SLAM at a fixed hight ($\sim$ 1.5m above the

floor), (ii) constraining the the movements on y axis during the registration, (iii) set the y components of ${}^1\mathbf{X}$ and ${}^1\mathbf{Y}$ to their average values respectively, and (iv) adding boundaries on $\mathbf{T}_2^1$ to the optimization solver. To this end, we adjust Eq. (6.1) a constrained optimization problem with reduced dimensions as follows.

$$\begin{aligned} \min_{\theta,\mathbf{T}_2^1} \mathrm{S}({}^1\mathbf{X},{}^1\mathbf{Y},\theta,\mathbf{T}_2^1) &= \min_{\theta,\mathbf{T}_2^1} \sum_{i\leq N} \omega_i(\hat{d_i} - d_i({}^1\mathbf{X},{}^1\mathbf{Y},\theta,\mathbf{T}_2^1))^2 \\ s.t. t_{y_{min}} &\leq t_y \leq t_{y_{max}} \end{aligned} \tag{6.2}$$

where t_y denotes the y component of $\mathbf{T}_2^1$, and $t_{y_{min}}$ and $t_{y_{max}}$ are boundaries of t_y.

6.1.3 Scalability

To this extent, we offer an instant registration for spontaneous collaborations between two users. For more than two users, we consider different situations: (i) multiple users form a new collaboration and (ii) one or more users join an existing collaboration. For the first situation, a total number of k users results $k(k-1)/2$ transformations, among which only $k-1$ transformations are independent. For example, with independent transformations $\mathbf{R}_2^1, \mathbf{T}_2^1$ and $\mathbf{R}_3^1, \mathbf{T}_3^1$, we can derive the homogeneous transformation as follows.

$$\begin{bmatrix} \mathbf{R}_3^2 & \mathbf{T}_3^2 \\ \mathbf{0} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_2^1 & \mathbf{T}_2^1 \\ \mathbf{0} & \mathbf{1} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{R}_3^1 & \mathbf{T}_3^1 \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{6.3}$$

We select $k-1$ independent transformations in a manner of one-to-many. Namely, we measure the distances from a single device to the rest of devices within the UWB network. Together with the corresponding local positions, we run $k-1$ times one-to-one registrations. Then we calculate all $k(k-1)/2$ transformations similar to Eq. (6.3). For the second situation, we select one node from the existing collaboration and perform a registration between the new users and this node only. Again, we propagate the rest of transformations similarly.

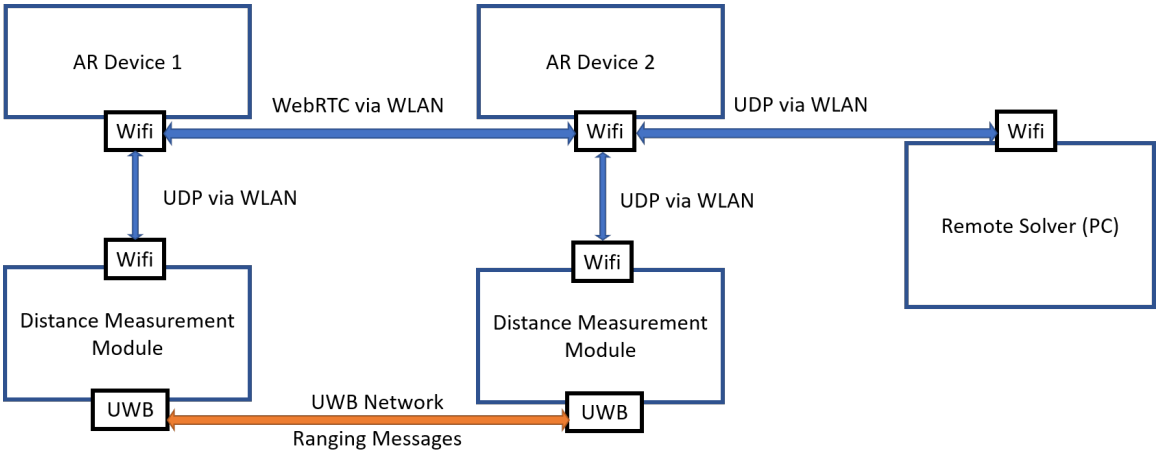

Figure 6.4. System overview of a prototype example with two AR devices and the distance measurement modules.

6.2 Implementation

SyncrhonizAR utilizes an indirect distance-based registration and requires no map sharing. As illustrated in Figure 6.4, our prototype system consists of AR devices, distance measurement modules, and a remote solver (e.g., PC) which were connected to a wireless local area network (WLAN). We developed the self-contained UWB based distance measurement module with off-the-shelf components. Through the UWB network, the distances were measured and packaged to an arbitrary MCU. Then the distance measurements were sent to the AR devices via UDP. The local coordinates of the AR devices have been shared through the WebRTC. Then, a remote solver fetched the sample packages which include the distances and the local coordinates by communicating with one of the AR devices through UDP. Our system supports heterogeneous SLAM based AR devices and corresponding SDKs [63, 122, 190] as long as we attach our UWB measurement modules onto them as shown in Figure 6.5.

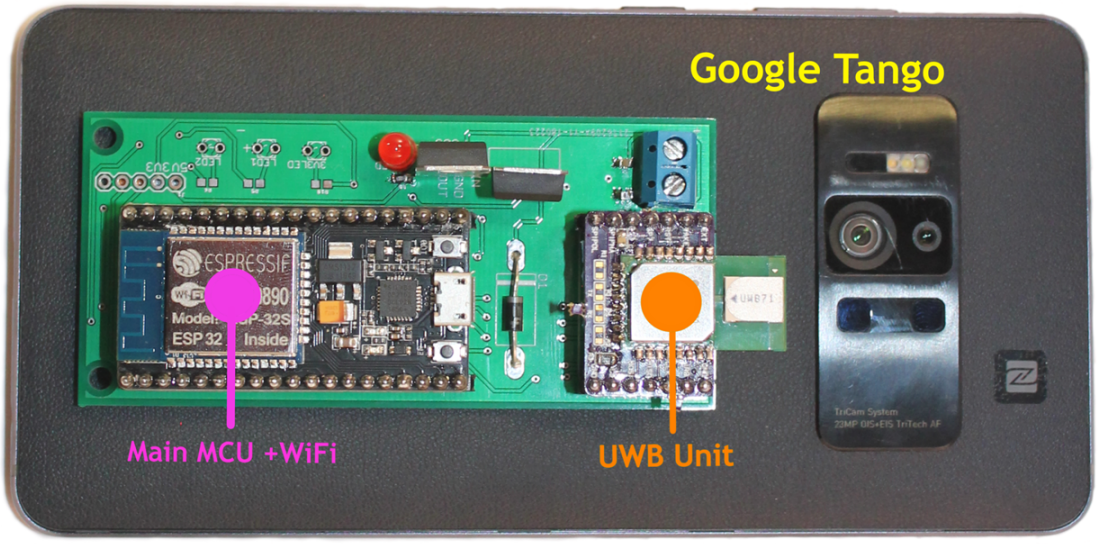

Figure 6.5. Hardware overview of the prototype. UWB based distance measurement module attached on a mobile AR device.

6.2.1 Hardware & Firmware

Our distance measurement module consists of a micro-controller unit (MCU), a UWB unit and peripheral circuits. The overall size of the board with all components assembled is $90mm \times 40mm \times 20mm$. We select a ESP32 (NodeMCU 32S) module as our MCU since it provides built-in WiFi communication function [53]. The UWB unit (DWM1000) connects with the MCU through SPI bus. We utilize a rechargeable Li-ion battery (9V, 600mAh) and a dual regulator set to power the MCU (5V) and UWB unit (3.3V) separately. As for the AR devices, we prototype our system with the ZenFone (ZS571KL) which runs a Google Tango system.

UWB units measures distances through a double-sided two-way ranging scheme operating on the MCU. This scheme corrects the time drift for the time-of-flight measurement by exchanging two round-trip messages [94]. When performing one to n ranging, we estimate the update rate is around $1000/(80+21n)$Hz with our current parameters, e.g., one-to-one ranging results in ~ 9.9Hz and one-to-two ranging results in ~ 8.1Hz. Correspondingly, in

a two-user or three-user registration, users are free to move with a normal speed ($\sim$ 1m). On the other hand, the SLAM from the AR device runs at a rate of $\sim$ 30Hz. Thus we keep synchronizing the newly received distance measurements with the most updated local positions as one complete sample, which yields an update rate $\sim$ 8.1Hz.

With continuous transceiving of UWB and WiFi, the whole board peak current reaches 300mA calculated based on the datasheet. A 600mAh battery lasts for $\sim$ 2hrs which means we can perform registration ($\sim$ 10s) about 720 times. After registration, we keep DWM1000 in sleep mode (550nA) so that the battery can last substantially.

6.2.2 Instant Registration

Recall Eq. (6.2), a sequential quadratic programming (SQP) algorithm is commonly used to effectively solve constrained optimization problems [128]. A number of software packages offer implementations for SQP. As in our prototype, we offload the solver onto a remote PC (CPU 2.5GHz, i7-6500U) which runs MATLAB Optimization Toolbox ([119]). We set the boundaries of t_y as $t_{y_{min}} = -0.1$m and $t_{y_{max}} = 0.1$m with the assumption that users initialize the SLAM within a height range of $[1.4 - 1.6]$m above the floor. For an one-to-one registration, we observe the algorithm converges in a short time (< 0.15s) with 100 samples. As a side note, we clarify that we do not focus on transplanting the SQP implementation onto mobile platforms here.

6.2.3 Collaborative AR Applications

Our applications need to manage three types of wireless communications: (i) the distance measurement modules and the AR devices, (ii) the AR devices and the remote solver and (iii) among different AR devices. We adopt the user datagram protocol (UDP) to transmit the measurements from the MCU to AR devices. As for synchronizing multiple users' positions, orientations, and collaborative activities, we set up a local server and utilize WebRTC [42] for real-time communications. Meanwhile, during the registration phase, we collect the local positions and distance measurements and feed them to the remote solver

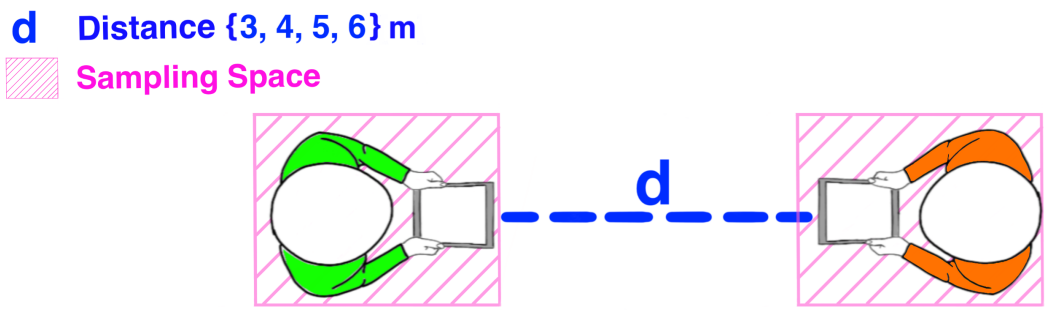

Figure 6.6. Technical evaluation setups.

through UDP as well. The AR collaboration applications have been implemented within Unity3D [175] using Google Tango API.

6.3 Technical Evaluation

To study the performance of our registration method, we set up a technical evaluation (Figure 6.6). Primarily, we considered a 2-user registration case. We studied the sampling parameters such as the sampling spaces and the distances between users. Since our approach requires users to roughly hold the device at a constant height during the sampling, we define the sampling space as an axis aligned bounding box ($l \times w$) on the horizontal plane $x-z$ plus a height level (h) along the y axis. And $r \in \{3,4,5,6\}$m denotes the distances between the sampling space centers of each user. We selected a sufficiently large 3D volume as the sampling space in order to capture the data systematically, i.e., $l \times w = 2 \times 2$m and $0.8 \leq h \leq 2.1$m. We collected 3000 samples for each r and repeated the same data capturing.

Our approach emphasizes on enabling spontaneous collaborations without sharing SLAM map. Thus we mainly compared with a registration given the shared map. For this purpose, the local positions of each AR device yielded the same coordinate system of the shared map. Then we synthetically created different frames by transforming the shared coordinate

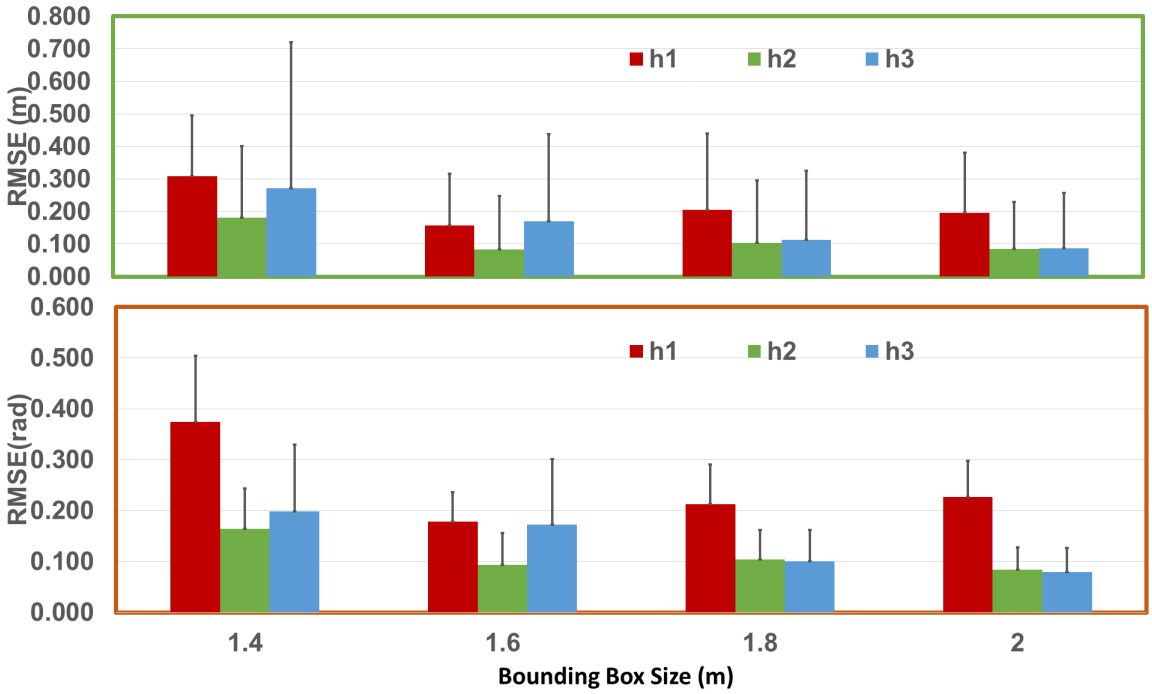

Figure 6.7. Results of evaluations of both translational (up) and rotational (down) accuracy on the sampling space with $l = w = 1.4, 1.6, 1.8, 2$ and three hight levels at $h_1 \in [0.9 - 1.5], h_2 \in [1.2 - 1.8]$, and$h_3 \in [1.5 - 2.1]$m.

system with randomly generated $\theta_g \in [-\pi, \pi]$ and $\mathbf{T}_g = [t_x, t_y, t_y]^T, -10 \leq t_x, t_z \leq 10$m and $-0.2 \leq t_y \leq 0.2$m. We intentionally varied t_y in a small range to simulate the real situation where different users would not be able to initialize the SLAM at the exact same height. We sub-sampled the datasets based on different test conditions and computed the synthetic local positions with the given ground truth transformations. Then we fed the optimization solver with the synthetic local positions and the true distance measurements. In the results, the accuracy of the registration was indicated by root mean square error (RMSE) of the translational (t_x, t_y, t_z) and rotational (θ) transformation separately.

