ProGesAR: Mobile AR Prototyping for Proxemic and Gestural Interactions with Real-world IoT Enhanced Spaces

Hui Ye
School of Creative Media, City University of Hong Kong
Hong Kong, China
huiye4-c@my.cityu.edu.hk

Hongbo Fu*
School of Creative Media, City University of Hong Kong
Hong Kong, China
hongbofu@cityu.edu.hk

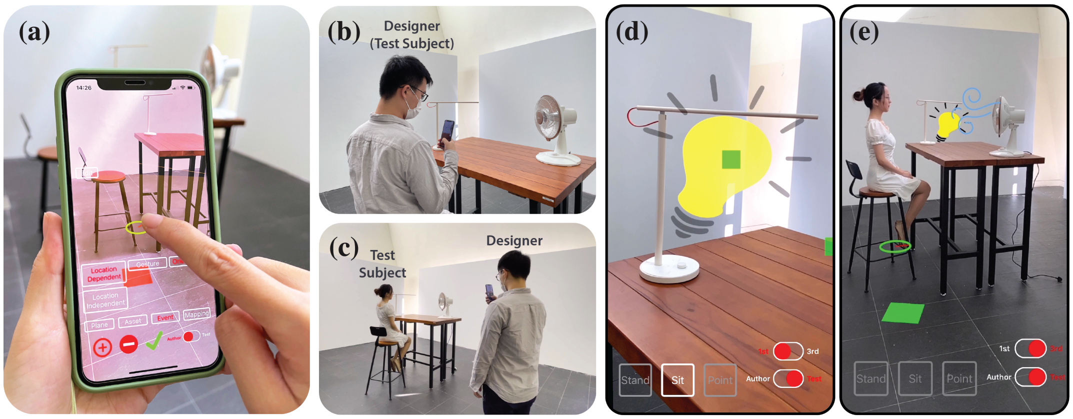


Figure 1: *ProGesAR* is a mobile AR prototyping tool (a), allowing a designer to easily and quickly prototype proxemic and gestural interactions of a test subject in a real environment from a designer's perspective, even without the presence of the test subject during the authoring process. The prototyped interactions can be examined from either a potential user's view (b)(d) or a designer's view (c)(e). In this example, when a test subject approaches the desk, a location-independent event (indicated by the square on the ground) is triggered to show a light brightening effect; when the test subject sits down on the chair and faces forward, a location-dependent event (indicated by the circle on the ground) is triggered to show a shaking effect of a wind asset. See the accompanying video for the demo.

ABSTRACT

Real-world IoT enhanced spaces involve diverse proximity- and gesture-based interactions between users and IoT devices/objects. Prototyping such interactions benefits various applications like the conceptual design of ubicomp space. AR (Augmented Reality) prototyping provides a flexible way to achieve early-stage designs by overlaying digital contents on real objects or environments. However, existing AR prototyping approaches have focused on prototyping AR experiences or context-aware interactions from the first-person view instead of full-body proxemic and gestural (*pro-ges* for short) interactions of real users in the real world. In this work, we conducted interviews to figure out the challenges of prototyping *pro-ges* interactions in real-world IoT enhanced spaces. Based on the findings, we present *ProGesAR*, a mobile AR tool for prototyping *pro-ges* interactions of a subject in a real environment from a third-person view, and examining the prototyped interactions from both the first- and third- person views. Our interface supports the effects of virtual assets dynamically triggered by a single subject, with the triggering events based on four features: location, orientation, gesture, and distance. We conduct a preliminary study by inviting participants to prototype in a freeform manner using *ProGesAR*. The early-stage findings show that with *ProGesAR*, users can easily and quickly prototype their design ideas about *pro-ges* interactions.

CCS CONCEPTS

- **Human-centered computing** → *Interaction techniques*; **Mixed/augmented reality**; **Graphical user interfaces**; *User interface toolkits*; **Mobile devices**.

KEYWORDS

Proxemic interaction; Gestural interaction; AR prototyping; Mobile augmented reality

ACM Reference Format:
Hui Ye and Hongbo Fu. 2022. *ProGesAR*: Mobile AR Prototyping for Proxemic and Gestural Interactions with Real-world IoT Enhanced Spaces. In *CHI*

---

[*] Corresponding author.

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3517689

Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3491102.3517689

1 INTRODUCTION

Real-world IoT (Internet of Things) enhanced spaces involves diverse proxemic [3, 45] and gestural [36, 56] (*pro-ges* for short) interactions between users and IoT devices/objects [54]. Such interactions mediate people's interactions in a room-sized ubicomp ecology, thus facilitating seamless and natural interactions [44] between users and nearby IoT devices/objects. Designing such interactions (e.g., design device functions responsive to users' proximity and gestures) is an emerging need in IoT industries. Since these interactions involve one or multiple users interacting with multiple devices and/or objects in physical environments, prototyping such interactions often requires specialized hardware like consumer-level depth sensors or even commercial motion capture (MoCap) systems to track the motions of users, devices, and objects [44]. It also often requires coding skills for specifying triggering events, desired effects, and their relationships in programming-based interfaces [4, 15, 44, 59]. This limits such prototyping techniques to indoor scenarios and prevents designers with little or no programming skill quickly prototype in situ in an ideation stage.

To avoid using additional hardware for motion tracking and low-level coding for specifying interactions, several techniques [26, 27] simulate large-scale environments using miniatures or virtual scenes, and allow designers to prototype interactions on their familiar schemes. While such techniques support a high degree of expressiveness, the lack of connection between created prototypes and real scenes makes it difficult to prototype physical interactions in real environments. On the other hand, Augmented Reality (AR), which overlays virtual contents on real environments, has been extensively adopted to design various AR prototyping and authoring tools [5, 13, 39, 65]. However, they focus on prototyping AR experiences or context-aware interactions from the first-person view, instead of full-body *pro-ges* interactions, and most of them rely on recorded videos or actions for authoring and editing [5, 39, 65].

The recent advances in computer vision algorithms have made it possible to use a single RGB camera like those on mobile phones or tablets to efficiently track the camera pose, estimate full-body human poses, and infer the geometry of 3D scenes. The accessibility and portability of these mobile devices potentially facilitate the prototyping process to be happened in situ conveniently. This motivated us to design a novel tool for prototyping full-body *pro-ges* interactions by using a single AR-enabled mobile device (Figure 1). To identify the challenges of current prototyping processes and inform the design of our new prototyping tool, we conducted depth interviews with five design experts in spatial interaction or smart environments. We distilled four design considerations for a desired mobile AR tool for prototyping *pro-ges* interactions. Based on the findings, we present *ProGesAR*, a novel mobile AR interface for prototyping *pro-ges* interactions in a real-world IoT enhanced environment, without the requirements of programming skills or extra hardware (Figure 1). For simplicity, we focus on a single human subject interacting with real/virtual devices statically situated in real environments.

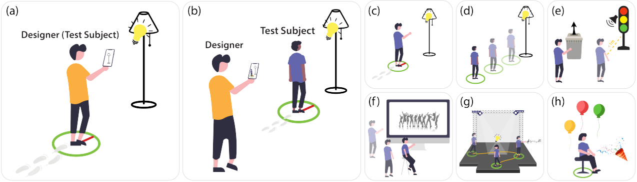


Figure 2: *ProGesAR* supports both first-person view testing (a) and third-person view testing (b). (a) In the first-person view testing, the designer acts as a test subject and performs interactions to trigger designed effects. The designer can also invite another user to test from the first-person view. (b) In the third-person view testing, the designer views another test subject performing designer-specified interactions to trigger effects by entering the AR view. (c)-(h) illustrate several intended uses of *ProGesAR* by various types of target users: (c) IoT designers: design and test proxemic and gestural relationships between potential users and IoT objects/devices; (d) spatial interaction designers: explore human factors (e.g., distance to smart objects) in smart space interaction design; (e) urban planners: validate the infrastructure setups and layouts in public scenes for special population; (f) game designers: test and demonstrate appropriate gestures for designing motion sensing games; (g) stage designers: depict dynamically changed states of actors and test the intended stage effects; (h) event planners: test and save the intended event effects for further setups.

In the authoring mode, our tool allows designers to specify desired spatial events and virtual effects to be triggered by certain events quickly from a designer's view, without requiring the presence of a test subject (i.e., a subject who will perform designer-specified events). Our event types are based on three key proxemic measurements: location, orientation, and distance [14]. We also consider full-body gestures, including sitting, standing, and pointing. For AR content authoring, our interface supports in-situ placement of both pre-defined and user-imported virtual assets. In the testing mode, our interface supports both first-person and third-person view testing of the prototyped interactions. Specifically, the designer might act as a test subject and perform the specified events in a first-person view (Figure 2(a)) to trigger the designed virtual effects. Alternatively, the designer might invite another person as a test subject and observe his/her interactions from a third-person perspective (Figure 2(b)). To evaluate our system, we conduct a preliminary study by inviting participants to perform freeform prototyping with *ProGesAR*. The early-stage findings show that users can easily and efficiently prototype design ideas with the help of *ProGesAR*.

ProGesAR targets the *prototype* phase in an entire user-centered design cycle [18]. Specifically, we focus on the stages between conceptual design and intermediate design. During these stages, designers have initial ideas in their minds and want to use low- or mediate-fidelity prototypes to explore and validate the overall interaction metaphor. The created prototypes are often considered as "private" to the project team for communication [18, 30]. Under the designing context of real-world IoT enhanced spaces, our target users are those who want to rapidly prototype *pro-ges* interactions for real-world environments, e.g., designing dynamic behaviors of smart objects responsive to users' proximity and gestures. They can be IoT designers, spatial interaction designers, urban planners, game designers, stage designers, etc. They care about the physical relationships between potential users and their surrounding environments and expect for demonstrating, communicating, and iterating the interactive behaviors easily and rapidly. They can use *ProGesAR* to quickly view and validate their ideas considering the physical spaces by themselves, demonstrate their conceptual designs to other people for iterations, and save the intended effects as references for further implementation and development. Besides professionals, *ProGesAR* is also useful for casual users for event planning (see Figure 6 and the accompanying video). Figure 2 (c)-(h) illustrates the intended uses of *ProGesAR* in various scenarios.

