MRCAT: In Situ Prototyping of Interactive AR Environments

Matt Whitlock[1(✉)], Jake Mitchell[1], Nick Pfeufer[1], Brad Arnot[1], Ryan Craig[1], Bryce Wilson[1], Brian Chung[1], and Danielle Albers Szafir[1,2,3]

[1] Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309, USA
{matthew.whitlock,jake.mitchell,nicholas.pfeufer, bradley.arnot,ryan.craig,bryce.d.wilson,brian.chung, danielle.szafir}@colorado.edu

[2] Department of Information Science, University of Colorado Boulder, Boulder, CO 80309, USA

[3] ATLAS Institute, University of Colorado Boulder, Boulder, CO 80309, USA

Abstract. Augmented reality (AR) blends physical and virtual components to create a mixed reality experience. This unique display medium presents new opportunities for application design, as applications can move beyond the desktop and integrate with the physical environment. In order to build effective applications for AR displays, we need to be able to iteratively design for different contexts or scenarios. We present MRCAT (Mixed Reality Content Authoring Toolkit), a tool for *in situ* prototyping of mixed reality environments. We discuss the initial design of MRCAT and iteration after a study ($N = 14$) to evaluate users' abilities to craft AR applications with MRCAT and with a 2D prototyping tool. We contextualize our system in a case study of museum exhibit development, identifying how existing ideation and prototyping workflows could be bolstered with the approach offered by MRCAT. With our exploration of *in situ* prototyping, we enumerate key aspects both of AR application design and targeted domains that help guide design of more effective AR prototyping tools.

Keywords: Augmented reality · Prototyping · Multimodal interaction

1 Introduction

Many display media have established iterative design workflows, where designers prototype at increased levels of fidelity to elicit feedback before the application is developed [18]. To elicit early feedback, designers will often employ simple sketch-based prototyping, but with continued iteration and increased fidelity, prototypes increasingly look as they will appear in the target display media (i.e., in a browser window for a web application or on a touchscreen for a mobile application). Despite notable research in AR content creation [25,29,38,42], AR

J. Y. C. Chen and G. Fragomeni (Eds.): HCII 2020, LNCS 12190, pp. 235–255, 2020.
https://doi.org/10.1007/978-3-030-49695-1_16

application prototyping lacks support for higher fidelity prototypes that allow designers to experience and refine prototypes in the target display.

Prototypes of AR applications typically come in the form of sketches and descriptions of how the environment should look. The practice of sketching and writing is accessible to designers and domain experts, relative to the alternative of developing the entire application in code. However, these sketches offer low fidelity representations of the idea in the designer's mind. Game engines such as Unity afford prototyping on a 2D display, but decontextualize virtual content from the real world. If designers were to instead prototype AR applications in AR, they could place virtual content in tandem with the physical environment. This *in situ* approach to prototyping AR environments allows designers and domain experts to truly express and experience their ideas as they become a reality, However, *in situ* AR prototyping tools do not yet support the needs of designers, in part due to lack of guidance on how to design effective *in situ* prototyping tools. With this paper, we address the benefits of an *in situ* approach to AR prototyping, discussing usability of prototyping tools and key aspects of environment design for AR prototyping tools to address going forward.

We introduce a tool for AR *in situ* prototyping called the Mixed Reality Content Authoring Toolkit (MRCAT) and discuss how our work with the tool elicited the needs for prototyping AR applications. We present a workflow for AR prototyping where designers can create, save, share and load AR experiences, placing, manipulating and annotating virtual models directly in the environment to craft mixed reality experiences *in situ*. We discuss the design of MRCAT in the context of common guidelines for prototyping tools and the results of a preliminary study exploring the needs of *in situ* prototyping tools compared to AR application prototyping in 2D. Through these efforts, we enumerate ways to increase the usability of AR prototyping tools and key aspects of *in situ* environment design for future prototyping tools.