6.3.1 Sampling Space

We evaluated the sampling space given the furthest distance between two users, i.e., 6m. Then we varied the planar bounding box of the sampling space as $l = w = 1.4, 1.6, 1.8, 2$m and dissected the heights into three levels $h \in [0.9 - 1.5], [1.2 - 1.8], [1.5 - 2.1]$. With these test conditions, we repeated the sub-sampling and optimization for 10 times and took the averages. Prior to the evaluation, our preliminary tests indicate a sampling number of 100 is a good balance between sampling time and the accuracy. Further 100 different ground truth transformations were drawn for each test.

A two-way univariate ANOVA result showed the bounding box size and the height level were significant to the accuracies of T and $theta$. Then we performed a post hoc pairwise comparisons with Bonferroni correction to examine the conditions separately. For both translational and rotational accuracy, we observed that, for $l = w \in \{1.6, 1.8, 2\}$, there were no significant differences ($p > 0.05$), yet $l = w = 1.4$ yielded a significant difference from others ($p < 0.05$). Further, pairwise tests with h still indicated significant differences from each other. As shown in the Figure 6.7, we confirmed that the average translational error stayed below 0.2m, and rotational one less than 0.21 ($\sim 12^\circ$) as the bounding box size became larger than 1.6m. The optimization result was sensitive to the distribution of the samples, e.g., a larger zone makes the optimization more robust. But when the region is sufficiently large, we suspected the optimization reaches to a limit because of the UWB accuracy.

Although h appeared to be significantly affecting the accuracy, the overall accuracy still remained low as long as $l = w \geq 1.6$m. Further from an ergonomic point of view, we selected a height level within $[1.2 - 1.8]$m. Note, our test adopted a strict condition on height variations (0.6m) to guarantee the effectiveness of our practical guidance.

6.3.2 Distances

Based on the results from the sampling space evaluation, we selected $l = w = 1.6$m and $h \in [1.2 - 1.8]$m for studying the effect of distance $r \in \{3, 4, 5, 6\}$m on the registration

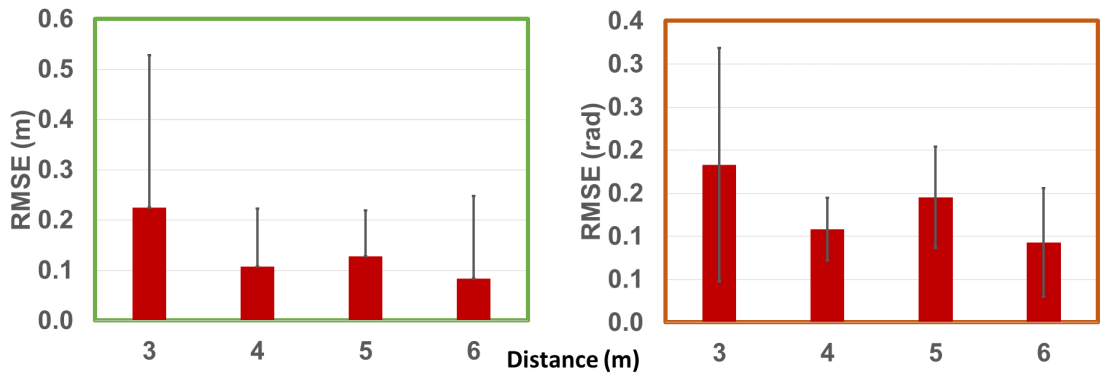

Figure 6.8. Results of evaluations of both translational (left) and rotational (right) accuracy on the distances ($r \in \{3,4,5,6\}$m) with $l = w = 1.6$m and $h \in [1.2 - 1.8]$m.

accuracy. With a one-way ANOVA test, we found that r significantly affects both both translational and rotational accuracies ($p < 0.05$). Pairwise comparisons with Bonferroni correction showed that within group of $r \in \{4,5,6\}$, there were no significant differences. We suspected that within a close range, the measurement accuracy of UWB unit may degrade. From Figure 6.8, we observed that, the average errors for $\mathbf{T}$ yielded below 0.25m for all r, and θ less than 0.23 (13.2^o).

6.3.3 Results

The investigations from the technical evaluation indicated we support one-to-one registration at various distances. With limited resources, we conservatively suggest the following sampling parameters for the registration: (i) initialize the SLAM device at a height of $\sim$ 1.5m from the floor, (ii) capture 100 synchronized local positions and distance measurements, (iii) during sampling, cover a space with $l = w \geq 1.6$m, (iv) hold the device at a constant height roughly ($h \in [1.2 - 1.8]$) for better accuracy. With these parameters, we observed an average translational accuracies of ~ 0.15m from Figure 6.7 and 6.8 and rotational one of ~ 0.13 (7.4°) when $r \geq 4$m.

6.4 Task Evaluation

To further verify the registration performance and examine the usability toward supporting spatial AR coordination activities, we conducted a task evaluation with users. We recruited 11 university students (10 male) with an average age of 25 to participate our study. The majority (9) of the participants were familiar with the concept of AR. We asked users to finish a two-session study which focused on view pointing and trace following with rendered AR cues respectively. Through these tasks, we emphasized comparing our distance based approach against the sharing map registration.

To setup a collaborative environment, one of the authors acted as *User A* and the participant played a role of *User B*. *User A* was provided with a pre-built SLAM map of the environment whereas *User B* always started the SLAM with arbitrary positions and orientations in the given environment. The visual cues were always created within the *User A*'s coordinate system at first . Then *User A* and *User B* held the device and kept moving on independent paths until enough samples were collected for the registration. With the runtime registration result, the visual cues were duplicated in User B's frame. Subsequently, with the AR cues, users were asked to finish the tasks. To remove possible learning effects, we offered a training and practice trial before the test.

We constrained the tasks to focus on evaluating the registration performance with the real users. Thus in this paper, we did not include any collaborative tasks and collect the subjective experiences. For the studies, we compared the performance against the central-map approach. Yet we did not let the user to explicitly experience the map sharing action (we set it up for users). For the *View Pointing* task, it took us about 15 minutes to scan the environment ($\sim 5 \times 7$m) and ~ 3 minutes to exchange the scanned map ($\sim$ 30MB) through a WLAN. As for the *Trace Following* task, we used a map ($\sim$ 50MB) for an environment of $\sim 10 \times 30$m. Further, we noticed the maps were sensitive to the ambient lighting condition.

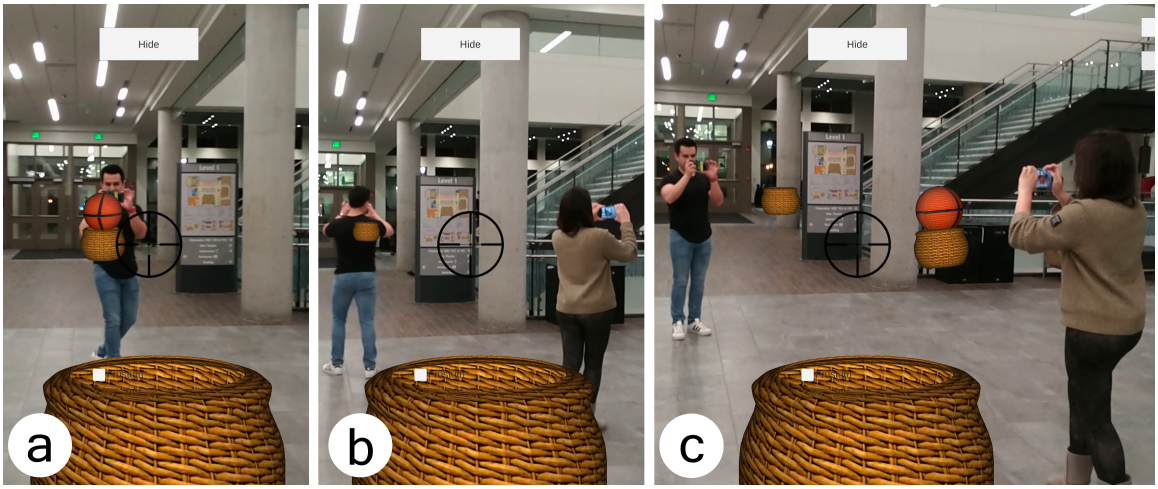

Figure 6.9. Setup for view pointing task evaluation. User sits on a rolling chair points to different directions with visual cues.

6.4.1 View Pointing

In a collaborative AR environment, it is essential to synchronize the orientations between users for spatial reference. As shown in Figure 6.9, we set up a top view camera in the physical environment so that the pointing results from *User A* and *User B* can be compared with a common reference. To be specific, *User A* positioned the virtual indicators while sitting in the rolling chair. After a registration, *User B* was asked to move towards the chair and sit in it. In each trial, we generated a randomized sequence containing 4 indices of the 8 evenly distributed virtual spheres. *User B* rotated the chair and pointed at a direction.

We asked the users to perform the registration followed by a trial 3 times in this task. In total, we obtained 132 images showing 11 users pointing at different directions. After processing the images with MATLAB, we recognized the triangle which is fixated on the chair and the corresponding direction in the image frame. Similarly, we captured the ground truth by averaging the pointing directions from 24 images of *UserA* pointing with the prebuilt SLAM map. Then we averaged the trials and compared with our ground truth. The overall mean error of 3.7° with a standard deviation of 9.0° is comparable with a sug-

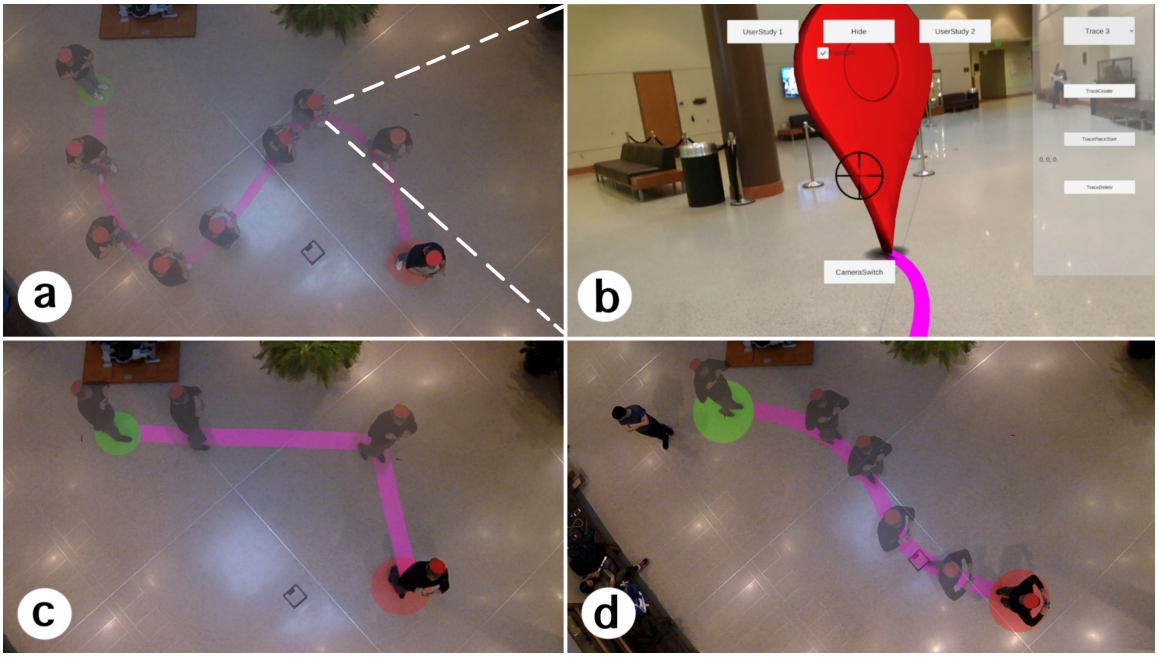

Figure 6.10. Illustration of path following task evaluation. Users follow 3 different virtual traces (a, c, d) in the AR scene (b).

gested viewfinder frustum field of view (8°) [8]. This result implies that *SynchronizAR* is applicable for orientation sensitive AR collaborations.

6.4.2 Trace Following

We selected a trace following task to evaluate the effects of both translational and rotational results on the AR guidance scenarios. Unlike the fixated rolling chair in task 1, users dynamically moved in a larger space ($\sim 5 \times 3$m). We generated a metric to evaluate the similarities between different paths from the recorded top-view videos. To eliminate the subjective motion from different users, we created baselines for each user. To be specific, instead of creating a ground truth from *User A* in prior, we requested users to follow the traces with the registration provided by a shared map twice. Then the ones with runtime registrations will be compared with this baseline.