In summary, we highlight our main contributions to be the *ProGesAR* system, which features: (1) an expressive early-stage prototyping tool for *pro-ges* interactions in real scenes, (2) multiple-view perspectives for authoring and testing the prototypes, and (3) the demonstration of potential utilities from a series of applications made by both the study participants and the authors, and (4) the identification of challenges and needs of target users from the interviews, through which we ground the design of *ProGesAR* and wish to inspire others for better understanding and studying this topic further.

2 RELATED WORK

In the past decades, many prototyping tools (e.g., video prototyping [37], paper prototyping [50]) have been proposed in the HCI community. A full review of them is beyond the scope of this paper. In this section, we will mainly focus on the techniques for prototyping interactive systems, prototyping proxemic and gestural interactions, and AR prototyping. Table 1 shows the main differences between *ProGesAR* and the existing close-related prototyping tools in multiple perspectives.

2.1 Prototyping Tools for Interactive Systems

Designers are often skilled at designing the appearance of interfaces but find it difficult to prototype interactive systems, since implementing such systems requires programming skills to represent responsive or dynamic behaviors [46, 47]. Researchers have explored easy-coding or even coding-free prototyping tools [11, 17, 28, 34, 40, 49] for building interactive behaviors using statechart, sketching, map, and desktop applications. For example, Papier-Mâché [28] introduces a toolkit for building tangible interfaces by using computer vision, electronic tags, and barcodes. d.tools [17] presents a statechart-based prototyping model that extends the storyboard-driven design practice. Most of these works utilize declarative techniques to specify interactive behaviors efficiently. This motivates us also to design our event specification by setting up event triggers in a declarative paradigm. However, unlike these tools, which are mainly proposed to prototype early-stage design ideas for 2D applications, *ProGesAR* focuses on prototyping full-body interactions of real users in 3D space.

2.2 Prototyping Proxemic Interactions

The concept of *proxemics* was first proposed by Hall [16] and then extended to *proxemic interactions* by Ballendat et al. [3] to describe the spatial relationships between human users and their surrounding environments. Several works have explored interactive techniques for prototyping ubicomp space with proxemic interactions. According to the usage scenarios, these works can be largely divided into two categories: for real scenes and for virtual/miniature scenes. The Proximity Toolkit [44] and SoD-Toolkit [59] belong to the first category. The former presents an API library to facilitate the rapid exploration of proxemic interaction techniques for real subjects in physical environments. It relies on an outside-in motion tracking system to detect proxemic interactions, and a programming-based UI to define various triggering events and effects. It is thus not suitable for designers lacking the specific hardware or programming skills. SoD-Toolkit is a toolkit using sensors to provide spatial awareness for prototyping multi-device applications and proxemic interactions. Besides the requirement of additional hardware, these works require extra setups to detect users' proxemic interactions (Table 1). In contrast, our tool is more lightweight and flexible, though our current implementation does not fully support all the features in [44, 59] (e.g., those based on motion tracking of multiple users and dynamic objects; see more discussions in Section 7) .

To avoid using motion tracking hardware, SketchStudio [26] and miniStudio [27] resort to virtual or miniature scenes (Table 1). More specifically, SketchStudio presents a sketch-based prototyping tool for generating animated scenarios and allows users to directly define various types of user interactions by creating a node graph. Since it focuses on 2.5D virtual scenes, it is unclear how to apply SketchStudio for prototyping real user interactions in physical environments. Based on the technique of spatial AR, miniStudio introduces a projection-based prototyping tool applied to tangible miniatures. Although the authors of miniStudio briefly demonstrate its extended use for real humans and objects, each user is tracked through an AR marker and his/her proxemic or gestural interactions thus cannot be detected. In contrast, our tool is designed to prototype real users' spatial interactions with objects in physical environments.

2.3 Prototyping Gestural Interactions

Gestural interactions play an important role in the daily life of human-human communication. Researchers have explored body gestures [25, 36] and hand gestures [22, 29, 35, 56] in ubicomp space. Most of the existing works have focused on designing gestural interactions for different functions of smart devices in a smart home scene. Several techniques have been proposed to explore gesture prototyping. For example, Gesture Coder [41] and Gesture Studio [42] help developers better author and develop multi-touch gestures by demonstration. GestureAnalyzer [21] is a visual analytics system to identify and characterize the gesture patterns of the motion tracking data. These works have mainly focused on studying the general-purpose gestures themselves, instead of prototyping them for smart environments. GestureWiz [61] introduces a prototyping environment for recording, defining, and recognizing hand gestures in a Wizard-of-Oz (WoZ) manner. However, such a WoZ based approach increases the burden of users to perform corresponding

Table 1: The differences between *ProGesAR* and several closely related prototyping tools. The "Context" column refers to the design context, where "Real-world space" means that the designed results are for the interactions happening in the physical environments, while "AR space" means the designed results are to be interacted in AR scenes. "Declarative" in the "Authoring" column refers to setting up event triggers interactively. "Playback" in the "Testing" column refers to playing videos or animations.

	Context	Content	Setup	Authoring	Testing	View
ProGesAR	Real-world space	*pro-ges* interaction	Mobile phone AR	Declarative	Live	1st+3rd
Proximity Toolkit [44]	Real-world space	Proxemic interaction	Motion capture	Programming	Live	3rd
miniStudio [27]	Miniature space	Proxemic interaction	Projected miniature	Declarative	Live	3rd
SketchStudio [26]	Virtual space	Virtual interaction	Web+Cardboard VR	Node graph+Puppetry	Playback	3rd
GesturAR [64]	AR space	Hand interaction	AR-HMD	Demonstration	Live	1st
Pronto [39]	AR space	AR interaction	Mobile tablet AR	Demonstration	Playback	1st
Rapido [38]	AR space	AR interaction	Mobile tablet AR+PC	Demonstration	Live/Playback	1st

effects. Different from the WoZ based methods, *ProGesAR* provides a direct event triggering prototyping system. Recently, GesturAR [64] presents an authoring tool for users to create customized hand-gesture-based AR interactions. It enables the specification of hand-based events by embodied demonstration and live testing (Table 1) and focuses on authoring AR experiences for interacting with virtual assets using hand gestures. Different from GesturAR, our work aims at a novel tool for prototyping physical interactions between full-body gestures/proximity and physical objects.

2.4 AR Prototyping Tools

To facilitate prototyping in real environments, researchers have designed various AR prototyping systems [43, 53]. Many AR prototyping interfaces [10, 23, 24, 43, 58] require coding or scripting skills to author virtual contents. Some AR prototyping tools [48, 60, 63] simulate large-scale subjects using miniature or virtual space, but they focus on specific domains such as urban planning. PintAR [13] is a first-person-view prototyping tool that enables the dynamic effects of sketches based on a user's position and/or gaze direction using a tablet and an AR-HMD. Different from PintAR, *ProGesAR* supports dynamic effects triggered by a user's full-body interactions including more general types (i.e., location, orientation, distance, gesture) for multiple perspectives, and is more portable for rapid prototyping with a single AR-enabled mobile phone.

Researchers have explored authoring tools for context-aware [5, 65] and body-driven AR experiences [57]. Our work has several key differences from them. For the authoring contents and targets, we allow designers to prototype spatial relationships between human and physical world, while CAPturAR [65] and GhostAR [5] allow end users to author customized context-aware (e.g., object, location, time) or human-robot activities. For the authoring requirements, they rely on the recorded actions for authoring. In contrast, we let users specify authoring contents without the presence of a real user. The effects of Saquib et al. [57] are body-centered, while ours are object-centered and triggered by *pro-ges* interactions; their effects are in the video space while ours are in 3D AR scenes.

Recently, mobile AR based prototyping systems such as Pronto [39], Rapido [38], and ProtoAR [51] have been presented to provide portable interfaces. Pronto and Rapido are tablet-based AR video prototyping systems, which allow in-situ prototyping on top of a pre-recorded video. They deploy enacted demonstration with sketches and lightweight assets while we use a declarative event triggering workflow for authoring (Table 1). ProtoAR presents an AR prototyping tool for quickly transiting physical prototypes to digital ones. Commercial tools like Adobe Aero [20] and Apple RealityComposer [1] can also be used to author proximity-triggered AR interactions. The above AR prototyping tools focus on prototyping situated experiences of end users from their first-person view, instead of full-body *pro-ges* interactions from a third-person view (Table 1).

Our work is also related to Scenariot [19], a method for discovering and localizing smart things via a mobile AR interface, with the support of proximity-based control [32]. It aims to empower users to quickly and intuitively interact with connected smart things. Since its supported proximity-based control is pre-defined, strictly speaking, Scenariot is not a prototyping tool. In contrast, we provide an interface for prototyping *pro-ges* interactions. Since in our case virtual assets are interactively placed in-situ in the camera's coordinate system, the spatial relationships between real objects (associated with virtual assets) and the AR device are determined by this interactive authoring process instead of additional hardware.

3 INTERVIEW: IDENTIFYING DESIGN CHALLENGES

3.1 Interviewees

To figure out the challenges of prototyping *pro-ges* interactions in real-world IoT enhanced spaces, we conducted semi-structured depth interviews. The interviewees consisted of 5 experts (E1-E5, 3 males and 2 females), including 1 UX designer (E5), 3 professors (E1, E3, E4), and 1 PhD student (E2). We recruited them via our research network. They had 4-12 years (mean: 8 years, SD: 3.16 years) experiences in designing or researching spatial interactions or smart environments. They worked for or collaborated closely with the IoT industries for designing proximity- and gesture-triggered functions of smart products, devices, and services (e.g., smart mirrors in shopping malls, interactive smart home, etc.).