2 Related Work

AR prototyping systems often either remove virtual content from physical context [29] or use an adapted form of sketching with ubiquitous materials like cardboard and glass [6], limiting the fidelity to the intended AR experience. Alternatively, prototypes offering higher fidelity often require programming knowledge in order to build [37]. As AR technology becomes more accessible, domain experts will increasingly need to be part of application design. Participatory design allows for ideation within a "third space" between technologists and domain experts that includes ideas novel to both fields through co-creation [31]. Within AR, previous systems have explored prototyping tools for domain experts such as educators [23] and museum exhibit curators [45] by editing video streams in a 2D AR browser. With our work, we explore how *in situ* prototyping can allow users to build high fidelity prototypes directly in the target environment using intuitive WYSIWYG tools. We build on past work in AR content creation and user interfaces (UIs) that will make AR prototyping workflows feasible to a broad range of users.

2.1 AR Content Creation

Research in AR content creation tools has explored different approaches to more intuitively create mixed reality experiences. Tools like DART [29] and ComposAR [38] allow users to augment video and picture representations of the physical environment with virtual content. Other content creation tools allow users to customize which 3D models are associated with different fiduciary markers in tangible AR applications [5,25,42]. With headsets having six degree-of-freedom tracking to localize within a room, markerless AR content creation tools allow users to place virtual objects in the room to change the appearance *in situ* [49].

In situ prototyping tools allow users to create and edit applications directly in the application's target environment. This approach is of particular interest as AR applications typically rely on blending virtual content and the physical space. For example, SceneCTRL allows users to edit arrangements of physical and virtual objects both by placing new objects and visually deleting existing physical objects. [49]. Built on the AMIRE content authoring framework, work on assembly tutorial authoring allows users to build AR tutorial components as they assemble the physical object [50]. Work in AR museum presentation authoring explores scene editing on a web browser [42] and is then extended to use a mobile phone to create and edit virtual models for a museum exhibit directly in the space [36]. While these use cases provide examples where designers can build with the display medium directly in the target environment, little is know about what exactly are the benefits to AR prototyping *in situ* or how to design applications that optimize for these benefits. Through our work with MRCAT, we propose design guidelines for AR prototyping tools and discuss scenarios in which *in situ* AR prototyping could improve existing design workflows.

2.2 AR Multimodal Interaction

UIs for prototyping tools must support a number of tasks. Effective UI design for *in situ* AR prototyping is further complicated by the fact that there are not standard interaction metaphors and best practices for AR UI design. Fluid interaction is critical to the success of prototyping AR applications. AR systems commonly make use of freehand gestures to manipulate object transforms [7,17] since the metaphor to grab and manipulate a virtual object maps to manipulation of physical objects. Freehand gestures have also been used to annotate [11,26], sketch [1,46], navigate menu systems [13,32] and update descriptive characteristics such as color [34]. Alternatives to gestural interaction include using mediating devices such as tangible markers [25], secondary tablet/phone displays [1,30] and video game controllers [43,44]. This disparate exploration of different modalities for interaction in AR makes it difficult to identify specific best practices when crafting an AR interface. However, performance differences across tasks indicate that multimodal interaction may provide more intuitive means for supporting the array of capabilities necessary to prototype AR experiences.

Research in multimodal interaction considers how input modalities can complement one another. For example, gaze plus gestural interaction typically utilizes the user's gaze for object specification and a hand gesture to perform object manipulation [9,12,41]. Voice is often used in tandem with gaze or freehand gestures. Users can use a gesture to specify an object to move and a voice command such as "move behind the table" to indicate where to move it [21,33] or to change the color or shape of an object [28]. Multimodal interactions can also provide *mutual disambiguation*, where input from multiple modalities probabilistically provides greater precision than either input on its own [24]. In AR prototyping, these multimodal approaches could provide greater accuracy and speed in interactions than individual modalities could achieve on their own, leading to more efficient design of high quality prototypes.

The target design tasks can also guide the best ways to interact with a system. For example, picking items in a data visualization may be well-suited to gestural interaction while higher-level commands like creating a new visualization would be well-suited to voice interaction [2]. High agreement scores in elicitation of translation, rotation and scaling gestures suggest that freehand gestures are intuitive for transform manipulations [34]. However, the low agreement scores for interface-level commands and descriptive characteristics suggest that a different modality should be employed for these tasks. To support fluid, interactive design *in situ*, we build on findings in multimodal interaction, utilizing gestural interaction to manipulate object transforms [7,9,17,40] and voice commands for descriptive characteristics [27,28] and interface-level commands [41,49].