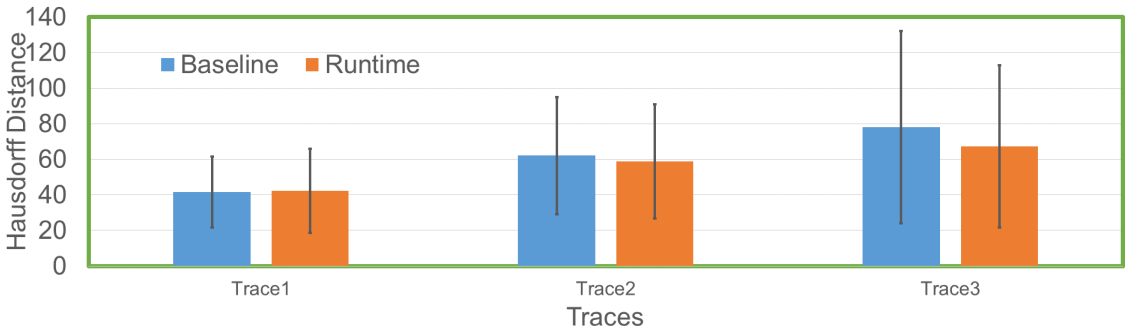

Figure 6.11. Results from trace following task.

As shown in Figure 6.10, we constructed 3 traces with different shapes (L-, S- shape, and a spline) with the same starting and ending points to represent curves with different curvatures. Each user was asked to follow all three traces 4 times in total, i.e., twice with ground truth and twice with runtime registrations. The camera captured the trace following movements where users wore a hat which was covered by a red dot. After processing the video, we obtained the paths of users in the image frames. A modified Hausdorff distance (pixels) increases monotonically as the amount of differences between two sets of points increases [52]. It is often used to compare the similarities of two curves. Thus we employed the Hausdorff distance as it is sensitive to both translational and rotational errors between the curves. For each user, we denoted the two sets of paths with ground truth as G_1 and G_2, and the ones with runtime registration as H_1 and H_2. Further, for each user, we calculated the Hausdorff distances between paths in G_1 and G_2 ($D_{G_1G_2}$) with respect to different traces. We composed $D_{G_1H_1}$, $D_{G_2H_1}$, $D_{G_1H_2}$, and $D_{G_2H_2}$ together and performed a T-test against $D_{G_1G_2}$ from all of the users.

For all three traces, we observed no significant difference between the baselines and the runtime registration results ($p = 0.92, 0.77, 0.55$ respectively). The mean errors and standard deviations are plotted in Figure 6.11. Through this task evaluation, we confirmed that our registration accuracy supports creating visual guidance in AR collaborations.

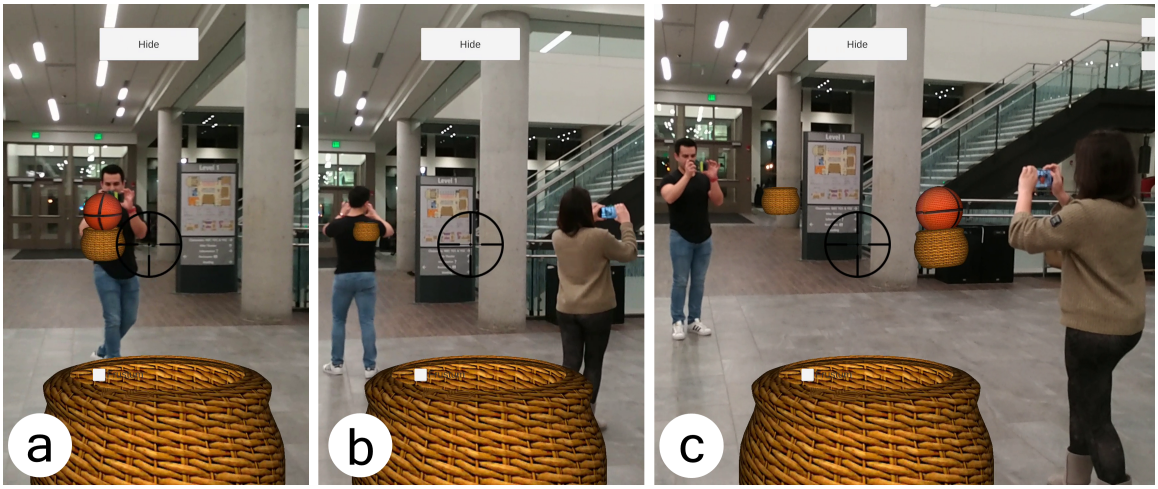

Figure 6.12. *SynchronizAR* supports spontaneous collaboration, i.e., a new user (b) join an existing AR collaboration (a) instantly (c).

6.5 Example Use Cases

By applying the registration result, *SynchronizAR* enables every AR device to be spatially registered with each other instantly and conveniently. Taking advantages of the spatial awareness across the users in an AR environment, we showcased four use cases with *SynchronizAR*.

6.5.1 Spontaneous Collaboration

Here we built a multiple-player ball catching game with support from *SynchronizAR*. We leveraged the spatial interactions such as pointing enabled by the registrations in AR collaborative games. Further we demonstrated our instant registration technique which enables a player to join any time during the game. At first two players started a game (Figure 6.12 a). Then a third player was able to join the game after a quick registration process with one of the original players (Figure 6.12 b). After that, the coordinate system of the new player was shared between the original collaboration environment and the game continued with three players (Figure 6.12 c).

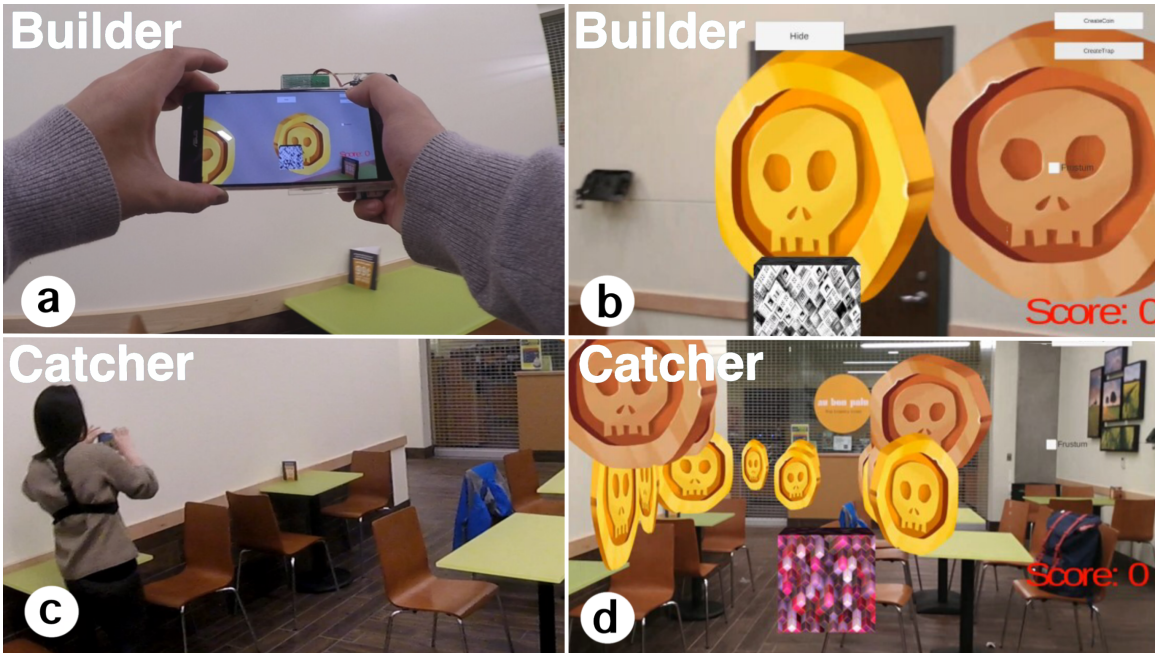

Figure 6.13. Interactive AR game creation. Two users act as a game world builder (a, b) and a player (c, d).

6.5.2 Interactive AR Game Construction

With *SynchronizAR*, we created an interactive AR game construction and playing experience to multiple users. Here we allow users to construct AR games in the physical world as a game map and instantly share it with other users once registered. For example in this coin-collection game, a builder (Figure 6.13 a, b) first placed golden coins and rusted coins in the café and turned it into a game scene. Then a catcher (Figure 6.13 c, d) registered with the builder and synchronized with the game world. With proximity based spatial movement, the catcher collected coins in the AR scene. We also support asynchronized collaboration as we need no infrastructure prior. After registering once, any user can revisit the scene and view collaborator's activities which happened while he/she was gone.

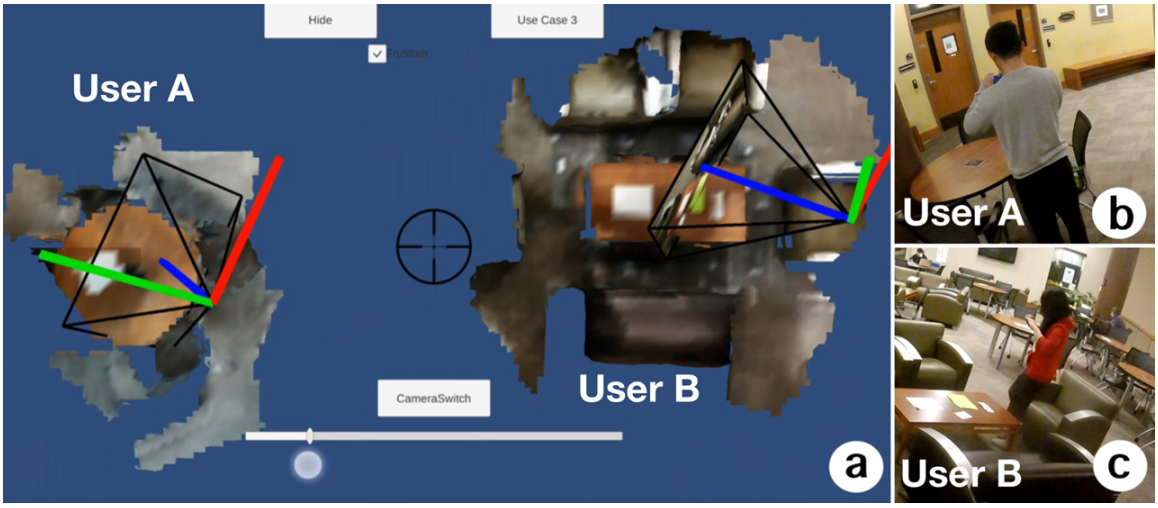

Figure 6.14. A spatially coherent virtual model (a) is created after user A and B scan their own surrounding environment (c, d). Two distant users can refer to each other's view with spatial references (a, b, e).

6.5.3 Spatial Aware Screen Sharing

In a co-located collaborative context, two users stays distant from each other may also want communicate through view sharing instantly. Different from a traditional video conferencing, *SynchronizAR* offered spatial awareness to the shared view. Also during the collaboration, we allow users to freely refer to each other's surrounding environment. Here, the users scanned the environment around each of them separately (Figure 6.14 c, d). Then the scanned geometry models can be registered using the spatial transformation from *SynchronzAR*. As the user walked around, the distant collaborator can access the first-person view through the frustum, also create an independent virtual navigation with the registered 3D model.

6.5.4 Human Robot Interactions

In the future, we envision that human beings and autonomous robots interact with each other naturally [163]. In this context, the spatial awareness will be critical. As Figure 6.15,

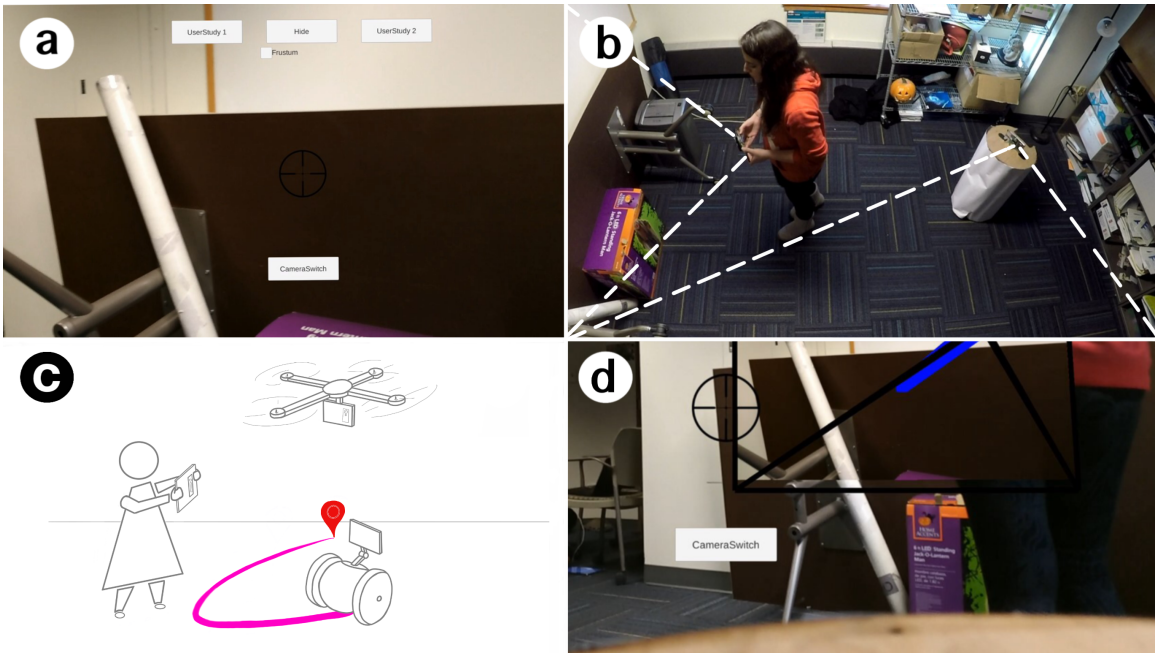

Figure 6.15. *SynchronizAR* being used for human-robot interactions(c). The robot mimics the user's movement (b). And they can access each other's views (a, d).

by attaching an AR device to an autonomous robot and registering it with a user, we co-ordinate the robot with respect to the user's position and orientation. Thus, the user can interact with the robot naturally through his/her spatial movement. For example, in this use case, we enable the robot to mimic the user's movement in the same direction and adjust the facing direction accordingly.