3.2 Procedure

All the interviews were performed remotely via video/audio conferencing tools, audio-recorded and transcribed for analysis. After collecting their background information, we asked them about their prototyping tools and workflows, common challenges and problems during prototyping, and expected features of a novel tool. They were allowed to freely talk about their experiences and opinions. Please refer to the supplementary materials for the detailed questions asked during the interviews. Each interview took about 30-40 minutes and each participant was compensated with a gift card.

3.3 Data Analysis

We exploited open coding following a Constructivist Grounded Theory method [7, 8] to collect and analyze the interview data. We discussed all the codes among the co-authors to identify emerging themes, which were then grouped into two themes (Sections 3.4.1 and 3.4.2). Within the second theme we further sub-categorized the codes into several sub-themes. We sorted these sub-themes by mentioned frequency and distilled five main issues (I1-I5), summarized as follows.

3.4 Findings

3.4.1 Typical Prototyping Tools and Workflows. All the participants usually sought for coding-free approaches. They typically drew 2D storyboards to depict continuous user actions and triggering effects in static format. Sometimes they further made videos or animations to demonstrate dynamic 2D user interactions, and delivered them to their colleagues and leaders for suggestions. To prototype 3D interactions, they relied on professional software (e.g., Blender, Unity 3D, Vuforia, etc.) to create 3D characters and effects in 3D scenarios or WoZ methods to record videos for 3D workflows and scenarios (E1, E2, and E5). For testing and experiencing the prototyped results, they built physical prototypes using hardware (e.g., sensors) or a WoZ technique [9] (E1, E2, E3, and E5). They aimed to communicate and evaluate ideas [30] by rapidly building prototypes regarding the overall interactive behaviors.

3.4.2 Common Prototyping Challenges and Needs. **I1: physicality.** All the participants mentioned that drawing 2D storyboards was easy but such storyboards only conveyed limited information. Specifically, they felt a huge gap between 2D storyboards and 3D real scenes. Since 2D storyboards lack the physicality and embodiment, they need to imagine how 2D storyboards would correspond to 3D scenarios. This conforms to the challenges of existing 2D tools identified in the previous works, i.e., "*difficult to design for the physical aspect of immersive experiences*" [2] and "*available tools lacking the 3rd dimension in the design space and the test environment*" [30]. E2 mentioned that when designing for a real scene, designers often had difficulties in perceiving faithful scales of real scenes, thus impeding conceiving proxemic and gestural interactions in a reasonable way. They expected an in-situ prototyping tool for real scenes.

I2: viewpoints. All the participants reported to use multiple perspectives in their prototyping workflows, but considered it inconvenient to switch views using different tools. They preferred to design from the third-person view, i.e., imaging and depicting how a user will interact with target scenes from a designer's viewpoint. E2 believed surrounding environments important in prototyping: "*Third-person designing allows me to consider the relationships between users and surrounding environments better"*. It resonates with E5's comments "*third-person designing lets me see more thing and thus design multiple user interactions quickly in a large view*". E3 and E4 considered first-person authoring and third-person demonstration unnecessary for specifying full-body *pro-ges* interactions, since the former focuses on a small view and the latter would take them more time.

On the other hand, the participants thought both first- and third-person views were necessary for testing and examining the prototyping results. This is mainly because 3D interactive behaviors are difficult to be fully designed and demonstrated from a single perspective, especially for multiple interactive behaviors in a single scene. It resonates with another identified challenge in [30], i.e., "*the difficulty to depict interactive and animated system behavior and storytelling*". They believed that the commonly used third-person testing offered a global expression on user interactions and workflow, and the first-person testing (usually in interactive prototypes, e.g., physical and WoZ prototypes) benefited to examine from a potential user's view. E4 considered first-person testing necessary because "*I can find the deviations from my concepts and ideas when I really try it*". However, there is no existing method supporting testing from both these two views (Table 1). The participants had to resort to a mix of various tools and methods [30] to obtain a comprehensive understanding of their design results. All of them suggested to support these two perspectives in a novel tool.

I3: re-test and re-design. Participants E3 and E5 mentioned that the physical prototypes were prone to misoperations. Sometimes test subjects do not intend to perform a certain interaction, but it triggers unexpected effects. In this case, designers often have to reset the events and effects, and test subjects need to wait for the next test. This makes the whole process time-consuming. They expected a tool supporting repetitive re-testing and re-designing with low time cost and high error tolerance.

I4: balance between cost and interactive prototypes. Another common concern of the participants is about the balance between prototyping cost and the interactivity of the prototypes. They thought creating non-interactive prototypes (e.g., based on sketches) was easy and low-cost, but not enough to demonstrate interactive behaviors including user actions [18]. Creating interactive prototypes (e.g., physical and animated prototypes) is expensive and time-consuming [30], and is thus not their first priority especially in the early-stage design. They expected a novel tool to help create interactive prototypes at a low cost.

I5: prototyping device. For a device suitable for prototyping *pro-ges* interactions in mobile AR, the participants believed that different mobile devices had their pros and cons. E1 and E4 mentioned "*AR-HMD is particularly useful for first-person testing since it's close to real using experiences*". They also pointed the drawbacks "*wearing an AR-HMD may limit and even mislead user interactions*" and "*it's not accessible to everyday users. Sometimes designers get ideas from daily lives and want to quickly prototype using a more accessible tool*". The participants thought mobile phones and tablets are more suitable for rapid prototyping in daily lives: E2 said "*mobile phones and tablets offer familiar operations and are more portable, [designers] hope to rely on their familiar schemes to do prototyping conveniently*".

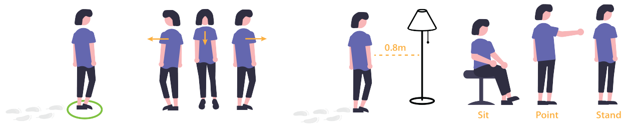
Based on the challenges and needs above, we distill four design considerations for a novel tool:

DC1: prototype for real-world scenes and real users in situ in mobile AR.

DC2: support first-person-view authoring, both first- and third-person view testing.

DC3: support re-test and re-design easily.

DC4: support to create interactive prototypes at a low cost.



(a) Location (b) Orientation (c) Distance (d) Gesture

Figure 3: Four types of spatial-related events supported by our system. (a) Location: occurs when a user comes into a certain spot, as indicated by the circle. (b) Orientation: occurs when a user faces a certain direction. (c) Distance: occurs when the distance between a user and a certain asset is below or above a certain threshold. (d) Gesture: occurs when a user performs a certain gesture.

4 SYSTEM DESIGN

Considering the familiar authoring preference [26], we choose to use a mobile phone to support rapid and portable prototyping. We design and develop our system for both AR content authoring and the detection of *pro-ges* interactions using an AR-enabled mobile phone (an iPhone XS in our implementation). This phone is used to observe and analyze *pro-ges* interactions of a human subject in a real environment from both a designer's view and a potential user's view.

4.1 Functional Components

Based on the design considerations distilled from the interviews, we design key functional components of our system.

In-situ AR content authoring. Our system allows for in-situ content authoring in mobile AR from the third-person view (DC1, DC2). The triggering events and virtual assets with dynamic effects can be placed and anchored at specific locations in AR space. In this way, designers can go to specific locations and view designed results (Figure 4(f)(g)).

According to the existing works [3, 19, 26, 31, 39], we derive and pre-define 25 types of common virtual assets in our system. These assets include visual elements (images, sketches, videos, 3D models, and animations) and audio elements (music). Due to the low-fidelity requirements [39], and the interview participants' interest more in the interactivity of *pro-ges* interactions instead of target effects (I4), we believe that the assets should just be used to simulate the final effects simply in the early stage. So we prepare most assets from simple 2D sketches, images and videos. We also allow designers to extend the asset set by importing project-specific asset elements from the device's album. We support seven dynamic effects (*Appear*, *Disappear*, *Shake*, *Play*, *Pause*, *Brighten*, and *Darken*) in our system.

Our system visualizes the horizontal and vertical planes automatically detected by Apple ARKit. The detected planes can be used to place the virtual assets in the AR space. For those planes that are difficult to be detected automatically, we provide a plane creation function, with which users specify two points on the ground plane to add a vertical plane passing through these two points. To indicate the positions of the located assets in a limited AR view, we set a small square indicator to each asset to represent its position. If an asset is in the camera view, its associated square is green and at the center of the asset, otherwise it will turn red and float on the border of the camera view [19] corresponding to its invisible 3D position (Figure 4(a)).

Spatial-relation events. Based on the proxemic measurements of [3], our system supports four spatial-related events: *Location*, *Orientation*, *Distance*, and *Gesture* (Figure 3). They include both proxemic and gestural interactions. Among them the former refer to the global proximity of human and the latter refer to a form of local body movements. Our current system does not support the *Identity* and *Motion* proxemic measurements in [3] for the following reasons. The *Identity* measurement is not needed, since our current implementation supports a single test subject only. It is possible to support the *Motion* measurement. But we find that the *Orientation* and *Distance* measurements together can already enable many similar features to the *Motion* measurement.

We categorize the events into two groups: *location-dependent* events (*Location*, *Location+Orientation*, and *Location+Gesture*) and *location-independent* events (*Distance*, *Orientation*, and *Gesture*). *Location-dependent* events are those occur at certain locations (e.g., sitting on a specific sofa in a room); *location-independent* events are location-variant (e.g., sitting anywhere in a room). They are displayed at different categories in our system. Multiple events within the same category can be added to one proxy for a compound event (e.g., *Location+Orientation+Gesture*).