3 Design Guidelines

We reviewed literature on prototyping methods and commercial prototyping tools to better understand limitations in current approaches and how these guidelines might extend to *in situ* approaches. We used these guidelines to create a preliminary version of MRCAT, grounded in prototyping best practices. MRCAT offers an extended suite of prototyping functionality, including directly placing/manipulating virtual objects and saving and loading scenes. While a complete survey of prototyping best practices is beyond the scope of this work (see Carter & Hundhausen [8] for a survey), we synthesized three guidelines from prior literature and our own experiences that extend these practices to the unique needs of *in situ* prototyping in AR:

D1: Full Experience Prototyping. To effectively create design artifacts, prototyping tools should allow designers to capture the intended experience and different application designs [3]. Traditional tools give designers the ability to design on a blank slate, adding GUI elements such as menus, text boxes and buttons. On the other hand, AR prototyping tools need to consider interactions of physical and virtual elements—both 2D and 3D—by giving designers the ability to enumerate relationships. For example, when prototyping an AR museum exhibit, designers should be able to place exhibit pieces on tables, floors and walls as they see fit. Considering that not all information may be represented by placement and manipulation of virtual models in the environment (e.g., the proposed interactive nature of the virtual object), AR prototyping tools should also consider methods to describe these additional needs. In MRCAT, we implement this guideline through combined model integration, transformation and text-based annotation.

D2: Intuitive UI. Creating interactive AR applications requires disparate design tasks (e.g., positioning models, mocking interactions, annotating models). While mouse and keyboard interactions would be efficient and familiar for these tasks in 2D prototyping tools such as proto.io[1], we need to consider task mappings for novel AR interfaces. Deeply nested menu structures common to 2D prototyping tools do not translate well to AR. In MRCAT, we guided interaction mappings with prior literature (Sect. 2.2), using freehand interactions for model manipulation, and voice commands for abstract operations such as deleting objects, changing color and saving.

D3: Constrained Interactions. Users should clearly understand how their input will change the environment. If the system does not clearly and efficiently convey how the interaction will affect the environment, users will be frustrated by unexpected outputs and will need to spend additional time correcting. We provide simple, understandable interactions by constraining the degrees of freedom manipulated at one time. For example, users can either be moving, scaling or rotating an object—but not more than one of these operations—at a time. This strategy is employed in common 3D modeling tools such as Unity[2] and Sketchup[3]. In MRCAT's implementation, we build on prior AR research that achieves this by mapping the same gesture to different functionalities [39]. Explicit mode switching—implemented via voice commands and menu options—ensures users know what manipulation they are performing (e.g. translation, rotation, scaling).

We combine these design guidelines with ideas from 2D prototyping tools and previous AR literature to implement a system for *in situ* AR prototyping.

4 MRCAT Preliminary Design

We built MRCAT to allow users to create and edit prototypes *in situ*. This system instantiates the design guidelines laid out in Sect. 3, providing full experience prototyping *(D1)*, an intuitive UI *(D2)* and constrained interactions *(D3)*.

4.1 System Overview

MRCAT is a prototyping tool built for the Microsoft HoloLens[4] that allows users to place and manipulate objects in the environment. Users can first copy desired prefabricated element (prefabs) or custom 3D models into the project folder to tailor the initial models to their target application. MRCAT starts by loading a billboarded main menu showing the functionality available to the user, including

[1] https://proto.io.
[2] https://unity.com.
[3] https://sketchup.com.
[4] https://docs.microsoft.com/en-us/hololens/hololens1-hardware.

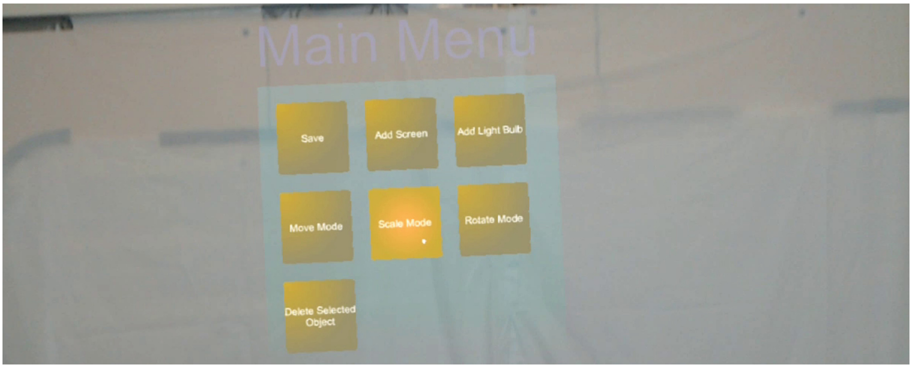

Fig. 1. MRCAT's main menu. Users can enter different interaction modes through the menu interface or through equivalent "MRCAT `mode`" commands. For example, the user can select "Scale Mode" from the menu, or say "MRCAT Scale."