6.6 Discussion and Limitation

Sampling Parameters. With limited resources, we were not able to fully investigate the sampling parameters. In our current setup, we primarily rely on a shared SLAM map as ground truth for testing. Despite the stable performance on Google Tango devices, we observed drift from time to time in a feature less environment. In the future, we also plan to introduce an external tracking system e.g., a VICON like system to study the effects

of possible drifts from the SLAM itself. Additionally, our distance based registration required users to move on independent paths. Although during the user study, we haven't observed identical walking patterns, it will be helpful to give AR walking cues to users during registration.

Temporal Synchronization. We run a 1-to-n pooling where the n distances were packaged on an arbitrary MCU and sent to the AR devices via UDP. The newly received distances package, together with last updated coordinates which were smoothed by a running average, were sent to the solver. Although we did not explicitly model the temporal differences between the measurements and the coordinates, the running average practically reduced the potential correspondence error. We acknowledge that the accuracy may improve with a dedicated synchronization scheme. Still we found the average positional RMSE (~ 0.25m) remains at the same level of the UWB accuracy (~ 0.1m).

Scalability. We believe the modern mobile device can solve our optimization problem given the fact it runs SLAM in real-time which usually involves heavy optimization. In a non-central deployment, the distance measurements and the local coordinates can be first synchronized and packaged on the local AR devices. Then the packed messages will be shared through a peer-to-peer communication. Finally, the optimization runs in the AR device instead of a remote server.

Potential Applications. Although the cloud based solution is capable of supporting the collaborative AR given a reliable map, our method is more suitable for cases where a reliable map is not available or hard to access (a dynamic environment), or not necessary (e.g., casual social AR activities). Also, for large spaces (e.g., urban planning), the users can start the collaboration at different locations instantly without scanning the map as shown in the *Spatial Aware Screen Sharing* case. Further our method can be used to augment other approaches. For examples, enhancing LBS with accurate registrations (e.g., Pokemon battles), and with cloudAR, enabling asynchronous and persistent experience.

Form Factors. For the AR devices, we selected Google Tango phones to prototype our AR applications. However, our registration is applicable to heterogeneous devices (HMD and handheld) running various SLAM algorithms since our indirect approach does not

require sharing SLAM map. Further, our registration can be utilized for establishing a co-located collaboration for virtual reality (VR) devices which rely on SLAM tracking. On the distance measurement side, we would like to work on minimizing the package of the module. Besides, it will interesting to generalize distance based registration approach with matured RF technologies (Bluetooth and WiFi) with different types of distance estimation (time-of-flight, time-difference-of-arrival, and angle-of-arrival) [144].

Accuracy. Although we observed a good translational and rotational accuracy within a large area, we found the UWB measurements can be distorted under heavy non-line-of-sight (NLOS) conditions such as solid walls. In the future, we need to identify the NLOS measurements and compensate or remove them. Besides, the SLAM algorithm itself may drift in a featureless environment causing inaccurate registration or shifting the AR rendering after the registration. Also we observed the standard deviation of the error remains high as shown in Figure 6.7 and 6.8. We suspect this is caused by the SLAM drift primarily. Future, we plan to determine the error resources by comparing with a VICON system.

Number of Users. Our current supports for more than 3 users rely on pairwise peer-to-peer registration. To further support more users being registered simultaneously, we need to overcome two issues: (i) sampling rate of distance measuring, and (ii) introducing distance constraints into the optimization. We plan to resolve the sampling rate limitation by introducing time-different-of-arrival. As for the highly nonlinear constrained optimization, we still need to investigate and select a method which is applicable for mobile devices [196].

6.7 Conclusions

In this work, we proposed *SyncrhonizAR*, enabling a co-located collaborative AR experience by spatially registering multiple users in a spontaneous manner. Through our technical evaluation, we conservatively suggested guidelines for using *SynchronizAR*. We observed an average translational accuracy of 0.15m and rotational accuracy of 7.4° when two users are at a distance $r > 4$m. Within the user study, we validated that with our reg-

istration, users can successfully perform AR spatial interactions accurately including view pointing and trace following. Therefore, we believe our work is applicable to a wide range of use cases leveraging the spatial registration of multiple SLAM devices.

To this end, we unlock and explore the spatial intelligence for co-located AR collaborations. Since our approach does not rely on prior knowledge or external infrastructure, we emphasize on enabling spontaneous collaborations in AR. We believe such an instant and easy-to-deploy registration method will further contribute to the pervasiveness of AR in the context of collaboration.

7. ADDITIONAL APPLICATIONS

7.1 Overview

Through Chapter 3 to Chapter 6, our studies mainly focus on augmenting human's interactions with physical environments through AR. The environment can be a local desktop setup (Chapter 3), a large space where users can freely walk around (Chapter 4), an smart environment distributed with connected IoT devices (Chapter 5), and a co-located space where multiple users can collaborate (Chapter 6).

In the vision of ubiquitous and pervasive computing, the Internet of Things (IoT) technologies are rapidly emerging and the embedded electronics are getting smaller, lower in cost, proliferating and being embedded in our everyday environment. Inevitably, to support a pervasive AR, we study human-IoT interactions as in Chapter 5. 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 [3]. 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 [145], and an authoring system for such robot-IoT interactive task planning is underdeveloped [178]. 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.

The robots' intelligence remain underdeveloped for a majority of the ac-hoc tasks in less controlled environments including our daily household environment [91]. We here propose to tackle the problem from both an autonomous robot direction and an interactive human authoring perspective. Based on our previous study results, we discuss two more

applications leveraging robots' and humans' spatial awareness of the environment. In the context of IoRT, we develop systems to: (i) enable spatial intelligence for autonomous robots by leveraging our distance based localization method, and (ii) support in-situ spatial aware authoring of spatially distributed robot-IoT tasks.

For the first system, we re-purpose the IoT device localization method from Chapter 5 to a robotic exploration and mapping use case. The IoTs here serve as as spatial landmarks which help the robots navigate and discover the surrounding environment. The IoTs could also include task related information such as the manipulation details for the robots.

In the second application, 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. 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.

7.2 Spatial Intelligence for Autonomous Robot

Within our surrounding environment, the ad-hoc tasks which we take for granted are often complex for robots because of their limited perception capabilities and underdeveloped intelligence algorithms [99]. Despite the beginnings of commercial successes of mobile robots, particularly in warehouses, they are mostly specialized in handling simplified and pre-defined tasks within controlled environments often with fixed navigation pathways. Further, many of the AI advances in navigation are in simple settings with many assumptions, and are not useful in realistic workflows and environments [156]. On the other hand, the rapidly emerging IoT ecologies bridge our physical world with digital intelligence. In contrast to ongoing advances in vision, we propose an integration of robots into the connected network, where they can leverage information collected from the IoT, and thus gain stronger situational awareness [164] and spatial intelligence, which is especially useful in exploration, planning, mapping and interacting with the environment without relying on AI/vision-based navigation only.

Recent advanced computer vision technologies, such as SLAM algorithms and depth sensing, have empowered mobile robots with the ability to self-localize and build maps within indoor environments using on-board sensors only [32], [126]. Although these systems provide good maps under favorable conditions, they are very sensitive to calibration and imaging conditions, and are not suitable in changing dynamic environments. Therefore, to fully support navigation and interaction with the environment, we need to extend robots' perception from a vision-based geometric level to a semantic level. Although researchers have made substantial progress in scene understanding, object detection and pose estimation [154], vision-based approaches largely rely on knowing the object representations a priori [192] and keeping the objects of interest in the camera's view. That said, vision-only approaches may be more suitable for local and specific tasks. Thus, mapping key parts of the environment and identifying the objects of interest, and especially finding means to interact with them using vision-based methods, usually do not have well-developed solutions.

In contrast, within a smart environment, wireless techniques such as Bluetooth, Zigbee, and WiFi allow for instant discovery of the connected objects via the network. Further, the robots could naturally access the semantic information stored in the local IoT devices which contributes towards understanding the environment and results in intelligent user interactions. Still, resolving the spatial distribution of the IoT-tagged devices or objects remains challenging. Using the wireless communication opportunistically, received signal strength indicator (RSSI)-based methods for localization of the sensor node have been studied extensively in the wireless sensor network (WSN) field [80]. Yet, the low accuracy of the results (a few meters) may prevent them from being employed for indoor mobile robots. Other researchers have developed UHF RFID-based object finding systems [47]. However, these systems introduce an extra bulky and expensive UHF antenna, suffer from a limited detection range ($\sim$3m), and, using their approach, a robot must perform a global search before navigating to and interact with the IoT tags.

Recently, researchers have been investigating distance-based localization methods using an ultra-wide bandwidth (UWB) wireless technique which provides accurate time-of-

flight distance measurements [49]. Such techniques have been further applied to enable users to interact with the smart environments in our previous work in Chapter 5 [86]. Inspired by these works, we propose a spatial mapping for IoT devices by integrating UWB with SLAM-capable robots. The SLAM-capable robots simply survey in a small local region and collect distance measurements to the UWB-IoT devices for a short time, and then our mapping method outputs the global locations of the devices relative to the SLAM map. Our method supports navigation and planning in previously unseen environments. We leverage the discovered IoTs as spatial landmarks which essentially work as beacons that help the robot familiarize itself with a complex environment quickly without accessing any pre-stored and static databases. Centering upon this mapping method, our contributions are three-fold as follows.

- A method to automatically explore and map a smart environment where UWB-IoT devices are distributed.
- A navigation pipeline that drives a robot to a target globally and then refines the object localization, for example with object handling and manipulations.
- Demonstration of our method with a prototype service robot (i) working with users through a task-oriented and spatially-aware user interface and (ii) exploring an unknown environment referring to IoT landmarks.

7.2.1 Workflow

We develop an IoT module consisting of a WiFi and a UWB communication component as shown in Figure 7.1. A commonplace use case scenario involves a set of IoT devices spanning an indoor environment and a SLAM-capable robot with an IoT module attached. The robot connects to the IoT through a WiFi network and the UWB network then primarily provides distance measurement capabilities. When entering an unknown environment, the robot surveys in a local small region ($1.5m \times 1.5m$) and collects the distance measurements to the IoT devices. A distance-based method is then used to estimate the multiple IoT

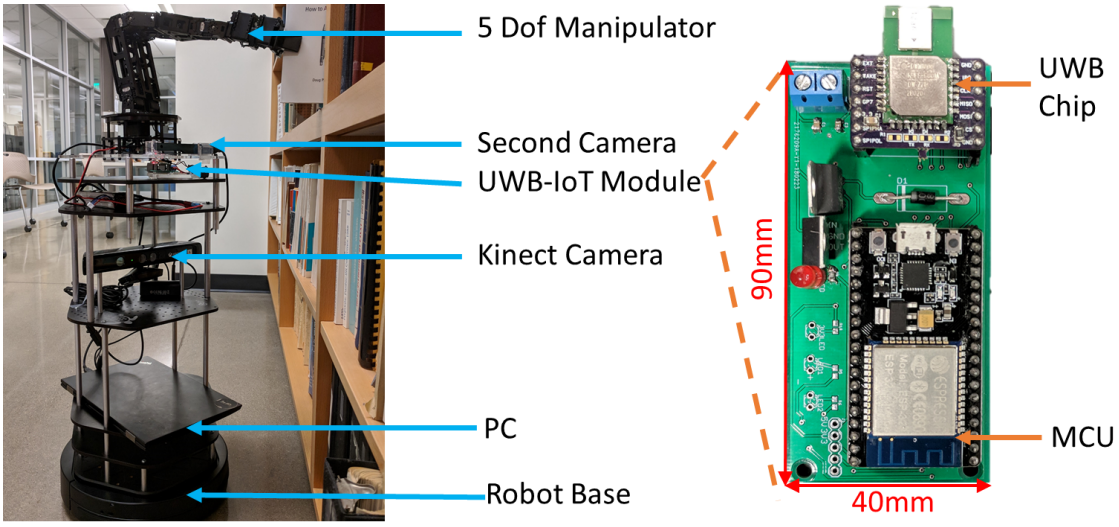

Figure 7.1. The robot platform and UWB-IoT module.

locations simultaneously and register them within the SLAM map, namely, mapping the smart environment. Depending on this semantic map, the robot navigates close to the targets and finishes tasks locally.

To complete a manipulation task, our robot needs a navigation strategy through three phases (Figure 7.2): (i) surveying movements to collect enough distance samples in a local region, (ii) globally approaching into the proximity of the IoT object, and (iii) locally adjusting poses for executing the manipulation.

For the first phase, we design a static random walk trajectory to guarantee the non-colinearity of the sample positions during the surveying. Further, based on our preliminary experiments and results from the previous work [86], we keep the footprint of the trajectory sufficiently large ($1.5m \times 1.5m$) to achieve accurate localization in a large room ($\sim 10m \times 10m$).

In the second phase, we employ a path planner which integrates a global costmap and a local costmap. Since we emphasize the exploration and navigation in an unknown environment, as the robot marches and the map updates, the planner re-plans the trajectory. The planner utilizes the local costmap to avoid dynamic obstacles during the exploration.