Event specification. To support third-person view authoring and not an extra subject for demonstration, our system provides a declarative way to specify desired events based on *pro-ges* interactions via setting triggering proxies from the third-person view (DC2). The same interface works for both the location-dependent and location-independent events, which are shown as circle and square proxies, respectively (Figure 4(b)-(d)).

Event detection. Our system supports both the first- and third-person views (DC2) for examining and testing the prototyped interactions for real users in real-world scenes (DC1, DC4). All the events can be re-triggered and re-tested for multiple times (DC3) after pressing the *"Replay"* button.

4.2 Prototyping Walk-through

As illustrated in Figure 4, there are mainly four steps to prototype with *ProGesAR*: (1) assets and effects authoring, (2) triggering events creation, (3) effects and events mapping, (4) result testing and viewing. Our system allows designers to go back to any step for changes to achieve iterative prototyping. Below we introduce each step in more detail.

Step 1: assets and effects authoring. Once the user selects a desired asset from the menu, it will be first added to the scene in front of the camera. Then when the user drags the asset on the screen, our system automatically snaps the asset to the closest plane. The user can use a pan or pinch gesture to rotate or scale the asset, respectively. Next the user can select a desired effect to the added asset. Each time the user selects an effect, it will be previewed on the selected asset. The created effects can be viewed or deleted when pressing the "*Edit*" or "*Delete*" button.

Use case: Jessie wants to prototype a living-room scenario: when a subject approaches a sofa, a WiFi signal becomes available indicated by an appearing WiFi sign, and once the subject sits down and faces forward, the light will turn on indicated by a brightening bulb and the speaker will play the music (Figure 4). She is first in

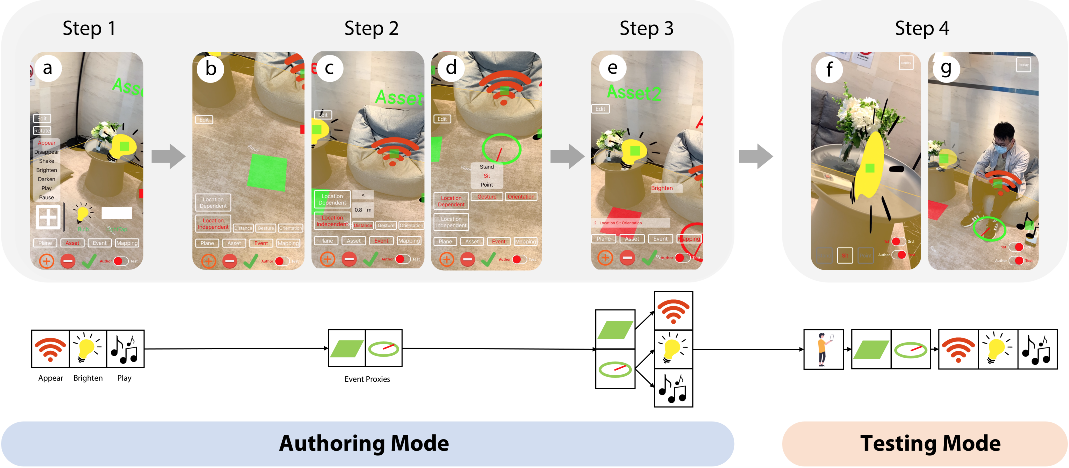


Figure 4: The workflow of *ProGesAR*. (a) Add a bulb asset to the scene and assign an "*Appear*" effect to it. (b) Create a square proxy for a location-independent event. (c) Create a *Distance* event represented by the square proxy. (d) Create a *Location+Orientation+Sit* event represented by a circle proxy. (e) Associate the "*Brighten*" effect of the bulb asset to the *Location+Orientation+Sit* event. (f) Test and view the result from the first-person view. (g) Test and view the result from the third-person view.

the "*Authoring*" mode (Figure 4(a)) and clicks the "*Asset*" button to open the asset set. She then swipes and selects a WiFi, a bulb, and a music asset from the asset set, and places them at desired locations. To assign effects to the created assets, Jessie selects each asset of interest and a desired effect from the effect menu (Figure 4(a), e.g., "*Appear*" for the WiFi asset, "*Brighten*" for the bulb asset, and "*Play*" for the music asset).

Step 2: triggering events creation. To add a location-dependent event, the user taps a specific location on the detected ground plane. A circle proxy (Figure 4(d)) will appear to visualize the newly added event. The user can leave the circle proxy to define a *Location* event only. To add other types of events to the same location, with the proxy selected in green, the user can select from the "*Gesture*" menu to define a *Location+Gesture* event (Figure 4(d)), or press the "*Orientation*" button and then adjust the red line (Figure 4(d)) around the location proxy to specify a *Location+Orientation* event.

Location-independent events can be added similarly. We visualize such an event as a square proxy (Figure 4(b)) representing a virtual location added to the ground plane in the AR scene. To define a *Distance* event, the user first taps to select a target asset, then specifies an inequality symbol, and finally inputs a specific number with the unit of the meter (Figure 4(c)). For adding a *Gesture* or *Orientation* event, it works similarly to the location-dependent events creation process. The user can view or delete the added event through the "*Edit*" button or "*Delete*" button, respectively.

Use case: After pressing on the "*Location Independent*" menu and "*Distance*" button, Jessie selects the WiFi asset and inputs "0.8" in the text box (Figure 4(b)(c)). Then an event describing the distance to the WiFi asset below 0.8 meters is created and visualized as a green square on the ground plane. Jessie then presses on the "*Location Dependent*" menu, and clicks on the ground plane to specify a spot. A circle representing this location (Figure 4(d)) is then added in the scene. She can drag and move the circle to change the location. Next she clicks the "*Gesture*" button and selects "*Sit*" to add a sit event. Afterwards she clicks the "*Orientation*" button, and drags to rotate the red line indicating the facing orientation. Finally she presses the green check button to confirm such a compounded event.

Step 3: effects and events mapping. We provide a direct mapping interface (Figure 4(e)) for defining the relationships between the created assets, effects, and events. From the AR scene, the user first taps to select a circle or square proxy, which may have multiple events associated with a physical or virtual location. The events related to this proxy will be listed for selection. After selecting an event, the user taps to select an asset still from the AR scene. The effects related to this selected asset will be displayed for selection. Once the two-step selection is completed, the user presses the green check button to save such a mapping. All the mappings can be viewed or removed through the "*Edit*" or "*Delete*" button, respectively.

Use case: Jessie first selects the square proxy and the displayed event "*<0.8m to Asset 1*", then selects the WiFi asset and the displayed effect "*Appear*", and finally clicks the green check button to confirm the mapping. Similarly, Jessie creates the mappings from the *Location+Orientation+Sit* event to both the "*Play*" effect of the music asset and the "*Brighten*" effect of the bulb asset (Figure 4(e)).

Step 4: result testing and viewing. The user can switch to the "*Testing*" mode (Figure 4(f)(g)) to test and view the prototyped result. In this mode, all the added assets will be hidden at the beginning and wait for being triggered. In the first-person view testing (Figure 4(f)), the user can hold and move the phone to trigger specified location, orientation, or distance events. For gesture events, our system provides the user with three buttons (Figure 4(f)) to manually activate them, since full-body gestures are difficult to recognize from the first-person viewpoint. For the third-person view testing (Figure 4(g)), when a test subject enters the camera view, our system automatically detects specified *pro-ges* interactions of the subject and displays or plays the corresponding dynamic effects being triggered. The user can view and test the prototyped results from different locations and views for multiple times.

Use case: Jessie switches to the "*Testing*" mode, which is set to the first-person testing by default (Figure 4(f)). Jessie holds the phone and walks approaching the WiFi asset. When the distance between the phone and the WiFi asset is below 0.8 meters, the WiFi asset will appear. Then when Jessie sits down on the sofa with the phone facing forward and presses the "*Sit*" button, the music is played and the bulb is brightened. To test in a third-person view (Figure 4(f)), Jessie asks another subject Fred to enter the AR view and perform interactions, similar to what Jessie did for the first-person view testing.

5 IMPLEMENTATION

Our system relies on Apple ARKit for real-time motion tracking. This is employed for the whole process of in-situ AR content authoring and *pro-ges* interaction detection. We believe our ideas can be easily implemented on iPad devices by redesigning our interfaces and Android-based mobile devices by exploiting Google ARCore.

5.1 AR Content Authoring

We employ real-time camera tracking and plane detection enabled by ARKit to implement a mobile AR interface for authoring triggering events and virtual assets. When the user places the virtual assets and effects in the AR space, our system records the position data and the pairs of assets and effects for further usage. Besides, after the user finishes defining the events, our system also saves the specified position, orientation, distance, and gesture data. Our system stores the specified mappings between the events and effects in a dictionary for efficient event detection.

5.2 Event Detection

To detect events from the first-person view, we first rely on motion tracking from ARKit to obtain the current 3D pose of the mobile phone. We then continuously calculate the horizontal distances between the obtained 3D device position and the positions of specified *Location* events (i.e., by projecting the device to the ground plane and calculating its distance to the location-dependent proxies), the horizontal angular differences between the 3D device orientation and the orientations of specified *Orientation* events, the horizontal distances between the 3D device position and the assets positions of specified *Distance* events to detect corresponding events respectively.

To detect events from the third-person view, we utilize human pose estimation and motion tracking from ARKit to obtain the 2D position and skeleton-based pose data. We back-project the 2D human position from the camera view into the automatically detected ground plane to get the 3D position of the human. We calculate the horizontal distances between a test subject and the assets in the ground plane. The calculated distances are used to detect the authored *Location* and *Distance* events. We train general SVM (Support Vector Machine) models to detect and classify the gestures and facing orientations of a test subject for the *Gesture* and *Orientation* events. The 5-fold cross validations show that the trained classifiers are highly accurate (over 99%). We implemented the classifications with Python on a PC server via a wireless network with HTTP requests. Please refer to our supplementary materials for more details.