a list of models extracted from the project folder (Fig. 1). MRCAT allows users to enable different modes that determine how their interaction will affect the selected objects: "Move", "Rotate", "Scale", "Annotate (Note)" and "Color".

To enter an interaction mode, the user can either select the mode from the set of options on the main menu or use a voice command (i.e. "MRCAT `mode`"). Users select menu options through the built-in gaze-tap gesture, where a cursor raycasted from the center of the user's gaze indicates which item to select with and a freehand tap gesture acts as a click. To engage with 3D models, the user selects an object with the same gaze-tap gesture, and subsequent interactions will affect all selected objects. Users can move, rotate or scale objects, depending on what interaction mode they have selected. This manipulation is done with the gaze-drag gesture, similar to the gaze-tap, but rather than a tap, the user presses down their finger to hold, moves their hand in front of the headset and finishes the interaction by releasing their finger back up.

After completing a prototype, the user can export the prototyped application as an XML file containing all relevant information about objects created. Files are relatively small (1.6 MB on average in the study, Sect. 5), and can be loaded into MRCAT to recreate and edit the scene. By changing the headset camera prefab prior to loading MRCAT, users can view the prototype in any headset, such as the GearVR, or on the desktop.

4.2 System Functionality

MRCAT enables users to interact with virtual content by moving between different interaction modes. To ensure that novice users are only able to perform one action at a time (*D3*), MRCAT employs voice commands and redundant menu options to allow the user to enter each interaction's mode. The user selects all objects they want to manipulate, rendering a red border around those objects to indicate engagement. The user can then either give the voice command "MRCAT

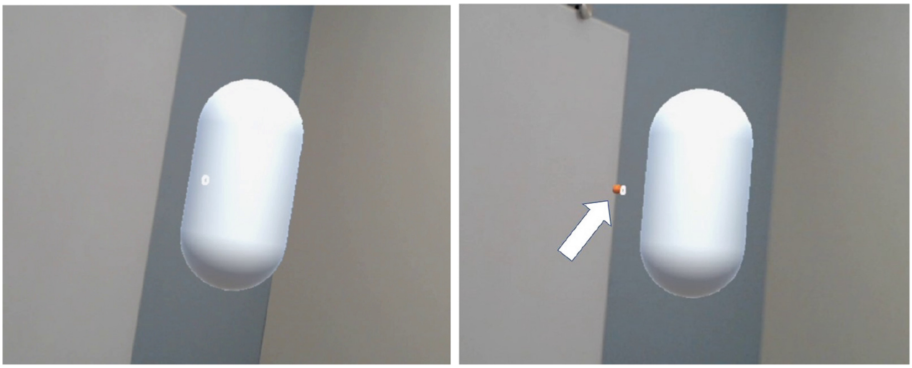

Fig. 2. When moving objects, a small yellow sphere (right) indicates attempted displacement. Here the user is trying to move the virtual capsule through a wall, but the capsule collides with and renders against the wall (arrow added for emphasis). (Color figure online)

`mode`" to enter that mode (e.g., "MRCAT Rotate") or select the corresponding menu option. This "MRCAT" initiation is similar to familiar voice-interaction with assistants such as Apple's Siri[5] and Amazon's Alexa[6] *(D2)*. Then the user performs the appropriate gesture to change all selected objects.

Object Placement/Translation: Users can add and reposition objects throughout the environment using MRCAT. To add an object, the user selects a menu item or says "Add `item name`". The user can then control the placement of a virtual object, moving the object with their gaze. The object sits 2 m in front of the middle of the user's gaze and a small yellow sphere appears in the middle of the object transform to indicate where the user is placing or attempting to place the object (Fig. 2). If the object collides with another virtual object or physical surface such as a floor, wall or table (as detected using the HMD's depth camera), the object temporarily rests on the surface it collided with. The yellow sphere visually cues that there has been a collision and that the user may need to move the object elsewhere. Once the user is satisfied with an object's location, they select the object again to finalize its position.