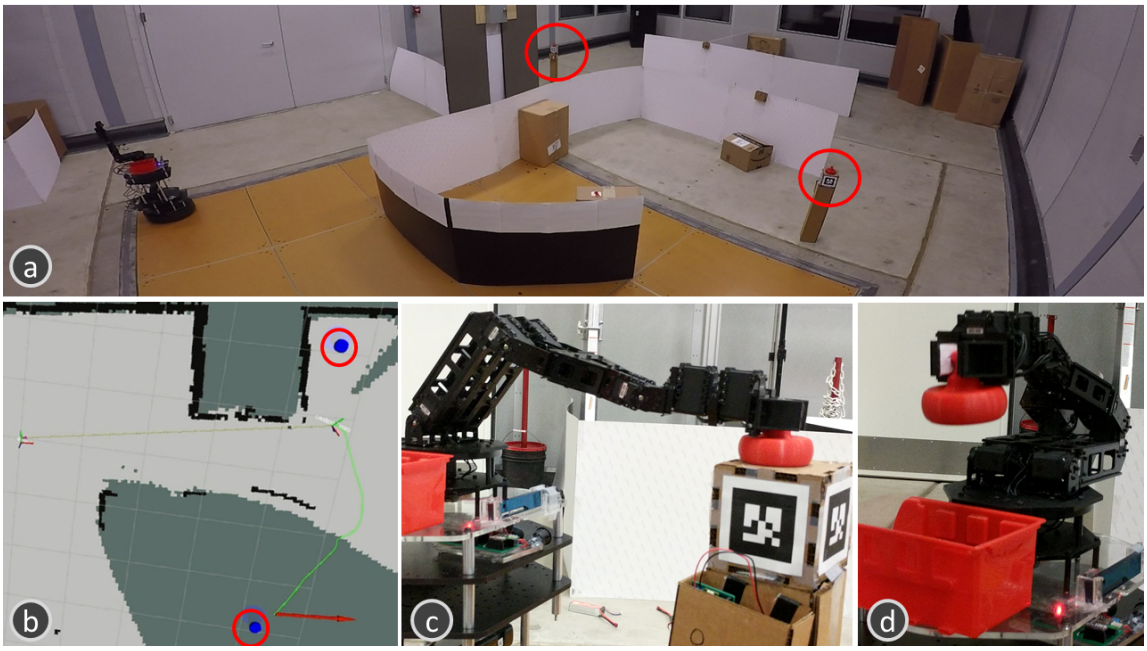

Figure 7.2. Setup for navigation and manipulation test: our robot visited two IoT targets (a, b) according to the localization results, then grabbed the target (c) and placed it to the basket (d).

Although the UWB-based localization is accurate enough to drive the robot close to the targets, the manipulation task usually requires millimeter-level accuracy. Thus, for the third phase, we employ vision-based tracking for the granular pose adjustment.

As the scope of this paper is on phase one and two, we simply use fiducial markers to perform the local manipulation. To handle the transition between phases two and three, we use the distance measurement as a threshold for proximity detection (e.g., less than 1 meter). Moreover, the IoT devices facilitate the manipulation procedure by providing semantic information, such as the offset from the marker and grasping directions.

7.2.2 Use Cases

Our workflow emphasizes autonomous mapping and interacting with the smart environment. We envision that the robot will be empowered with spatial awareness of the

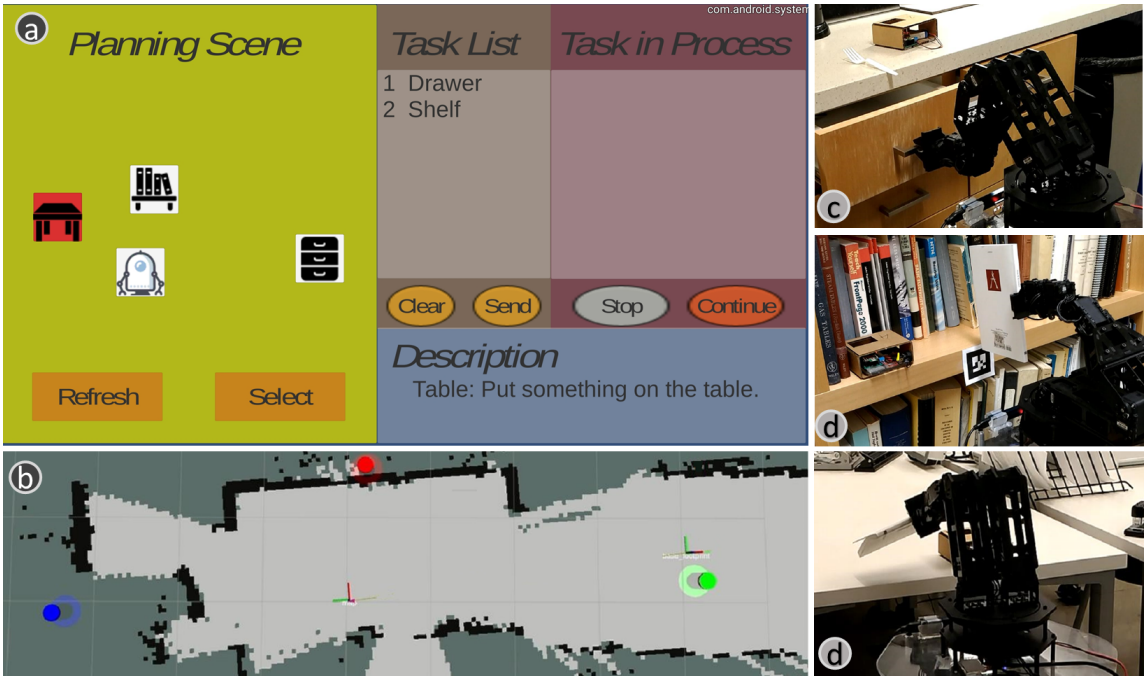

Figure 7.3. Through a spatial-aware programming interface (a), a user schedules a robot to perform a sequence of tasks: cleaning the kitchen table (c), delivering a book from a bookshelf to a desk (d, e).

distributed IoT devices. Here, we selectively demonstrate two use cases leveraging the enhanced spatial intelligence of the robot.

Task-Oriented and Spatial-Aware Programming

Our approach in general contributes to a higher level autonomy for robots to interact with a smart environment, e.g., general purpose service robots interacting with a smart home. As shown in Fig. 7.3, to command such a robot to conduct a sequence of tasks, a user simply uses a mobile user interface to schedule the IoT-indicated tasks. Then, the robot is capable of localizing the targets and accessing the back-end knowledge from the IoT network. The real-time spatial relationship between the robot and the IoT targets is updated to the users for better task planning.

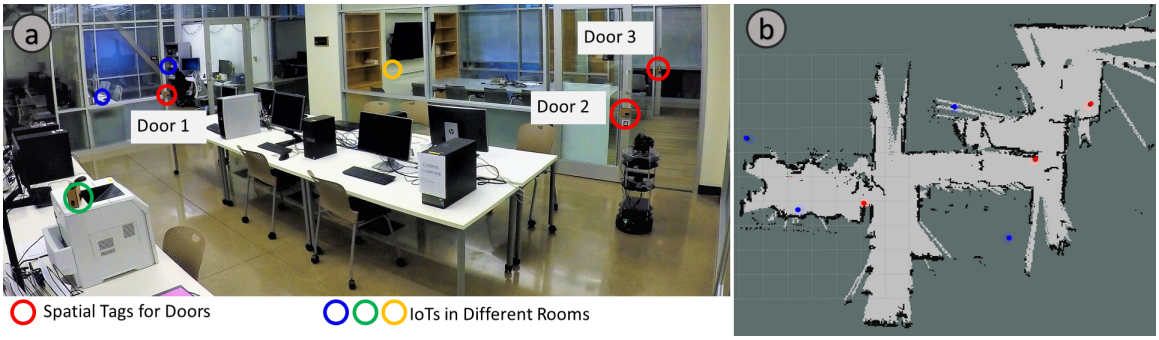

Figure 7.4. A robot explores an environment which includes multiple rooms by referring to spatial tags on the doors.

Autonomous Exploration Using Spatial Tags

Although UWB-based localization suffers less in non-line-of-sight (NLOS) scenarios compared to approaches using computer vision, a heavy NLOS condition such as walls still degrades the accuracy. To mitigate this issue, we propose to use UWB-IoT as spatial landmarks and references for the robot to navigate and explore multiple rooms in a continuous manner. As illustrated in Fig. 7.4, we showcase a robot navigating through three IoT-tagged doors and exploring three rooms. Each tag on the doors provides spatial knowledge about a local region. Finally, we localize all IoT devices in the rooms and register them onto a single SLAM map. With our autonomous exploration, we foresee greatly lowering the barriers to deploy the robots in realistic environments.

7.3 Spatially Aware Human-Robot-IoT Interface

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 [1], users can also coordinate multiple devices working with other productivity tools or social medias via active human-IoT communication [3, 40]. Even in these coordinated works, the IoT tasks are rather spatially independent. In these cases, conventional graphical user inter-

faces (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 [2] 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 [91]. 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 [23].

In the context of robots-IoT ecology [145], 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 [79]. In a similar manner, robots have been attached with tags and tracked through the users' AR camera view [33, 97, 116]. 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 alternatives including external tracking systems (e.g., infrastructured cameras

[56, 74, 199, 93]) and pre-scanned and manually tagged environment maps [127, 110, 41] 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.

7.3.1 Workflow

As illustrated in Figure 7.5, 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. Be-

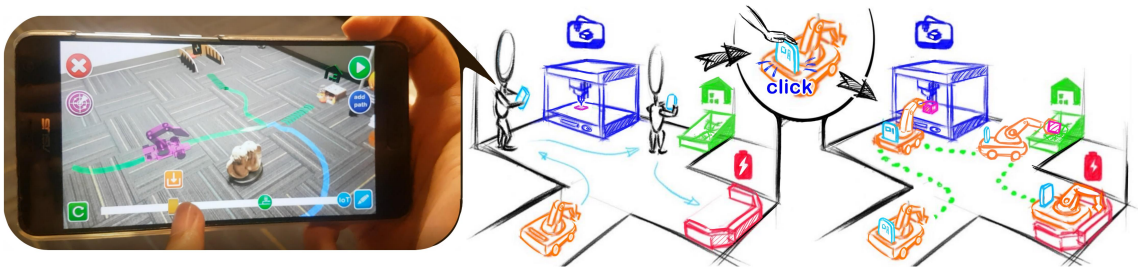

Figure 7.5. V.Ra system workflow. Using an AR-SLAM mobile device, the user first spatially plan the task in the AR interface, then place the device onto the mobile robot for execution. The room-level navigation of the robot is guided by the SLAM feature on mobile device.

cause of the transparency between the users' intents and robots' actions in the AR authoring phase, we achieve programming a robot in a WYDWRD fashion.

7.3.2 Use Cases

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 7.6 (1)), then pinpoints the area for cleaning using the *SpotSweeping* robot function (Figure 7.6 (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 ro-

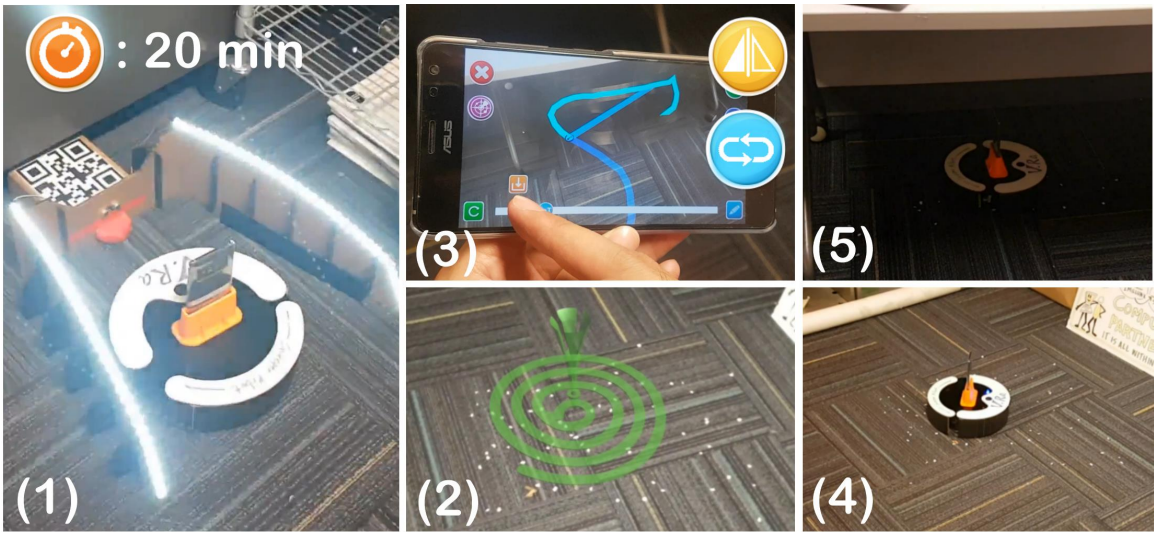

Figure 7.6. Use case 1. (1) Battery charging for 20 minute. (2) Using the spotSweeping feature to author floor cleaning. (3) Using the Mirror and Loop feature to author repeated sweeping path under the table. (4) SweeperBot cleaning the floor. (5) Robust navigation under the table with poor lighting condition.

bustness of the system's navigation capability, that the robot is able to successfully cruise under the table with poor lighting conditions (Figure 7.6 (5)).

8. DISCUSSIONS AND CONCLUSIONS

8.1 Discussions

This thesis explored various spatial interactions in different AR use case scenarios. After the interaction exploration, technique development, and application expansion, I would like to take a retrospective approach and examine the spatial awareness in AR again. First, I recall the research questions guiding this thesis through. (i) What are the interaction metaphors suitable for the evolving AR, and the corresponding enabling techniques? (ii) How can we create seamless AR experience across different use scenarios? In Chapter 1, I justify our studies on spatially aware interactions based on the definition of AR: the object in the physical world being augmented by computer generated contents. To register the physical world and digital contents, by nature, an AR scene requires spatial awareness of the physical world. I discuss more at a deeper level, for example, human psychology, intelligence, and cognition level. At this manner, I can better justify the significance of spatial awareness in AR. More importantly, as AR has been emerging and advocated as the next generation interface beyond the mobile phones, I hope to share some ideas on identifying the critical gaps and challenges where the power of AR really matters and can be unleashed, supporting the prospects of AR. Thus, our discussions may inspire a wide range of future research directions.

From Howard Gardner's Theory of Multiple Intelligence, as one of the nine kinds, visual-spatial intelligence has been defined as *an ability to form a representation of the world* [58]. The spatial intelligence has been characterized as an important individual attribute which is particularly relevant to learning scientific-technical materials [183], i.e., science, technology, engineering, and mathematics. Thus, educators have proposed methods to measure the spatial intelligence and to develop it through instruction and training [76]. As the computer graphics technologies develops, external visualization has been

introduced to assist mental imaging of the representation and the spatial relationships. Such applications can be designed to be interactive to provide highly engaging user experiences. Then the questions can be re-framed as: what does essentially AR provides beyond the normal computer graphics applications? Here I provide some perspectives centering around leveraging human spatial intelligence in AR as follows.