6 EVALUATION

To evaluate the effectiveness and usability of *ProGesAR*, we have conducted a preliminary usage study [33] by means of an open-ended study.

6.1 Participants

We invited 8 participants (5 males and 3 females, aged from 23 to 33), including 1 UX designer (P1), 5 researchers in design (P2-6), and 2 casual users with interest in design (P7-8). They had the background of smart environment design (P1, P2, P4, P6), UI/UX design (P1, P2, P5), product/industry design (P3), environment art design (P4), and AR development (P7, P8). P1-P6 had project experiences on designing spatial interactions. All the participants had prototyping experiences except P7. Although some of them had programming skills (low: P1 and P3, medium: P6-8), they usually utilized coding-free approaches for prototyping. They mentioned 2D sketching, paper prototyping, physical prototyping (e.g., build sensors and circuits), 2D software (e.g., Mockingbot), and 3D software (e.g., Unity 3D, Blender, Maya) for prototyping in their work. All of them were daily mobile phone users, and had AR experiences on mobile phones or HoloLens. We will give detailed comments and comparison in Section 6.4. For the evaluation, we used an iPhone XS with iOS 13.7 running *ProGesAR*, which was wirelessly connected to the PC server for gesture and orientation classification. The whole process was screen-recorded for further time, interaction, and result analysis. Each participant was required to create at least one result. The participants conducted the study freely without the requirement of time duration, result diversity, or result quality.

6.2 Training Session

We started the study with a training session, which taught the participants how to prototype a representative scenario from [3] (Figure 5(a)) using *ProGesAR*. We showed a pre-recorded video of a target prototyping design (created by one of our authors) to them, and introduced the interface, workflow, and menu functions of *ProGesAR* to them. Then we took them to a target scene and they utilized our app to reproduce the target design. After they finished creating all the elements, we first let them test the results from the first-person view by themselves, and then asked a test subject to enter the camera view to test their designed results from the third-person view. We measured the time from the recorded videos by calculating the accumulated duration of each mode/view. All the participants successfully used our system to prototype the target design in 256.75s on average, including 177.25s for authoring, 40.00s for the first-person view testing, and 39.50 for the third-person view testing.

6.3 Formal Session

After the training session, we asked every participant to prototype free-form interactions and scenarios using *ProGesAR*. We gave them one day in advance to think about possible scenarios, including their associated interactions and effects in real environments. Besides

(a) (b) (c)

Figure 5: (a) The target scene in the training session: when a test subject approaches the wall, a video appears; when sitting down on the chair, the video plays; when pointing out, the video stops playing. (b) User prototyping process in the open-ended study. (c) Mean scores of SUS in a 5-point scale. Please refer to the supplementary materials for the detailed SUS questions.

the existing assets in our system, they were also allowed to utilize their own prepared assets. During the study, they used *ProGesAR* to turn their ideas into AR prototypes (Figure 5(b)). At the end of the study, we let them fill in a questionnaire of System Usability Scale (SUS) in a 5-point scale (1 = strongly disagree to 5 = strongly agree), and asked them about the user experiences with *ProGesAR*.

Results. The participants created 14 prototyping results in total. Figure 6 shows the representative results designed by them. The participants used *ProGesAR* to quickly validate their design ideas mainly for the dynamic interactive behaviors and see whether and how their ideas work in real-world 3D spaces. The results show various potential applications such as smart household appliances (Figure 6(b)(i)), smart home/facility reminding (Figure 6(d)(f)(j)), inductive public facilities (Figure 6(a)(c)(g)), cross-device sharing in office (Figure 6(e)), and event plannings (Figure 6(h)). These applications were designed for diverse groups of audiences (e.g., public/home users including physically impaired users, office staffs, etc.) to enhance their living convenience and working efficiency. Most of the results demonstrate close connections between the created prototypes and real scenes. For example, Figure 6(b) and (c) depict two use cases showing the novel interaction ways for the existing functions of electronic appliances and public facilities. Figure 6(b), (d), (g), and (i) target the additional functions and attachments on the existing devices and furniture. Figure 6(a), (e), (f), (h), and (j) show the designs of displaying the desired contents on the existing physical objects (e.g., walls, desks, glass). The application results show diverse intended usages of designers (Table 2), e.g., validating the setups of public facilities, demonstrating the intended design for appliance functions, exploring human factors in target scenario design, etc. These prototyped designs reflect the rich expressiveness and a wide design space of *ProGesAR* and its great potential for diverse applications.

In each result, from 1 to 4 *pro-ges* interactions were designed to trigger the associated effects of the added assets. *Location* was the most frequently involved event. Except for the *Distance* event, many other types of events were created together with the *Location* event (location-dependent), since the participants preferred to ensure the interactions to happen at a precise location to avoid unintended triggering (e.g., Figure 6(a) and (c)). *Distance* events were usually defined with coupled effects (e.g., in Figure 6(f), "< 0.5m" triggers sign appearing and "> 0.5m" triggers sign disappearing). This encourages us to improve the UI by providing a coupled event creation mechanism for *Distance* events.

The average time for creating one prototype was 157.5s, including 106.85s for authoring, 27.14s for the first-person view testing, and 23.51s for the third-person view testing. Since most results involve one scene with easy-to-perform interactions, *ProGesAR* helped the participants tested the results quickly. Figure 5(c) shows the subjective rating of the SUS questionnaire. The scores indicates that the participants appreciated the functions provided by *ProGesAR*, and agreed that *ProGesAR* could help them design, demonstrate, and test the early-stage or planning ideas accurately, effectively, and expressively. They were satisfied with their prototyping results in diverse aspects. For example, P1 considered her results as "interactive storyboards" (Figure 6(b)) to fully demonstrate using scenario consisting of multiple user actions. P2 thought her desired office scene (Figure 6(e)) was precisely designed and tested with the facing orientation as triggers, since the re-design and re-test cycle let her explore the appropriate human factors (e.g, facing orientation) efficiently.

6.4 Findings

By observing the participants' prototyping processes and discussing with them, we extract several findings as follows.

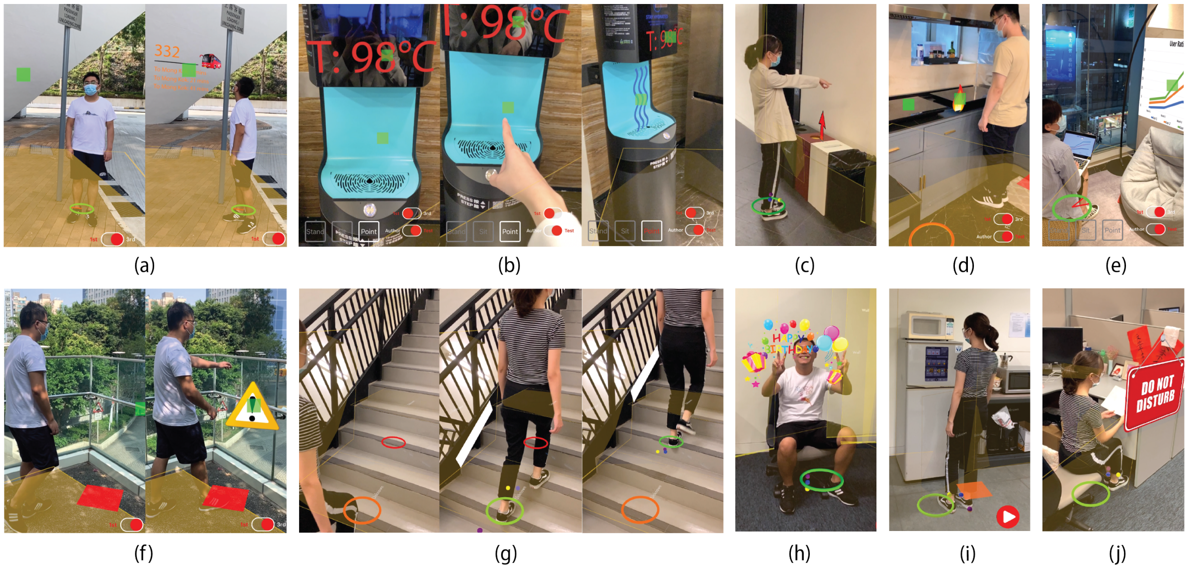
6.4.1 *Qualitative behavioral insights.*

All of the participants learned how to use *ProGesAR* very quickly and smoothly, and especially got used to specify interactions using our event-triggering workflow. Some participants (P3-7) thought up design ideas first and then found proper places to realize the ideas. The others (P1, P2, P8) found the scenes first and thought about potential ideas based on the scenes. Before starting prototyping, most of the participants looked around the environments and verbally described the intended setups of virtual effects. When they set an asset/event trigger, they often walked around to view and adjust it from different perspectives. They watched the results from different views for multiple times to check whether their ideas fit the current scenes well. These observations show that *ProGesAR* is a powerful tool for users to rapidly apply their ideas to physical scenarios.

6.4.2 *Qualitative attitudinal insights.*

From the participants' comments, we summarized their attitudes towards *ProGesAR* by categorizing them into strengths and limitations. We will first discuss the positive feedback, followed by the suggestions for possible improvements of our system. We found that the advantages of *ProGesAR* mentioned by the participants resonate with our design considerations in Section 3.4, and *ProGesAR* reduces the solution viscosity in several aspects, fitting the Olsen's heuristics [52].