Users can also move objects already placed in the environment, entering "Move" mode with a "MRCAT Move" command. To move objects, users first select the object, outlining it in red. They can then use hand gestures to move the object to different points in the environment. We employ only gestures for position refinement to allow for more precise object placement, rather than coarse-grain object placement that allows users to quickly get the desired objects into the environment before fine-tuning the positioning. The user can freely position objects in the environment rather than having objects locked 2 m from their forward gaze, which may require them to crane their neck to precisely move

[5] https://apple.com/siri/.
[6] https://developer.amazon.com/en-US/alexa.

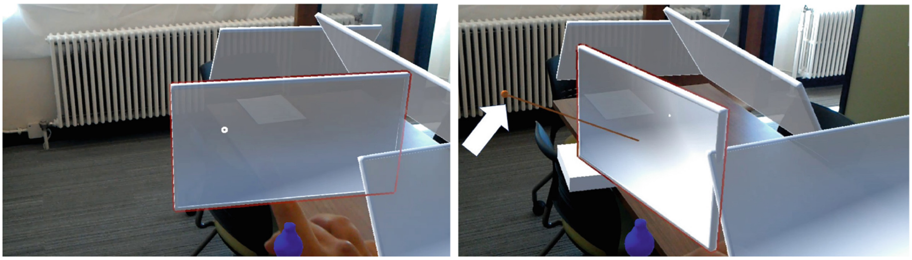

Fig. 3. Rotating a virtual screen about the Y axis. When the user begins dragging, a virtual joystick appears (right) to give the user an interaction metaphor of pulling a joystick to the left (label added for emphasis).

an object. Using the gaze-drag gesture, users can grab and move all selected objects, such that displacement of the hand along the X, Y, and Z axes maps proportionally to displacement of the selected objects. As with initial object placement, objects cannot be moved through a surface or another object. To end translation, the user releases the gaze-drag gesture.

Rotation: MRCAT allows users to rotate virtual objects placed in the environment. To rotate an object, the user says "MRCAT Rotate" and enters a rotation mode. As with translation, the user presses and holds their hand to begin rotating selected objects. In piloting, we found that users preferred to rotate objects about one axis at a time for better control and precision *(D3)*. MRCAT then processes whether the user's initial hand movement is primarily along the X, Y or Z axis, and locks the object to rotate about one axis. If the initial hand displacement is to the left or right relative to the user, the object rotates about the Y axis (yaw). Similarly, hand displacement up and down maps to rotation about the X axis (pitch), and hand displacement along the Z axis maps to rotation about the Z axis (roll). To provide a visual indicator of the rotation control, a stick with a ball at the end appears in the middle of the object's transform, inspired by the metaphor of pushing and pulling a joystick (Fig. 3).

Scale: MRCAT also allows users to resize placed objects, entering this mode through the "MRCAT Scale" command. To begin scaling, the user presses and holds their hand, establishing the hand position as the initial grab point. Displacement of the hand along both the X and Y axes corresponds to uniform scaling of the object along all axes simultaneously. Grabbing and dragging either up or to the right increases object size, while dragging either down or to the left decreases size. As with scaling functionality of popular 3D modeling software (such as Unity and SketchUp), scaling an object to a negative number results in a mirrored positive scaling. To finish scaling, the user releases their finger.

Change Material: Users can change object appearance through voice commands. To change the appearance of selected objects, the user says "MRCAT `material name`," and all selected objects will change to have that material. For simplicity, we limited materials to the colors of the rainbow, black, white and ghost (transparency).