Situatedness. As for a specific task, I refer the *situatedness* as the degree the digital information and person are connected to the task, the location, and/or another person [172]. Modern advancing computer graphics technologies enable externalizing designer's imagination into 3D. Further, VR allows users to experience the virtual contents in 3D with high immersion. By blending virtual creations into pictures, movies, or even 3D scanned models in a realistic manner, users experience "being-in" the situation. But for AR tasks, I argue the following unique features which stands out from other approaches of "in-situ".

- AR requires users to physically present in the scene which provides multi-sensory perception of the environment, e.g., haptics, scent, and audio. The multi-sensory stimulus, which are hard to be reproduced in pure virtual environments, often promote ad-hoc creativity. In the future, once the AR technology stack is mature, supporting users spontaneously and instantly access the digital augmentations in the physical world could be an essential feature.

- AR allows users to interact with both the physical and the virtual objects. On the other hand, to enable such highly interactive experiences, the AR system needs to have a real-time responsiveness. The system needs to be responsive to the user's interactions against virtual contents as well as the physical environments. For examples, if the users changed the physical environment, the associated virtual contents need to be adjusted. Also, if the physical environment is changing according to some protocols, the virtual contents need to be responsive to both user's interactions and the environmental changes.

Mobility. For virtual contents presented in 2D monitors, screens, and VR devices, users are mostly experiencing them in a constrained setup. The movements of the perspectives

are usually mapped to conventional keyboard or controller inputs. From this view, I argue that mobile AR provides mobility in a large scene by allowing users freely to move around and change perspectives.

- Users' innate spatial abilities are tightly connected with their kinesthetic abilities. Sometimes, the physical and embodied movements better reflect the mental and abstract representations. In a lot of spatial tasks in AR, for example, layout reconfiguration for an existing environment, the capability of being able to move around freely is crucial. The movement of the users can be directly used as natural interactions such as changing perspective freely and examining the contents as in Chapter 4, and proximity based interface adjustment as in Chapter 5.

- Interactive AR applications can go beyond simple visualizations. If this is the case, the movements of users may serve part of the spatial tasks themselves. For example, in Chapter 6, I explored constructing AR games where users move around, place game contents spatially in the environment, and accomplish the game by walking. Further, users' reactions to the digital augmentations are often bound to their movements. Instead of digitizing users' behaviours which sometimes can be inaccurate and time consuming, mobile AR introduces users' true actions/reactions into the scene directly.

Collaboration. Collaborations can happen in multiple scenarios including co-located and remote, synchronous and asynchronous, human-to-human and human-to-robots, and so on. Therefore, to deliver efficient communications in a collaborative task, users usually demand a common context to refer for all the scenarios. I foresee AR has the potential to infer the common context for collaborations.

- For collaborations which involves spatial tasks, reasoning the communication in a common spatial context is crucial. For example, when communicating through voice, simple navigation guides such as "on your left" will require spatial reasoning. Since interactive mobile AR naturally augments the physical world by spatially inferring

the environment, I can leverage the scene understanding results to construct the common context.

- Compared with human to human collaborations, human-robot collaborations require inferring and reasoning about the human's interactions. AR devices usually provide egocentric view of users' motions, users' locations in the environment, voice, and even gaze, which can all be leveraged for human-robot communication purposes. More importantly, the intents of the robots' actions can also be easily expressed through AR. Such active feedback from the robots could contribute to a safe and effective human-robot teamwork.

8.2 Future Works

Based on the takeaways from the discussions, I recommend the following future directions to continue exploring and investigating spatially-aware interactions in AR, resolving the technical challenges and developing applications across different domains.

8.2.1 Human-in-the-loop Simulation Through AR

I envision a physical reality simulation platform where real humans wear AR devices and operate in an augmented environment. Through AR, I can test the new factory designs before actually building it with real humans. The new factory can be simulated in a physical environment by introducing mock-ups of the factory environment and machines. The augmentation through AR will ensure the situatedness of the users. Moreover, in the future factory, robots will play an important role. In the simulation engine, the virtual robots interact with the inferred physical environment. Further, through AR, users can interact and collaborate with the robots. To this end, I introduce a human-in-the-loop simulation to test the human-robot-machine interactions. I also identify two critical AR technology challenges for building such an engine as follows.

- In order to simulate the robots and machines in the engine, I need to infer the physical environment and embed them into the simulation. With the combination of computer vision and IoT technologies, I reconstruct the scene in a geometric level, associate the physical attributes with detected objects, author digital information and virtual representations bound to the objects.

- Through AR, I need to capture the humans' interactions with the physical environment and the augmentations and embed them into the simulation. For example, the virtual world in the engine needs to be updated according to the humans interactions. This is especially critical for simulating robots/machines with human-awareness.

8.2.2 Sharing Context for Heterogeneous Agents

In order to establish an effective collaboration environment between multiple agents including humans with AR and autonomous/semi-autonomous robots, I argue that sharing contexts across these heterogeneous agents is the key. Heterogeneous agents are usually equipped with various perception hardware and tracking/localizing software. Thus, compared to inferring context in the human-in-the-loop simulation, sharing contexts between multiple heterogeneous agents could be more challenging. Here, I mainly discuss from the perspectives of spatially registering, collaboratively infer the contexts, and data exchange across multiple agents.

- Resolving the transformations across multiple agents with heterogeneous perception capabilities can be difficult. For example, the common AR devices rely on visual-inertial based SLAM, while ground robots on the other hand may use cheaper Lidar sensors. Moreover, even using the same tracking method, e.g., rgb camera based SLAM, different agents may have very different perspectives, e.g., drone could be flying higher, ground robots' vision system is mainly confined at a lower height. I suggest two possible methods here: (i) a co-SLAM approach where multiple agents start from a common anchor scene and then collaboratively create the map, and (ii)

an instant registration leveraging extra measuring devices such as UWB distance measurement modules from Chapter 6.

- Instead of a single source of input, the sharing contexts for multiple agents can be inferred with all the available sources. Again, if I assume the agents are equipped with heterogeneous perception capabilities, resolving the conflicts across the multiple sources could be challenging.

- The context could include a large amount of data, such as the geometry data, semantic data from the IoT environment, states of the agents, and interactions from the humans. In order to infer the contexts collaboratively in real-time, creating a sophisticated data-exchange is necessary. For example, issues such as how to balance the data throughput and the local computation loads, and how to schedule different types of data according to their priorities, become critical.

8.2.3 Transparent Knowledge Transferring Through AR

One of the major application domains for AR is instructing novice to complete tasks. An important reason is that users are provided with superimposed digital guidance which dynamically progresses as users perform the tasks. But not many solutions are available for creating the AR guidance effectively and efficiently. I see this as a future opportunity. To be specific, I want to use the AR system as a capturing system to create the knowledge representations. Facilitated by AR, I can externalize our intentions more efficiently. Further, the communication and knowledge transfer may not be limited to human-to-human but also human-robot, and robot-robot collaborations.

- One effective way for creating knowledge representations and AR guidance is by demonstrating. But most of the current explorations remain in a post-processing fashion. For example, usually an instruction video is recorded, then the AR guidance is created offline manually. I envision a real-time instant guidance creation through

with the help from AR itself, i.e., use AR for demonstrating actions, capturing the data and segmenting the data.

- Currently, AR authoring system usually are focused on single-directional instructions. For instance, users communicate his/her intention to the other one. However, I see a gap for communications in collaborative tasks. If one of the collaborator is less familiar with the task context, one common way to communicate the coordinated plan is through demonstration. Users need to complete such a task through both spatial and temporal coordination. I propose an AR demonstration system where a user (User A) first acts out his/her planned motion through AR, then acts out the collaborator's (User B) motion through AR. In particular, the externalized motions of user A can further be used as references while demonstrating User B's part. This way, I unleash the power of AR for space and time manipulation for demonstrating a collaborative task plan.

- I further envision the above AR demonstrating system can be generalized to many other collaborative scenarios. One very interesting scenario is human-robot collaboration since creating a spatial aware context for the robots could be challenging. But using the demonstration system, users can transfer their plan for the robots accurately and effortlessly. More importantly, the knowledge transfer is almost transparent. For another example, both collaborative agents are robots, the user acts each robot's task in turns.

8.3 Conclusions

Amid the rapid developing AR technologies, especially the emerging commercially available AR devices and SDKs, the basic concept of AR has become prevalent to a larger population. Although there are quite a few impressive AR concepts prototyped as research projects, the majority of AR applications available on the commercially available platforms remain simple and are suitable mainly for demonstration purposes. Although these proliferating platforms bring the basic concept of AR closer to the public, we still lack serious

AR application scenarios, or in other word, the killer applications. It has a long way to go before unleashing the true power of AR and changing the way people interact with the world.

Through this thesis, I contribute towards this goal by (i) exploring spatially-aware interactions, (ii) studying enabling techniques to infer the physical world into AR, and (iii) applying the findings in a wide range of spatial AR tasks. Further, I extend the developed technologies and interactions into a more generalized area: enabling autonomous navigation for a robot in an IoT environment, and building an AR authoring interface for robots to conduct spatially distributed tasks.

In this last section, I mainly focus on discussing the retrospectives about the different phases of this thesis, and suggesting the possible research directions based on the takeaways. Essentially, I identify some unique features which can be leveraged to spot future AR research projects and impactful AR application scenarios, namely, situatedness, mobility, and the capability to support collaborations. I also recommend some concrete projects exploiting these unique features: creating a human-in-the-loop simulation through AR, sharing contexts between heterogeneous agents including humans, robots, and IoT devices, and building a transparent knowledge transferring interface through AR for human-to-human instructing, human-robot collaborations, and multi-robots collaboration.

REFERENCES

[1] Ifttt, 2018. https://ifttt.com/.

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

[3] 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*, 17(4):2347–2376, 2015.

[4] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. A. Al-Ammar, and H. S. Al-Khalifa. Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors*, 16(5):707, 2016.

[5] I. Amundson and X. Koutsoukos. A survey on localization for mobile wireless sensor networks. *Mobile entity localization and tracking in GPS-less environnments*, pages 235–254, 2009.

[6] M. Annett, F. Anderson, W. F. Bischof, and A. Gupta. The pen is mightier: understanding stylus behaviour while inking on tablets. In *Proceedings of the 2014 Graphics Interface Conference (GI'14)*, pages 193–200, 2014.

[7] Apple. Arkit, 2017. Retrieved September 1, 2017 from https://developer.apple.com/arkit/.

[8] F. Argelaguet and C. Andujar. Visual feedback techniques for virtual pointing on stereoscopic displays. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, pages 163–170. ACM, 2009.

[9] R. Arisandi, M. Otsuki, A. Kimura, F. Shibata, and H. Tamura. Virtual handcrafting: Building virtual wood models using tooldevice. *Proceedings of the IEEE*, 102(2):185–195, 2014.

[10] C. Arth, C. Pirchheim, J. Ventura, D. Schmalstieg, and V. Lepetit. Instant outdoor localization and slam initialization from 2.5 d maps. *IEEE transactions on visualization and computer graphics*, 21(11):1309–1318, 2015.

[11] ARToolKit. Artoolkit, 2017. Retrieved Dec 1, 2017 from https://www.hitl.washington.edu/artoolkit/.

[12] D. Ashbrook, P. Baudisch, and S. White. Nenya: subtle and eyes-free mobile input with a magnetically-tracked finger ring. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems (CHI'11)*, pages 2043–2046, 2011.

[13] ASUS. Zenfone-ar, 2017. Retrieved September 1, 2017 from https://www.asus.com/us/Phone/ZenFone-AR-ZS571KL/.

[14] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and virtual environments*, 6(4):355–385, 1997.

[15] S. K. Badam, S. Chandrasegaran, N. Elmqvist, and K. Ramani. Tracing and sketching performance using blunt-tipped styli on direct-touch tablets. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pages 193–200. ACM, 2014.

[16] S. Bai and H. Qi. Tackling the flip ambiguity in wireless sensor network localization and beyond. *Digital Signal Processing*, 55:85–97, 2016.

[17] Y. Baillot, L. Davis, and J. Rolland. A survey of tracking technology for virtual environments. *Fundamentals of wearable computers and augumented reality*, page 67, 2001.

[18] T. Ballendat, N. Marquardt, and S. Greenberg. Proxemic interaction: designing for a proximity and orientation-aware environment. In *ACM International Conference on Interactive Tabletops and Surfaces*, pages 121–130. ACM, 2010.

[19] T. Baudel. A mark-based interaction paradigm for free-hand drawing. In *Proc. of UIST*, pages 185–192. ACM, 1994.

[20] S. Benford, C. Greenhalgh, G. Reynard, C. Brown, and B. Koleva. Understanding and constructing shared spaces with mixed-reality boundaries. *ACM Transactions on computer-human interaction (TOCHI)*, 5(3):185–223, 1998.

[21] H. Benko, E. W. Ishak, and S. Feiner. Collaborative mixed reality visualization of an archaeological excavation. In *Proceedings of the 3rd IEEE/ACM international Symposium on Mixed and Augmented Reality*, pages 132–140. IEEE Computer Society, 2004.

[22] A. Bianchi and I. Oakley. Magnid: Tracking multiple magnetic tokens. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction (TEI'15)*, pages 61–68, 2015.

[23] M. Billinghurst, A. Clark, G. Lee, et al. A survey of augmented reality. *Foundations and Trends® in Human–Computer Interaction*, 8(2-3):73–272, 2015.

[24] M. Billinghurst, H. Kato, and S. Myojin. Advanced interaction techniques for augmented reality applications. *Virtual and Mixed Reality*, pages 13–22, 2009.

[25] M. Billinghurst, I. Poupyrev, H. Kato, and R. May. Mixing realities in shared space: An augmented reality interface for collaborative computing. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 3, pages 1641–1644. IEEE, 2000.

[26] I. Borg and P. J. Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

[27] S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch. Touch projector: mobile interaction through video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2287–2296. ACM, 2010.

[28] P. Bouchier. Razor 9DOF AHRS, 2013. Retrieved August 1, 2015 from `https://github.com/ptrbrtz/razor-9dof-ahrs`.

[29] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision*, pages 536–551. Springer, 2014.

[30] B. Brumitt, J. Krumm, B. Meyers, and S. Shafer. Ubiquitous computing and the role of geometry. *IEEE Personal Communications*, 7(5):41–43, 2000.