Low cost and low skill barrier for creating interactive prototypes. Most of the participants (P1, P2, P3, P4, P5, P8) pointed out that *ProGesAR* saved a lot of time, labor, and money costs for building interactive prototypes. Such prototypes are important to enhance interdisciplinary communication in early-stage design [30]. Besides validating design ideas by themselves, the participants pointed out that the prototyped results could be shown among other people in/outside their teams to facilitate the creation of more effective designs. For example, they could show and invite development teams to test using *ProGesAR* in order to check the



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 6: A gallery of open-ended prototyping results produced by the eight users of our system. Please refer to the accompanying video for the dynamic effects, and additional results. Table 2 gives the detailed description of each result.

Table 2: Descriptions of the open-ended prototyping results (see Figure 6). The "L", "O", "D", and "G" in the "Event Type" column refer to location, orientation, distance, and gesture, respectively.

ID	Event	Effect	Event Type	Intended Usage	Application	Target Audience
(a)	Face to the bus stop board	The bus information is displayed	L+O	Validate the setup of information board in the bus stop	Inductive information notification	Public users
(b)	Come to a touchless water dispenser	The water temperature appears on it	L	Demonstrate *pro-ges* interactions for water dispenser functions	Smart household appliances	Public/home users
	Point at the dispenser	Water is delivered	L+G			
(c)	Come to and point at a touchless trash bin	An arrow shakes to show the opening action of the bin	L+G	Demonstrate intended gestures for trash bin functions	Inductive public facilities	Public users
(d)	Approach a cook	It displays a shaking fire sign to indicate its untouchable status due to high temperature	D	Explore human factors in smart kitchen design	Smart appliance reminding	Cooks, home users
(e)	Rotate the head and face to the wall	A slide from the laptop is projected on the wall	L+O	Explore human factors in smart office design	Cross-device sharing	Office staff
(f)	Approach the glass handrail	A danger sign is displayed on the glass to indicate the danger of moving forward	D	Demonstrate *pro-ges* interactions for handrail function	Smart facility reminding	Public users
(g)	Go upstairs	The lights on the stairs turn on and follow the subject	L	Depict users' changed states and test the intended light effects	Inductive public facilities	Public users
(h)	Sit down on a chair	A shaking birthday surprise is projected on the wall behind him	L+G	Test and save the birthday effects for further setups	Event planning	Friends
(i)	Come to a spot near the fridge	The food stored in the fridge is displayed on the door	L	Demonstrate *pro-ges* interactions for fridge functions	Smart household appliance	Home users
	Leave from the fridge	The information disappears	D			
(j)	Come to a chair and sit down	A "do not disturb" sign is displayed next to the desk	L+G	Test and save the intended effects for further setups	Smart office reminding	Office staff, Library students

feasibility of their design ideas. It can also be applied to interaction design classes as part of facilitation tools for designers-in-training. P1 and P3 both mentioned that with *ProGesAR*, they "*don't need to go to the market and buy physical components, materials, and sensors to test their ideas in an interactive manner*". Besides, our system also fits the *Expressive Match* of Olsen's heuristics [52] well due to its low skill barrier. P2 and P3 believed this coding-free tool reduced much learning cost with low barrier of entry. Specifically, P3 said "*I don't need a computer to create interactive models for real scenes, and code for input and output*".

Multiple-view authoring and testing. *ProGesAR* also reduces design viscosity [52] by providing a flexible solution for *Expressive Leverage* and *Flexibility*. All the participants appreciated the integration of multi-view perspectives in a single interface, which reduces the choices and repetitions needed in a mix of tools. P6 said "*two-view authoring and testing* compensates with each other". The third-person view authoring allowed them to set effects and events "*from a wide view*" (P2), and "*considering surrounding environments*" (P1), and "*standing in the whole scene to do the prototyping*" (P3). It helped them understand the relationship between potential users and the environments better. P3 elaborated further: "*when I design the products in smart scenes, not only do I design individual products, but also I design the whole environment including the relative location and orientation of human. I think this tool helps me consider this point well.*"

The participants also considered the first- and third-person view testing very useful to help them "*find more problems in their design* *for further improvement*" (P6). P4 and P1 both pointed out the problems of the only support of the third-person view in traditional storyboard-driven methods. P4 said "*designers usually are used to prototype from the third-person view, which is also supported by this system. But he/she may ignore the first-person view feelings, which can usually be obtained only at the very last stage of the design – when it comes to high-fidelity design products*". P1 agreed that it enabled her to get real experiences by testing the prototyped results from the first-person view first, and to find more problems by viewing the results from an observer's perspective due to its wide view. P6 liked the third-person view testing for demonstration since he would like to record the videos of the real user testing in *ProGesAR* from different views and show them to his peers for comments.

Immersive experience for real-world scenes. Our system is also proved by the participants, to fit the *Expressive Match* of Olsen's heuristics [52] well, since the intuitive in-situ authoring and testing provide a closer match to the real situations. Most of the participants (P2, P3, P5, P6, P8) mentioned that *ProGesAR* provided them with an efficient way to prototype real user interactions in real scenes, whose embodiment and physicality are difficult to obtain in the testing stage using existing approaches [2, 30]. P2 particularly loved the tool due to its support of faithful physical features (e.g., interaction scale and area). She said "*size, scale, position [of user interaction] in the space are reasonable [using ProGesAR]. Prototyping by using traditional software leads to a larger difference to the actual scenes in scale and position*". P8 believed that *ProGesAR* was good at conveying the precise interactions between users and objects.

Ease-of-use, controllability, expressiveness, and iterative designing. *ProGesAR* also encouraged the participants to prototype easily and expressively with good controllability. P2 appreciated the quick and dynamic storytelling of *ProGesAR*, which is a challenge in designing and building AR applications [2]. She further explained that "*unlike 2D storyboards, which usually have several static pictures, I can prototype in a dynamic process [using ProGesAR]: I can walk there, sit down, and trigger effects. I can see a set of actions. I can design inputs and outputs easily, and see them happening clearly. I can control the time by myself instead of making an animation by inserting a lot of keyframes and designing the static timing for triggering*". P7 and P8 commented the convenient mapping from the event and effect list and a clear workflow for prototyping their customized ideas in daily lives. P3 thought that *ProGesAR* supports to revise the ideas easily and test them efficiently, and thus could provide high flexibility and adaptability for collaboratively refining design ideas [30].

Limited screen size and FoV ((Field of View). Four participants (P1, P2, P3, P7) mentioned that the main limitation of *ProGesAR* was the limited screen size and FoV. It becomes an obvious problem in the first-view testing for multiple continuous scenarios from some participants, since the triggering event proxies are on the ground planes while the assets are usually in the air. The participants often needed to see the event proxies first to remind them what interactions they would perform to trigger the effects. When they changed the camera view from the event proxies to the target assets, the effects sometimes had already been playing. To see the entire effects, they need to re-trigger the events. This might be addressed by adding a replay feature for the effects in the future version of our tool. P2 and P8 said that limited screen sizes and FoV are common problems with mobile devices, and these issues can be relieved by using AR-HMD devices. But there is a trade-off between the portability and accessibility, and immersive interaction space and large FoV. As discussed in the interviews (Section 3), we would like to support rapid and portable prototyping in users' daily lives, and thus AR-HMD is out of our consideration. Compared with video-based prototyping tools [39], which allow users to examine results from a specific view in the timeline of videos, and indirect fully-virtual representations in virtual coordinate systems [1], our interface provides users with more freedom to manipulate the viewing angles, but takes more efforts to position a specific focus view. To address the identified issues, one possible solution is to play triggered effects only when they appear in the camera view. We also believe that such a limitation in the first-person view can be compensated by the third-person view testing to some extent.

Fixed gesture set. Participants P3 and P6 mentioned the limited gestures supported by *ProGesAR*. P3 suggested adding more elaborate gestures (e.g., press on the chair) for diverse prototyping. On the one hand, he thought that our current design already supported a general framework for prototyping and testing with the three common body gestures. With these gestures he already could rototype various *pro-ges* interactions needed in smart environments. On the other hand, for further interactions, it is better to allow the use of elaborate hand gestures. P6 also mentioned this point and suggested to allow users to define gestures needed by themselves [64], and import user-defined gestures to the system. We will consider supporting more types of gestures (e.g., swiping) in the future.

Touch-based interaction for gesture triggering in the first-person view. In our current design, users need to press buttons on the screen to trigger three gesture events in the first-person view, as also pointed by P6. He said "*although it's quick to press the buttons to trigger, it's different from I really perform those gestures to trigger the events*". To address this issue, we need to perform gesture recognition, without seeing the whole body of the user. This might be possible by training a classifier based on the IMU data of the mobile device and the image-based data from the camera view.

7 DISCUSSIONS AND FUTURE WORK

Dependence on real environments. Ideally, a prototyping tool should be able to combine different modalities (e.g., simulating virtual/miniature scenes, in-situ authoring, remote authoring) to provide users with a one-stop solution. Since our target using scenario is for real-user interaction with real-world environments, *ProGesAR* is applicable for prototyping through the interactions with physical objects or devices. With our current implementation, it is difficult for users to design a totally virtual scenario with freeform setups in an initial planning phase or re-use the prototyped results in different places. So *ProGesAR* is not a holistic prototyping solution. Compared with the prototyping tools for virtual [26] and miniature scenes [27], *ProGesAR* requires users to find a proper physical scene for prototyping. It might restrict the design space of designers for more freeform scenarios. However, virtual or miniature based prototyping and real-scene based prototyping [19, 39, 51] have their own pros and cons. The first type of tools provides designers with more freedom to think about design ideas, and designers usually do not need a proper real space. But these tools lack the capabilities of bringing test subjects into immersive real-world environments to understand the needs behind the physical world. Besides, users can only view the prototyping process and results from limited views using these tools. In contrast, real-environment based prototyping tools rely on physical objects and devices, but they can provide users with real-user interaction and testing. Users can obtain the direct and intuitive feedback from the real scenes, and then refine their conceptual design ideas quickly. To provide users with more freedom, we also involve the virtual objects and devices (e.g., fan) in our pre-built asset set, which can be employed by users for further design on them. To some extent, it balances the gap between the two types of prototyping approaches. We might further reduce the dependence on current scenes by incorporating the idea of remote prototyping [66]. To make the prototypes be migrated to different physical contexts, we can also pair the specified events to the anchor points, planes, or objects in the current scene. Then when designers move to another physical scene, the paired events can be re-anchored to newly detected anchors. This would enable testing in different environments by reusing the prototyped results. The above ideas would help move *ProGesAR* in the direction of a holistic prototyping system in the future.