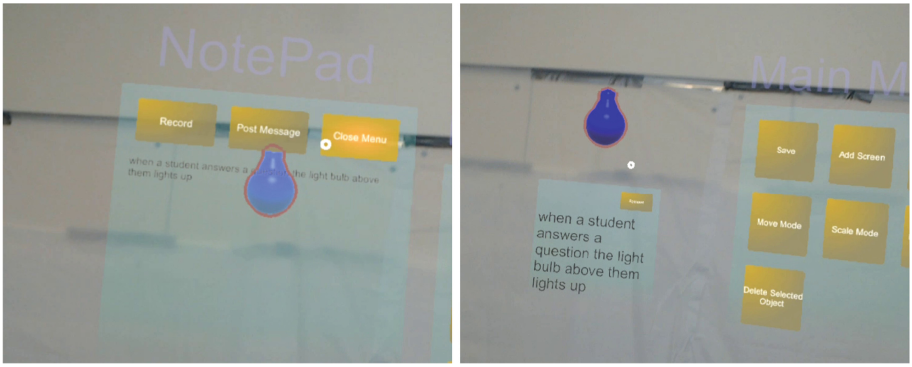

Fig. 4. Interface for recording user dictation as a note (left) and posting it to a virtual object (right). After pulling up the Notepad interface with the "MRCAT Note" command, users select the "Record" button to begin recording. Recording ends after two seconds of silence, at which point users can select the "Post Message" button to post the note to highlighted objects.

Annotation: MRCAT allows users to textually annotate objects in the scene to note additional ideas the designer has in mind or provide feedback on an element of the design *(D1)*. These annotations can indicate relationships between objects, fill in gaps where the 3D model design may fall short or allow for prototype feedback *in situ*. Users can annotate an object by first entering annotation mode by saying "MRCAT Note." An annotation interface then appears with buttons for recording, posting and closing (Fig. 4). As with the main menu, the annotation interface and the notes placed in the environment are billboarded to always face the user. To record a text annotation, the user says "MRCAT Note," and MRCAT plays a short "listening" audio clip to indicate that recording has begun. The user then dictates the note and the recording ends when the user stops speaking for 2 s. The user says "MRCAT Post" to render the note as a sticky-note style panel with a "Remove" button. The note appears above the object and moves with the object. To avoid occlusion, the note renders above its associated object if the user is looking down at the object and will render below the object if the user is looking up or directly at the object.

We employed basic dictation recognition for text entry, as text entry remains an open problem in AR and VR research [16,47,48]. Voice dictation is relatively fast but inaccurate, whereas use of a gesture-enabled virtual keyboards are slow but accurate [19]. Due to the already prominent use of voice interaction in MRCAT and a prioritization of speed over accuracy, we chose dictation for freeform text input, likened to post-it notes. The primary limitation of this approach is that mistakes result in a re-recording, rather than editing individual words. We anticipate annotations being fairly brief as designers typically prioritize visual depictions over long descriptions when building prototypes [3]. The brief nature of these annotations mitigates this limitation, as re-recording the intended text is generally inexpensive.

System-Level Interactions: Saving and loading prototypes is implemented using voice commands. Users can export prototyped scenes to XML files using the command "MRCAT Save" or by selecting the "Save" item on the main menu. MRCAT then plays the "listening" sound to indicate recording has begun, and the user says the name of the target XML file. The file is saved as the name specified by the user with spaces as underscores. MRCAT then plays audio "Space saved as `file name`." To load a saved file, the user says "MRCAT Load," followed by a file name.

5 Evaluation

To identify the benefits of prototyping *in situ*, we evaluated MRCAT against a 2D prototyping alternative built in Unity. We conducted a 2 (prototyping tool) × 2 (scenario) mixed factors study with 14 participants (12M, 2F). The study asked participants to prototype two IoT configurations with smart devices: a conference room and a classroom. In each scenario, participants completed four tasks: create a preliminary design, illustrate an example use case, integrate a new design constraint, and propose an alternative design.

5.1 Desktop Prototyping

For our user study, we use a subset of functionality from Unity to parallel the functionality of MRCAT. Desktop content creation tools [38,42] typically use a hierarchical view of objects, available to Unity users in a "Hierarchy" view pane. Unity provides icons in the top left part of the UI that allow the user to switch between rotation, translation and scaling modes. Unity also has built-in functionality to allow users to drag pre-built objects into the scene, and to change their materials. Using a prefabricated Note element, we also allow users to drag annotations onto objects to label them To begin study tasks, we provide an initial mock-up of the target environment's furniture arrangement to start.