[31] A. Butler, S. Izadi, and S. Hodges. Sidesight: multi-touch interaction around small devices. In *Proceedings of the 21st annual ACM symposium on User interface software and technology (UIST'08)*, pages 201–204, 2008.

[32] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[33] 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*, pages 419–428. ACM, 2018.

[34] R. Castle, G. Klein, and D. W. Murray. Video-rate localization in multiple maps for wearable augmented reality. In *Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium on*, pages 15–22. IEEE, 2008.

[35] L. Chan, R.-H. Liang, M.-C. Tsai, K.-Y. Cheng, C.-H. Su, M. Y. Chen, W.-H. Cheng, and B.-Y. Chen. Fingerpad: private and subtle interaction using fingertips. In *Proceedings of the 26th annual ACM symposium on User interface software and technology (UIST'13)*, pages 255–260, 2013.

[36] K.-Y. Chen, K. Lyons, S. White, and S. Patel. utrack: 3d input using two magnetic sensors. In *Proceedings of the 26th annual ACM symposium on User interface software and technology (UIST'13)*, pages 237–244, 2013.

[37] X. Chen, J. Schwarz, C. Harrison, J. Mankoff, and S. E. Hudson. Air+ touch: interweaving touch & in-air gestures. In *Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST'14)*, pages 519–525, 2014.

[38] Y.-H. Chen, B. Zhang, C. Tuna, Y. Li, E. A. Lee, and B. Hartmann. A context menu for the real world: Controlling physical appliances through head-worn infrared targeting. Technical report, CALIFORNIA UNIV BERKELEY DEPT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCES, 2013.

[39] H.-L. Chi, S.-C. Kang, and X. Wang. Research trends and opportunities of augmented reality applications in architecture, engineering, and construction. *Automation in construction*, 33:116–122, 2013.

[40] 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*, page 274. ACM, 2018.

[41] 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*, pages 5313–5320. IEEE, 2012.

[42] codelabs. Webrtc, 2017. Retrieved Dec 1, 2017 from `https://webrtc.org/`.

[43] J. A. Costa, N. Patwari, and A. O. Hero III. Distributed weighted-multidimensional scaling for node localization in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 2(1):39–64, 2006.

[44] A. A. de Freitas, M. Nebeling, X. Chen, J. Yang, A. S. K. Karthikeyan Ranithangam, and A. K. Dey. Snap-to-it: A user-inspired platform for opportunistic device interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5909–5920. ACM, 2016.

[45] J. De Leeuw and P. Mair. Multidimensional scaling using majorization: Smacof in r. *Department of Statistics, UCLA*, 2011.

[46] decaWave. Production documentation, 2017. Retrieved Dec 1, 2017 from `https://www.decawave.com/product-documentation`.

[47] T. Deyle, M. S. Reynolds, and C. C. Kemp. Finding and navigating to household objects with uhf rfid tags by optimizing rf signal strength. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.*, pages 2579–2586, 2014.

[48] C. Di Franco, E. Bini, M. Marinoni, and G. C. Buttazzo. Multidimensional scaling localization with anchors. In *Autonomous Robot Systems and Competitions (ICARSC), 2017 IEEE International Conference on*, pages 49–54. IEEE, 2017.

[49] C. Di Franco, A. Prorok, N. Atanasov, B. P. Kempke, P. Dutta, V. Kumar, and G. J. Pappas. Calibration-free network localization using non-line-of-sight ultra-wideband measurements. In *IPSN*, pages 235–246, 2017.

[50] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli. Euclidean distance matrices: Essential theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 32(6):12–30, 2015.

[51] V. Domiter and B. Žalik. Sweep-line algorithm for constrained delaunay triangulation. *International Journal of Geographical Information Science*, 22(4):449–462, 2008.

[52] M.-P. Dubuisson and A. K. Jain. A modified hausdorff distance for object matching. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 566–568. IEEE, 1994.

[53] Esspressif. Esp32, 2017. Retrieved Dec 1, 2017 from `https://www.espressif.com/en/products/hardware/esp32/overview`.

[54] M. Fiorentino, R. de Amicis, G. Monno, and A. Stork. Spacedesign: A mixed reality workspace for aesthetic industrial design. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, page 86. IEEE Computer Society, 2002.

[55] P. Fleck, C. Arth, C. Pirchheim, and D. Schmalstieg. [poster] tracking and mapping with a swarm of heterogeneous clients. In *Mixed and Augmented Reality (ISMAR), 2015 IEEE International Symposium on*, pages 136–139. IEEE, 2015.

[56] 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*, pages 282–287. IEEE, 2011.

[57] S. R. Fussell, R. E. Kraut, and J. Siegel. Coordination of communication: Effects of shared visual context on collaborative work. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 21–30. ACM, 2000.

[58] H. Gardner. The theory of multiple intelligences. *Annals of dyslexia*, 37(1):19–35, 1987.

[59] S. Gauglitz, B. Nuernberger, M. Turk, and T. Höllerer. World-stabilized annotations and virtual scene navigation for remote collaboration. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 449–459. ACM, 2014.

[60] H. Gellersen, C. Fischer, D. Guinard, R. Gostner, G. Kortuem, C. Kray, E. Rukzio, and S. Streng. Supporting device discovery and spontaneous interaction with spatial references. *Personal and Ubiquitous Computing*, 13(4):255–264, 2009.

[61] Google. Cardboard, 2013. Retrieved August 1, 2015 from `https://www.google.com/get/cardboard`.

[62] Google. Tango developer overview, 2016. Retrieved August 1, 2016 from `https://developers.google.com/tango/developer-overview`.

[63] Google. Arcore, 2017. Retrieved September 1, 2017 from `https://developers.google.com/ar/`.

[64] Google. Google glass, 2017. Retrieved September 1, 2017 from `https://www.x.company/glass/`.

[65] J. Grubert, T. Langlotz, S. Zollmann, and H. Regenbrecht. Towards pervasive augmented reality: Context-awareness in augmented reality. *IEEE transactions on visualization and computer graphics*, 23(6):1706–1724, 2017.

[66] U. Gruenefeld, A. E. Ali, W. Heuten, and S. Boll. Visualizing out-of-view objects in head-mounted augmented reality. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, page 81. ACM, 2017.

[67] J. Gugenheimer, E. Stemasov, J. Frommel, and E. Rukzio. Sharevr: Enabling co-located experiences for virtual reality between hmd and non-hmd users. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4021–4033. ACM, 2017.

[68] A. Guo, J. Kim, X. Chen, T. Yeh, S. E. Hudson, J. Mankoff, and J. P. Bigham. Facade: Auto-generating tactile interfaces to appliances. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 5826–5838. ACM, 2017.

[69] D. Hahnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose. Mapping and localization with rfid technology. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 1015–1020. IEEE, 2004.

[70] X. Han, H. Seki, Y. Kamiya, and M. Hikizu. Wearable handwriting input device using magnetic field: Geomagnetism cancellation in position calculation. *Precision engineering*, 33(1):37–43, 2009.

[71] C. Hand. A survey of 3d interaction techniques. In *Computer graphics forum*, volume 16, pages 269–281. Wiley Online Library, 1997.

[72] N. B. Hansen and P. Dalsgaard. The productive role of material design artefacts in participatory design events. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, pages 665–674. ACM, 2012.

[73] C. Harrison and S. E. Hudson. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology (UIST'09)*, pages 121–124, 2009.

[74] 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.

[75] M. Hazas, C. Kray, H. Gellersen, H. Agbota, G. Kortuem, and A. Krohn. A relative positioning system for co-located mobile devices. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 177–190. ACM, 2005.

[76] M. Hegarty. Components of spatial intelligence. In *Psychology of Learning and Motivation*, volume 52, pages 265–297. Elsevier, 2010.

[77] S. R. Herring, C.-C. Chang, J. Krantzler, and B. P. Bailey. Getting inspired!: Understanding how and why examples are used in creative design practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 87–96, New York, NY, USA, 2009. ACM.

[78] 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*, pages 307–310. ACM, 2013.

[79] 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*, pages 307–310. ACM, 2013.

[80] K. Heurtefeux and F. Valois. Is rssi a good choice for localization in wireless sensor network? In *IEEE 26th Int. Conf. on Advanced Information Networking and Applications*, pages 732–739, 2012.

[81] K. Hinckley, M. Pahud, H. Benko, P. Irani, F. Guimbretière, M. Gavriliu, X. A. Chen, F. Matulic, W. Buxton, and A. Wilson. Sensing techniques for tablet+stylus interaction. In *Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST'14)*, pages 605–614, 2014.

[82] T. Höllerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway. Exploring mars: developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers & Graphics*, 23(6):779–785, 1999.

[83] A. Hook, P. Fite-Georgel, M. Meisnieks, A. Maes, M. Gardeya, and L. Naimark. Generation and sharing coordinate system between users on mobile, Sept. 18 2014. US Patent App. 13/835,822.

[84] J. Huang, T. Mori, K. Takashima, S. Hashi, and Y. Kitamura. Im6d: magnetic tracking system with 6-dof passive markers for dexterous 3d interaction and motion. *ACM Transactions on Graphics (TOG)*, 34(6):217, 2015.

[85] C. Hummels and J. Frens. The reflective transformative design process. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 2655–2658. ACM, 2009.

[86] K. Huo, Y. Cao, S. 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*, pages –. ACM, 2018.

[87] K. Huo, K. Ramani, et al. Window-shaping: 3d design ideation by creating on, borrowing from, and looking at the physical world. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*, pages 37–45. ACM, 2017.

[88] 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*, pages 19–30. ACM, 2018.

[89] S. Hwang, A. Bianchi, M. Ahn, and K. Wohn. Magpen: magnetically driven pen interactions on and around conventional smartphones. In *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'13)*, pages 412–415, 2013.

[90] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3d freeform design. In *Acm siggraph 2007 courses*, page 21. ACM, 2007.

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

[92] H. Ishii and B. Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 234–241. ACM, 1997.

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

[94] Y. Jiang and V. C. Leung. An asymmetric double sided two-way ranging for crystal offset. In *Signals, Systems and Electronics, 2007. ISSSE'07. International Symposium on*, pages 525–528. IEEE, 2007.

[95] B. Jones, R. Sodhi, D. Forsyth, B. Bailey, and G. Maciocci. Around device interaction for multiscale navigation. In *Proceedings of the 14th international conference on Human computer interaction with mobile devices and services (MobileHCI'12)*, pages 83–92, 2012.

[96] S. Kasahara, V. Heun, A. S. Lee, and H. Ishii. Second surface: multi-user spatial collaboration system based on augmented reality. In *SIGGRAPH Asia 2012 Emerging Technologies*, page 20. ACM, 2012.

[97] 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*, pages 223–228. ACM, 2013.

[98] D. F. Keefe, D. A. Feliz, T. Moscovich, D. H. Laidlaw, and J. J. LaViola Jr. Cavepainting: a fully immersive 3d artistic medium and interactive experience. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 85–93. ACM, 2001.

[99] C. C. Kemp, A. Edsinger, and E. Torres-Jara. Challenges for robot manipulation in human environments [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, 14(1):20–29, 2007.

[100] H. Ketabdar, M. Roshandel, and K. A. Yüksel. Magiwrite: towards touchless digit entry using 3d space around mobile devices. In *Proceedings of the 12th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'10)*, pages 443–446, 2010.

[101] I. Khalfin and H. S. Jones Jr. Electromagnetic position and orientation tracking system with distortion compensation employing wireless sensors, Apr. 2 2002. U.S. Patent 6,369,564.

[102] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[103] P.-O. Kristensson and S. Zhai. Shark^2: A large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, pages 43–52, New York, NY, USA, 2004. ACM.

[104] N. Kyriazis and A. Argyros. Scalable 3d tracking of multiple interacting objects. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3430–3437. IEEE, 2014.

[105] D. Lakatos, M. Blackshaw, A. Olwal, Z. Barryte, K. Perlin, and H. Ishii. T (ether): spatially-aware handhelds, gestures and proprioception for multi-user 3d modeling and animation. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*, pages 90–93. ACM, 2014.

[106] D. Lakatos, M. Blackshaw, A. Olwal, Z. Barryte, K. Perlin, and H. Ishii. T(ether): Spatially-aware handhelds, gestures and proprioception for multi-user 3d modeling and animation. In *Proceedings of the 2nd ACM Symposium on Spatial User Interaction*, SUI '14, pages 90–93, New York, NY, USA, 2014. ACM.

[107] G. Laput, C. Yang, R. Xiao, A. Sample, and C. Harrison. Em-sense: Touch recognition of uninstrumented, electrical and electromechanical objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 157–166. ACM, 2015.

[108] M. Lau, M. Hirose, A. Ohgawara, J. Mitani, and T. Igarashi. Situated modeling: A shape-stamping interface with tangible primitives. In *ACM Conference on Tangible, Embedded and Embodied Interaction (TEI '12)*, pages 275–282, 2012.

[109] D. Ledo, S. Greenberg, N. Marquardt, and S. Boring. Proxemic-aware controls: Designing remote controls for ubiquitous computing ecologies. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 187–198. ACM, 2015.

[110] 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*, pages 1–8. IEEE, 2012.

[111] Lenovo. Phab 2 pro, 2016. Retrieved August 1, 2016 from `http://shop.lenovo.com/us/en/tango/`.

[112] R.-H. Liang, K.-Y. Cheng, C.-H. Su, C.-T. Weng, B.-Y. Chen, and D.-N. Yang. Gausssense: attachable stylus sensing using magnetic sensor grid. In *Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST'12)*, pages 319–326, 2012.

[113] 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*, 16(3):872–885, 2017.

[114] Y.-T. Lin, Y.-C. Liao, S.-Y. Teng, Y.-J. Chung, L. Chan, and B.-Y. Chen. Outside-in: Visualizing out-of-sight regions-of-interest in a 360 video using spatial picture-in-picture previews. In *Proceedings of the 30th Annual Symposium on User Interface Software and Technology*, pages –. ACM, 2017.

[115] magic leap. Magic leap, 2019. Retrieved Jan. 1, 2019 from `https://www.magicleap.com/`.

[116] 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*, pages 153–158. ACM, 2015.

[117] S. Magnenat, D. T. Ngo, F. Zund, M. Ryffel, G. Noris, G. Rothlin, A. Marra, M. Nitti, P. Fua, M. Gross, and R. W. Sumner. Live texturing of augmented reality characters from colored drawings. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):1201–1210, Nov. 2015.