Event authoring and testing styles. Our system provides a declarative event authoring technique by interactively setting up event triggers. It is applicable for quickly specifying discrete events (e.g., coming to a certain location) but less convenient for continuous forms of interactions (e.g., the movement dimension of proxemic interactions). For example, when a user wants to prototype an interaction of moving forward continuously, he/she must set up several location triggers to approximate the continuous interaction behavior (Figure 6(g)). Such continuous events can be better specified by deploying embodied or enact demonstration [38, 64] since users can perform such continuous interactions directly. But if the embodied and enactment techniques are utilized in our system, it will require an extra subject to perform full-body interaction, thus increasing the design cost during the authoring phase. In the future, we will explore keyframe-based authoring techniques (i.e., specifying the start and end states interactively and inferring the medium states automatically) to improve the authoring of continuous interactions. In addition, our system enables a live-testing mode. Compared with video/animation playback testing [26, 39], live testing allows users to interactively control the testing pace, thus helping simulate the real usage scenarios better.

UI design and scalability. The usability of our tool can be further enhanced by introducing additional visualization techniques, e.g., for adding a virtual line to visualize the distance between a test subject and a target asset, adding a virtual agent with the selected gestures, re-organizing the UI layouts, etc. Besides, in our current implementation, 2D assets are attached to 2D plane geometries, which have the fixed orientations during the prototyping process. We are interested in implementing 2D assets as billboards so that they will always face the viewer. Besides, our system can scale to more complex scenarios while maintaining the current UI framework and workflow (e.g., integrating object-related events based on object detection). Various effects in commercial tools and customized effects (e.g., motion captures of SpatialProto [12], animations of Pronto [39]) can be integrated. More types of assets (e.g., elaborate 2D assets and 3D models) can also be added to increase the prototype resolution. In addition, the created prototypes through *ProGesAR* can be exported for developing final applications further. The created event proxies can be re-used as indicators for developing and installing sensors for detecting humans' movements. The interactive prototypes can also be employed to test and determine the detection thresholds. In the future, the addition of a secondary device [38, 51] to the authoring procedure can help enhance the collaborative prototyping experiences and the compatibility with designers' familiar prototyping schemes. Involving multiple networked devices can be employed to handle scenarios involving collaborative interactions. In the final product envision, the smartphone can also work as sensors of IoT devices along with real users to help detect the *pro-ges* interactions.

Trends and design space of design applications. In most of the study results, three dimensions – attention, accuracy, and naturalness of the proximity and gestures were considered by the participants for the IoT space design. For example, the body orientation, pointing gesture, and distance to objects reflect the attention focus of users, the frequently co-created location event enables the interactions to happen without mis-triggering, and the sitting gesture provides a comfortable interaction state. In such cases, the participants preferred to design compound events, to better improve the interactive experiences with real-world environments of target audience. Compared to actual ubicomp applications, there remain a lot of complex factors to be considered. Nevertheless, *ProGesAR* opens up a new path that allows designers to facilitate the implementation of natural *pro-ges* interactions in IoT spaces.

Support of more interaction types and asset effects. Our current implementation has focused on the case of a single test subject. The state-of-the-art human pose estimation algorithms (e.g., OpenPose [6]) allow robust tracking of multiple users in a scene. It is thus possible to enhance *ProGesAR* to support the prototyping of multi-user interactions. To this end, it is necessary to add the *Identity* measurement [3]. The recent development of real-time object detection techniques (e.g., YOLO [55]) makes it feasible to model a richer set of proxemic interactions between users and digital devices, devices and devices, and non-digital physical objects and users/devices [3]. We can also include the context of creating embedded AR visualizations by tracking physical objects in the scene of [62]. Besides, our system currently supports only seven types of effects. Although they are enough to create expressive prototyped results, we are interested in including more types of effects (e.g., rotating and moving effects in PintAR [13], enacting the movement of objects with the mobile device in Pronto [39]) in the future.

Dependence on ARKit's environment understanding ability. Our current implementation is highly dependent on the environment understanding ability of ARKit, which, however, sometimes does not work well. We found that when there are sufficient visual features in AR view, assets can be located to the detected feature points easily even without creating any planes. Meanwhile, since the dense feature points in a small area may offer too many locating candidates, it requires users to change the view to locate an asset to a desired position precisely. It would be interesting to explore and utilize other techniques (e.g., create freeform plane proxies using device orientation [31]) to help locate assets with less effort when the environment understanding of ARKit does not work well. Besides, our system cannot work well for scenes on multiple floors due to the motion tracking limitation or large distances between test subjects and target effects due to the limited field of view.

8 CONCLUSION

In this paper, we have presented *ProGesAR*, a mobile AR tool for prototyping *pro-ges* interactions of a subject in a real-world IoT enhanced environment. We have conducted depth interviews with five experts to identify the challenges for prototyping such interactions and found five issues regarding the **physicality**, **viewpoints**, **re-test and re-design**, **balance between cost and interactivity**, and **prototyping device**. Based on the identified issues, we distilled four considerations for designing *ProGesAR* and informing further studies on ubiquitous design of IoT spaces. With *ProGesAR*, a user can follow a declarative event-triggering technique to specify spatial triggering events and effects to be triggered quickly from a designer's view. In the testing mode, the user can view and test by himself/herself from a potential test subject's view, or invite and view another subject to perform the interactions from a designer's perspective. The usability and expressiveness of *ProGesAR* has been demonstrated by a preliminary usage study with diverse potential applications. We believe that our work has just taken the first step towards effective tools for prototyping *pro-ges* interactions in mobile AR. Our system can be improved in various aspects and extended to more general and holistic solutions for IoT enhanced space design.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments, and the interview and user study participants for their time. This work was supported by gifts from Adobe and grants from the City University of Hong Kong (Project No. 7005729, 9667234, 7005590, 9229094), and the Centre for Applied Computing and Interactive Media (ACIM) of the School of Creative Media, CityU.

REFERENCES

[1] Apple. 2019. Reality Composer. https://apps.apple.com/us/app/reality-composer/id1462358802

[2] Narges Ashtari, Andrea Bunt, Joanna McGrenere, Michael Nebeling, and Parmit K Chilana. 2020. Creating augmented and virtual reality applications: Current practices, challenges, and opportunities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[3] Till Ballendat, Nicolai Marquardt, and Saul Greenberg. 2010. Proxemic interaction: designing for a proximity and orientation-aware environment. In *ACM International Conference on Interactive Tabletops and Surfaces*. 121–130.

[4] Andrea Bellucci, Aneesh P Tarun, Ahmed Sabbir Arif, and Ali Mazalek. 2016. Developing Responsive and Interactive Environments with the ROSS Toolkit. In *Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*. 782–785.

[5] Yuanzhi Cao, Tianyi Wang, Xun Qian, Pawan S Rao, Manav Wadhawan, Ke Huo, and Karthik Ramani. 2019. GhostAR: A time-space editor for embodied authoring of human-robot collaborative task with augmented reality. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 521–534.

[6] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2018. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *arXiv preprint arXiv:1812.08008* (2018).

[7] Kathy Charmaz. 2008. Constructionism and the grounded theory method. *Handbook of constructionist research* 1, 1 (2008), 397–412.

[8] Kathy Charmaz. 2014. *Constructing grounded theory*. sage.

[9] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. 1993. Wizard of Oz studies—why and how. *Knowledge-based systems* 6, 4 (1993), 258–266.

[10] Mark Fiala. 2005. ARTag, a fiducial marker system using digital techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 2. IEEE, 590–596.

[11] James Fogarty, Jodi Forlizzi, and Scott E Hudson. 2002. Specifying behavior and semantic meaning in an unmodified layered drawing package. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*. 61–70.

[12] Maxime Garcia, Rémi Ronfard, and Marie-Paule Cani. 2019. Spatial Motion Doodles: Sketching Animation in VR Using Hand Gestures and Laban Motion Analysis. In *Motion, Interaction and Games*. 1–10.

[13] Danilo Gasques, Janet G Johnson, Tommy Sharkey, and Nadir Weibel. 2019. What you sketch is what you get: Quick and easy augmented reality prototyping with pintar. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–6.

[14] Saul Greenberg, Nicolai Marquardt, Till Ballendat, Rob Diaz-Marino, and Miaosen Wang. 2011. Proxemic interactions: the new ubicomp? *interactions* 18, 1 (2011), 42–50.

[15] Sinem Güven and Steven Feiner. 2003. Authoring 3D hypermedia for wearable augmented and virtual reality. In *Proceedings of IEEE International Symposium on Wearable Computers (ISWC'03)*. 21–23.

[16] Edward Twitchell Hall. 1966. *The hidden dimension*. Vol. 609. Garden City, NY: Doubleday.

[17] Björn Hartmann, Scott R Klemmer, Michael Bernstein, Leith Abdulla, Brandon Burr, Avi Robinson-Mosher, and Jennifer Gee. 2006. Reflective physical prototyping through integrated design, test, and analysis. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*. 299–308.

[18] Rex Hartson and Pardha Pyla. 2012. *The UX Book: Process and guidelines for ensuring a quality user experience*. Elsevier.

[19] Ke Huo, Yuanzhi Cao, Sang Ho Yoon, Zhuangying Xu, Guiming Chen, and Karthik Ramani. 2018. Scenariot: spatially mapping smart things within augmented reality scenes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.

[20] Adobe Inc. 2019. Adobe Aero. https://apps.apple.com/us/app/adobe-aero/id1401748913

[21] Sujin Jang, Niklas Elmqvist, and Karthik Ramani. 2014. GestureAnalyzer: visual analytics for pattern analysis of mid-air hand gestures. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*. 30–39.

[22] Runchang Kang, Anhong Guo, Gierad Laput, Yang Li, and Xiang'Anthony' Chen. 2019. Minuet: Multimodal interaction with an Internet of Things. In *Symposium on Spatial User Interaction*. 1–10.

[23] Hirokazu Kato and Mark Billinghurst. 1999. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*. IEEE, 85–94.

[24] Annie Kelly, R Benjamin Shapiro, Jonathan de Halleux, and Thomas Ball. 2018. ARcadia: A rapid prototyping platform for real-time tangible interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–8.

[25] Daehwan Kim and Daijin Kim. 2006. An intelligent smart home control using body gestures. In *2006 International Conference on Hybrid Information Technology*, Vol. 2. IEEE, 439–446.

[26] Han-Jong Kim, Chang Min Kim, and Tek-Jin Nam. 2018. Sketchstudio: Experience prototyping with 2.5-dimensional animated design scenarios. In *Proceedings of the 2018 Designing Interactive Systems Conference*. 831–843.

[27] Han-Jong Kim, Ju-Whan Kim, and Tek-Jin Nam. 2016. miniStudio: Designers' Tool for Prototyping Ubicomp Space with Interactive Miniature. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 213–224.

[28] Scott R Klemmer, Jack Li, James Lin, and James A Landay. 2004. Papier-Mache: toolkit support for tangible input. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 399–406.

[29] Barry Kollee, Sven Kratz, and Anthony Dunnigan. 2014. Exploring gestural interaction in smart spaces using head mounted devices with ego-centric sensing. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*. 40–49.

[30] Veronika Krauß, Alexander Boden, Leif Oppermann, and René Reiners. 2021. Current practices, challenges, and design implications for collaborative AR/VR application development. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.

[31] Kin Chung Kwan and Hongbo Fu. 2019. Mobi3DSketch: 3D Sketching in Mobile AR. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM.

[32] David Ledo, Saul Greenberg, Nicolai Marquardt, and Sebastian Boring. 2015. 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*. 187–198.

[33] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. 2018. Evaluation strategies for HCI toolkit research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–17.

[34] David Ledo, Jo Vermeulen, Sheelagh Carpendale, Saul Greenberg, Lora Oehlberg, and Sebastian Boring. 2019. Astral: Prototyping Mobile and Smart Object Interactive Behaviours Using Familiar Applications. In *Proceedings of the 2019 on Designing Interactive Systems Conference.* 711–724.

[35] Sang-Su Lee, Jeonghun Chae, Hyunjeong Kim, Youn-kyung Lim, and Kun-pyo Lee. 2013. Towards more natural digital content manipulation via user freehand gestural interaction in a living room. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing.* 617–626.

[36] Wei-Po Lee, Che Kaoli, and Jhih-Yuan Huang. 2014. A smart TV system with body-gesture control, tag-based rating and context-aware recommendation. *Knowledge-Based Systems* 56 (2014), 167–178.

[37] Germán Leiva and Michel Beaudouin-Lafon. 2018. Montage: A video prototyping system to reduce re-shooting and increase re-usability. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology.* 675–682.

[38] Germán Leiva, Jens Emil Grønbæk, Clemens Nylandsted Klokmose, Cuong Nguyen, Rubaiat Habib Kazi, and Paul Asente. 2021. Rapido: Prototyping Interactive AR Experiences through Programming by Demonstration. In *The 34th Annual ACM Symposium on User Interface Software and Technology.* 626–637.

[39] Germán Leiva, Cuong Nguyen, Rubaiat Habib Kazi, and Paul Asente. 2020. Pronto: Rapid Augmented Reality Video Prototyping Using Sketches and Enaction. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems.* 1–13.

[40] Yang Li, Jason I Hong, and James A Landay. 2004. Topiary: a tool for prototyping location-enhanced applications. In *Proceedings of the 17th annual ACM symposium on User interface software and technology.* 217–226.

[41] Hao Lü and Yang Li. 2012. Gesture coder: a tool for programming multi-touch gestures by demonstration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* 2875–2884.

[42] Hao Lü and Yang Li. 2013. Gesture studio: authoring multi-touch interactions through demonstration and declaration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* 257–266.

[43] Blair MacIntyre, Maribeth Gandy, Steven Dow, and Jay David Bolter. 2004. DART: a toolkit for rapid design exploration of augmented reality experiences. In *Proceedings of the 17th annual ACM symposium on User interface software and technology.* 197–206.

[44] Nicolai Marquardt, Robert Diaz-Marino, Sebastian Boring, and Saul Greenberg. 2011. The proximity toolkit: prototyping proxemic interactions in ubiquitous computing ecologies. In *Proceedings of the 24th annual ACM symposium on User interface software and technology.* 315–326.

[45] Nicolai Marquardt and Saul Greenberg. 2012. Informing the design of proxemic interactions. *IEEE Pervasive Computing* 11, 2 (2012), 14–23.

[46] Nolwenn Maudet, Germán Leiva, Michel Beaudouin-Lafon, and Wendy Mackay. 2017. Design Breakdowns: Designer-Developer Gaps in Representing and Interpreting Interactive Systems. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing.* 630–641.

[47] Brad Myers, Sun Young Park, Yoko Nakano, Greg Mueller, and Andrew Ko. 2008. How designers design and program interactive behaviors. In *2008 IEEE Symposium on Visual Languages and Human-Centric Computing.* IEEE, 177–184.

[48] Yasuto Nakanishi. 2012. Virtual prototyping using miniature model and visualization for interactive public displays. In *Proceedings of the Designing Interactive Systems Conference.* 458–467.

[49] Tek-Jin Nam. 2005. Sketch-based rapid prototyping platform for hardware-software integrated interactive products. In *CHI'05 extended abstracts on Human factors in computing systems.* 1689–1692.

[50] Michael Nebeling and Katy Madier. 2019. 360proto: Making Interactive Virtual Reality & Augmented Reality Prototypes from Paper. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems.* 1–13.

[51] Michael Nebeling, Janet Nebeling, Ao Yu, and Rob Rumble. 2018. Protoar: Rapid physical-digital prototyping of mobile augmented reality applications. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems.* 1–12.

[52] Dan R Olsen Jr. 2007. Evaluating user interface systems research. In *Proceedings of the 20th annual ACM symposium on User interface software and technology.* 251–258.

[53] Hyungjun Park, Hee-Cheol Moon, and Jae Yeol Lee. 2009. Tangible augmented prototyping of digital handheld products. *Computers in Industry* 60, 2 (2009), 114–125.

[54] Sarah Prange and Florian Alt. 2020. I Wish You Were Smart (er): Investigating Users' Desires and Needs Towards Home Appliances. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems.* 1–8.

[55] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 779–788.

[56] Gang Ren, Wenbin Li, and Eamonn O'Neill. 2016. Towards the design of effective freehand gestural interaction for interactive tv. *Journal of Intelligent & Fuzzy Systems* 31, 5 (2016), 2659–2674.

[57] Nazmus Saquib, Rubaiat Habib Kazi, Li-Yi Wei, and Wilmot Li. 2019. Interactive body-driven graphics for augmented video performance. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems.* 1–12.

[58] Hartmut Seichter, Julian Looser, and Mark Billinghurst. 2008. ComposAR: An intuitive tool for authoring AR applications. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality.* IEEE, 177–178.

[59] Teddy Seyed, Alaa Azazi, Edwin Chan, Yuxi Wang, and Frank Maurer. 2015. Sod-toolkit: A toolkit for interactively prototyping and developing multi-sensor, multi-device environments. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces.* 171–180.

[60] Kihoon Son, Hwiwon Chun, Sojin Park, and Kyung Hoon Hyun. 2020. C-Space: An Interactive Prototyping Platform for Collaborative Spatial Design Exploration. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems.* 1–13.

[61] Maximilian Speicher and Michael Nebeling. 2018. GestureWiz: A human-powered gesture design environment for user interface prototypes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems.* 1–11.

[62] Ryo Suzuki, Rubaiat Habib Kazi, Li-Yi Wei, Stephen DiVerdi, Wilmot Li, and Daniel Leithinger. 2020. RealitySketch: Embedding Responsive Graphics and Visualizations in AR through Dynamic Sketching. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology.* 166–181.

[63] John Underkoffler and Hiroshi Ishii. 1999. Urp: a luminous-tangible workbench for urban planning and design. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems.* 386–393.

[64] Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Yuanzhi Cao, and Karthik Ramani. 2021. GesturAR: An Authoring System for Creating Freehand Interactive Augmented Reality Applications. In *The 34th Annual ACM Symposium on User Interface Software and Technology.* 552–567.

[65] Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Ke Huo, Yuanzhi Cao, and Karthik Ramani. 2020. CAPturAR: An Augmented Reality Tool for Authoring Human-Involved Context-Aware Applications. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology.* 328–341.

[66] Zeyu Wang, Cuong Nguyen, Paul Asente, and Julie Dorsey. 2021. DistanciAR: Authoring Site-Specific Augmented Reality Experiences for Remote Environments. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems.* 1–12.