[118] G. Mao, B. Fidan, and B. D. Anderson. Wireless sensor network localization techniques. *Computer networks*, 51(10):2529–2553, 2007.

[119] MathWorks. Matlab optimization toolbox, 2017. Retrieved Dec 1, 2017 from `https://www.mathworks.com/products/optimization.html`.

[120] 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*, pages 239–242. ACM, 2013.

[121] Microsoft. Surface, 2012. Retrieved August 1, 2015 from https://www.microsoft.com/surface.

[122] Microsoft. Hololens, 2017. Retrieved September 1, 2017 from https://www.microsoft.com/en-us/hololens.

[123] G. Mohanarajah, V. Usenko, M. Singh, R. D'Andrea, and M. Waibel. Cloud-based collaborative 3d mapping in real-time with low-cost robots. *IEEE Transactions on Automation Science and Engineering*, 12(2):423–431, 2015.

[124] J. Müller, R. Rädle, and H. Reiterer. Virtual objects as spatial cues in collaborative mixed reality environments: How they shape communication behavior and user task load. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1245–1249. ACM, 2016.

[125] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. Fibermesh: Designing freeform surfaces with 3d curves. In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[126] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *10th IEEE Int. Symp. on Mixed and Augmented Reality*, pages 127–136, 2011.

[127] 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*, pages 467–474. IEEE, 2013.

[128] J. Nocedal and S. J. Wright. *Sequential quadratic programming*. Springer, 2006.

[129] B. Nuernberger, K.-C. Lien, M. Turk, et al. Interpreting 2d gesture annotations in 3d augmented reality. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 149–158. IEEE, 2016.

[130] B. Nuernberger, E. Ofek, H. Benko, and A. D. Wilson. Snaptoreality: Aligning augmented reality to the real world. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1233–1244. ACM, 2016.

[131] NVIDIA. Shield, 2013. Retrieved August 1, 2015 from http://shield.nvidia.com/tablet.

[132] O. Oda, C. Elvezio, M. Sukan, S. Feiner, and B. Tversky. Virtual replicas for remote assistance in virtual and augmented reality. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 405–415. ACM, 2015.

[133] L. Olsen, F. Samavati, and J. Jorge. Naturasketch: Modeling from images and natural sketches. *IEEE Comput. Graph. Appl.*, 31(6):24–34, Nov. 2011.

[134] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, et al. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 741–754. ACM, 2016.

[135] M. Otsuki, K. Sugihara, A. Kimura, F. Shibata, and H. Tamura. Mai painting brush: an interactive device that realizes the feeling of real painting. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 97–100. ACM, 2010.

[136] G. Pan, J. Wu, D. Zhang, Z. Wu, Y. Yang, and S. Li. Geeair: a universal multimodal remote control device for home appliances. *Personal and Ubiquitous Computing*, 14(8):723–735, 2010.

[137] 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*, pages 1716–1725. ACM, 2016.

[138] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials*, 16(1):414–454, 2014.

[139] S. Pillai and J. Leonard. Monocular slam supported object recognition. *arXiv preprint arXiv:1506.01732*, 2015.

[140] C. Piya, Vinayak, Y. Zhang, and K. Ramani. Realfusion: An interactive workflow for repurposing real-world objects towards early-stage creative ideation. In *Proceedings of Graphics Interface 2016*. ACM, 2016.

[141] Polhemus. Polhemus, 2017. Retrieved Dec 1, 2017 from `https://polhemus.com/applications/electromagnetics/`.

[142] M. Project. Math.net numerics, 2017. Retrieved September 1, 2017 from `https://numerics.mathdotnet.com/`.

[143] K. Qian, C. Wu, Z. Zhou, Y. Zheng, Z. Yang, and Y. Liu. Inferring motion direction using commodity wi-fi for interactive exergames. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1961–1972. ACM, 2017.

[144] quuppa. Quuppa technology, 2017. Retrieved Dec 1, 2017 from `http://quuppa.com/`.

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

[146] G. Reitmayr and D. Schmalstieg. *Collaborative augmented reality for outdoor navigation and information browsing*. na, 2004.

[147] J. Rekimoto, Y. Ayatsuka, M. Kohno, and H. Oba. Proximal interactions: A direct manipulation technique for wireless networking. In *Interact*, volume 3, pages 511–518, 2003.

[148] L. Riazuelo, J. Civera, and J. Montiel. C2tam: A cloud framework for cooperative tracking and mapping. *Robotics and Autonomous Systems*, 62(4):401–413, 2014.

[149] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, Aug. 2004.

[150] A. Rubin and J. P. Bellinghausen. Passive and active video game controllers with magnetic position sensing, Dec. 31 2013. U.S. Patent 8,616,974.

[151] E. Rukzio, K. Leichtenstern, V. Callaghan, P. Holleis, A. Schmidt, and J. Chin. An experimental comparison of physical mobile interaction techniques: Touching, pointing and scanning. *UbiComp 2006: Ubiquitous Computing*, pages 87–104, 2006.

[152] S. Russell, G. Dublon, and J. A. Paradiso. Hearthere: Networked sensory prosthetics through auditory augmented reality. In *Proceedings of the 7th Augmented Human International Conference 2016*, page 20. ACM, 2016.

[153] E. Sachs, A. Roberts, and D. Stoops. 3-draw: A tool for designing 3d shapes. *IEEE Computer Graphics and Applications*, 11(6):18–26, 1991.

[154] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1352–1359, 2013.

[155] Samsung. Galaxy note edge, 2014. Retrieved August 1, 2015 from `http://www.samsung.com/global/microsite/galaxynoteedge`.

[156] N. Savinov, A. Dosovitskiy, and V. Koltun. Semi-parametric topological memory for navigation. *arXiv preprint arXiv:1803.00653*, 2018.

[157] S. Schkolne, M. Pruett, and P. Schröder. Surface drawing: creating organic 3d shapes with the hand and tangible tools. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 261–268. ACM, 2001.

[158] D. Schmalstieg, A. Fuhrmann, G. Hesina, Z. Szalavári, L. M. Encarnaçao, M. Gervautz, and W. Purgathofer. The studierstube augmented reality project. *Presence: Teleoperators & Virtual Environments*, 11(1):33–54, 2002.

[159] D. Schmidt, D. Molyneaux, and X. Cao. Picontrol: using a handheld projector for direct control of physical devices through visible light. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 379–388. ACM, 2012.

[160] K. Schmidt and C. Simonee. Coordination mechanisms: Towards a conceptual foundation of cscw systems design. *Computer Supported Cooperative Work (CSCW)*, 5(2-3):155–200, 1996.

[161] 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*, pages 1607–1614. ACM, 2016.

[162] U. Schultheis, J. Jerald, F. Toledo, A. Yoganandan, and P. Mlyniec. Comparison of a two-handed interface to a wand interface and a mouse interface for fundamental 3d tasks. In *2012 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 117–124, 2012.

[163] S. H. Seo, J. E. Young, and P. Irani. Where are the robots? in-feed embedded techniques for visualizing robot team member locations. In *Robot and Human Interactive Communication (RO-MAN), 2017 26th IEEE International Symposium on*, pages 522–527. IEEE, 2017.

[164] P. Simoens, M. Dragone, and A. Saffiotti. The internet of robotic things: A review of concept, added value and applications. *International Journal of Advanced Robotic Systems*, 15(1):1729881418759424, 2018.

[165] Sixense. Sixense, 2017. Retrieved Dec 1, 2017 from `https://www.sixense.com/platform/hardware/`.

[166] R. S. Sodhi, B. R. Jones, D. Forsyth, B. P. Bailey, and G. Maciocci. Bethere: 3d mobile collaboration with spatial input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 179–188, New York, NY, USA, 2013. ACM.

[167] H. Song, H. Benko, F. Guimbretiere, S. Izadi, X. Cao, and K. Hinckley. Grips and gestures on a multi-touch pen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'11)*, pages 1323–1332, 2011.

[168] H. Song, F. Guimbretière, C. Hu, and H. Lipson. Modelcraft: Capturing freehand annotations and edits on physical 3d models. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pages 13–22, New York, NY, USA, 2006. ACM.

[169] J. Song, G. Sörös, F. Pece, S. R. Fanello, S. Izadi, C. Keskin, and O. Hilliges. In-air gestures around unmodified mobile devices. In *Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST'14)*, pages 319–329, 2014.

[170] S. Sridhar, F. Mueller, M. Zollhöfer, D. Casas, A. Oulasvirta, and C. Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *European Conference on Computer Vision*, pages 294–310. Springer, 2016.

[171] Y. Thiel, K. Singh, and R. Balakrishnan. Elasticurves: Exploiting stroke dynamics and inertia for the real-time neatening of sketched 2d curves. In *Proc. of UIST*, pages 383–392. ACM, 2011.

[172] B. H. Thomas, G. F. Welch, P. Dragicevic, N. Elmqvist, P. Irani, Y. Jansen, D. Schmalstieg, A. Tabard, N. A. ElSayed, R. T. Smith, et al. Situated analytics. In *Immersive Analytics*, pages 185–220. Springer, 2018.

[173] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[174] ubiquity6. ubiquity6, 2018. Retrieved July 1, 2018 from `http://ubiquity6.com/`.

[175] Unity3D. Unity3d, 2017. Retrieved September 1, 2017 from `https://unity3d.com/`.

[176] P. Välkkynen and T. Tuomisto. Physical browsing research. *PERMID*, 2005:35–38, 2005.

[177] S. Venkatesh and R. Buehrer. Non-line-of-sight identification in ultra-wideband systems based on received signal statistics. *IET Microwaves, Antennas & Propagation*, 1(6):1120–1130, 2007.

[178] 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.

[179] Vinayak, D. Ramanujan, C. Piya, and K. Ramani. Mobisweep: Exploring spatial design ideation using a smartphone as a hand-held reference plane. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '16, pages 12–20, New York, NY, USA, 2016. ACM.

[180] Vuforia. Vuforia, 2017. Retrieved September 1, 2017 from `https://www.vuforia.com/`.

[181] C. Wacharamanotham, K. Todi, M. Pye, and J. Borchers. Understanding finger input above desktop devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'14)*, pages 1083–1092, 2014.

[182] S. Wahid, S. M. Branham, D. S. McCrickard, and S. Harrison. Investigating the relationship between imagery and rationale in design. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, DIS '10, pages 75–84, New York, NY, USA, 2010. ACM.

[183] J. Wai, D. Lubinski, and C. P. Benbow. Spatial ability for stem domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. *Journal of Educational Psychology*, 101(4):817, 2009.

[184] E. J. Wang, T.-J. Lee, A. Mariakakis, M. Goel, S. Gupta, and S. N. Patel. Magnifisense: Inferring device interaction using wrist-worn passive magneto-inductive sensors. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 15–26. ACM, 2015.

[185] R. Want, K. P. Fishkin, A. Gujar, and B. L. Harrison. Bridging physical and virtual worlds with electronic tags. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 370–377. ACM, 1999.

[186] C. Weichel, J. Alexander, A. Karnik, and H. Gellersen. Spata: Spatio-tangible tools for fabrication-aware design. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction (TEI'15)*, pages 189–196, 2015.

[187] C. Weichel, M. Lau, D. Kim, N. Villar, and H. W. Gellersen. Mixfab: A mixed-reality environment for personal fabrication. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 3855–3864, New York, NY, USA, 2014. ACM.

[188] C. Weichel, M. Lau, D. Kim, N. Villar, and H. W. Gellersen. Mixfab: a mixed-reality environment for personal fabrication. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 3855–3864. ACM, 2014.

[189] M. Weiser. The computer for the 21st century, 1999. Retrieved September 1, 2017 from `http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html`.

[190] Wikitude. Wikitude, 2017. Retrieved September 1, 2017 from `https://www.wikitude.com/`.

[191] C.-J. Wu, S. Houben, and N. Marquardt. Eaglesense: Tracking people and devices in interactive spaces using real-time top-view depth-sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3929–3942. ACM, 2017.

[192] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2411–2418, 2013.

[193] R. Xiao, G. Laput, Y. Zhang, and C. Harrison. Deus em machina: On-touch contextual functionality for smart iot appliances. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4000–4008. ACM, 2017.

[194] M. Xin, E. Sharlin, and M. C. Sousa. Napkin sketch: Handheld mixed reality 3d sketching. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, VRST '08, pages 223–226, New York, NY, USA, 2008. ACM.

[195] N. Yadav and C. Bleakley. Accurate orientation estimation using ahrs under conditions of magnetic distortion. *Sensors*, 14(11):20008–20024, 2014.

[196] O. Yeniay. A comparative study on optimization methods for the constrained nonlinear programming problems. *Mathematical Problems in Engineering*, 2005(2):165–173, 2005.

[197] S. H. Yoon, K. Huo, and K. Ramani. Tmotion: Embedded 3d mobile input using magnetic sensing technique. In *Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 21–29. ACM, 2016.

[198] S. H. Yoon, Y. Zhang, K. Huo, and K. Ramani. Tring: Instant and customizable interactions with objects using an embedded magnet and a finger-worn device. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 169–181. ACM, 2016.

[199] 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*, pages 173–182. ACM, 2009.

[200] F. Zhou, H. B.-L. Duh, and M. Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 193–202. IEEE Computer Society, 2008.

[201] F. Zünd, M. Ryffel, S. Magnenat, A. Marra, M. Nitti, M. Kapadia, G. Noris, K. Mitchell, M. Gross, and R. W. Sumner. Augmented creativity: Bridging the real and virtual worlds to enhance creative play. In *SIGGRAPH Asia 2015 Mobile Graphics and Interactive Applications*, SA '15, pages 21:1–21:7, New York, NY, USA, 2015. ACM.

VITA

Ke Huo is a Ph.D. student in the School of Mechanical Engineering at Purdue. Prior to joining the C Design Lab, he obtained his Bachelor's degree from Beihang University and subsequently Master's from University of Florida both in Aerospace engineering, worked on projects involving dynamics and control of electro-mechanical system. His current research interests include augmented reality, tangible interfaces, interactive creation, and robotics. Generally, he is fascinated by the intersection point where our physical world can be leveraged by digital intelligence.