5.2 Scenarios

We described two IoT-based scenarios to participants that would be relatively familiar, but also required similar considerations to effectively prototype in AR. We chose these IoT-based scenarios for two reasons. First, prototyping interactive AR applications may be a foreign concept so asking participants to prototype something like an AR game may be a challenging task to understand. With smart devices becoming increasingly ubiquitous, building an IoT application provides participants with a more familiar set of tasks. Additionally, a key aspect of prototyping interactive environments is designing for the interplay of physical and virtual content *(D1)*. We conducted the study in a conference room with tables and chairs, and both scenarios require that participants utilize the layout of the space in the design of the interactive environment. Each participant completed both scenarios: one with MRCAT and one with Unity. Scenarios were counterbalanced between participants.

The smart conference room scenario required participants to prototype a network of connected smart devices that would improve collaborative screen sharing in meetings in the conference room. The four required tasks in this scenario were: *Initial Design:* Add primary and secondary displays to the room with smart light bulbs for each of the four people in the room to later associate users with displays. *Example Use Case:* Illustrate usage of the smart conference room where meeting members connect to the smart displays. *New Design Constraint:* Prototype what it would look like if light bulbs needed to hang from the ceiling. *Alternative Configuration:* Change the displays to be on different walls and resize them to be of equal priority, rather than a primary and secondary.

In the learning room scenario, we gave participants tasks to prototype a room that facilitates learning through a collaborative tabletop display and quiz questions. The four required tasks in this scenario were as follows: *Initial Design:* Prototype using the table as a tabletop display for all students to use and smart light bulbs at each seat for the four individual students. *Example Use Case:* Illustrate usage of the prototyped environment where students in the room are answering an administered quiz question. *New Design Constraint:* Prototype a similar environment where each student has a tablet-sized display at their seat. *Alternative Configuration* The table may be too small for a dedicated display, so explore an alternative where the screen is wall-mounted instead.

5.3 Procedure

After signing a consent form and being briefed on the study, participants were shown how to use prototyping tool, walking through the interactions and allowing them to practice. We then instructed participants on the scenario's tasks one at a time, exiting the environment while participants completed each task. After all tasks in the scenario, we administered a questionnaire to elicit feedback through the System Usability Survey (SUS) and 17 additional Likert-scale questions measuring the perceived efficacy and usability of the tool followed by open-ended feedback. Participants repeated the process with the second scenario and second prototyping tool before completing a 24 Likert-scale question survey with open-ended feedback asking participants to directly compare the two methods for accomplishing particular tasks. Participants then completed a demographics questionnaire and were compensated with a $10 Amazon gift card.

5.4 Results

We collected objective and subjective measures related to the usability of each paradigm for effective prototyping. We measured time to completion and questionnaire responses, using open-ended feedback to contextualize our results.

Overall, the objective measures pointed to significant limitations of MRCAT in comparison with Unity. Participants generally took longer with MRCAT ($\mu = 28.3\,\text{min}$) than with Unity ($\mu = 25.78\,\text{min}$). Participants reported higher

System Usability Score (SUS) for Unity ($\mu = 72.1$) than for MRCAT ($\mu = 51.9$). Participants also typically preferred Unity when asked to directly compare the two for particular tasks. Specifically, they preferred the Unity for object placement, translation, rotation, scaling and annotating, while preferring MRCAT only for object recoloring.

Despite objective limitations, open ended feedback pointed to significant opportunities for *in situ* prototyping, offering potential improvements to system design and hardware limitations that help us reason about displays and interaction modalities. Participants responded positively to editing and navigating the virtual and physical environment in parallel. They noted positive aspects of being "able to interact with the world in 3D and to be able to see what [they are] trying to do in real time. "*(P10)* and to "see what [they] built from multiple angles easily." *(P3)*. Participants also identified that "working the actual room was useful...to get a better sense of scale" *(P1)* and saw value in "getting a true feel for environment" (P5). This heightened sense of scale enabled them to prototype to higher fidelity. Participants stated that MRCAT "definitely allowed for the user to better visualize how the room would look in reality, which is a pretty significant advantage over Unity. Seeing exactly where everything would theoretically go in person is a much different experience than exploring a room through a computer." *(P9)* and that a prototype built with MRCAT "could be much closer to a convincing prototype" *(P14)*.

The most negative feedback for MRCAT and in favor of the 2D prototyping tool related to the inefficiencies of transform manipulations in MRCAT. Participants felt like "[users] can be a lot more precise using a mouse and keyboard" *(P6)* and that "the HoloLens tool wasn't as accurate with the placement of the objects, so [they] couldn't get things to look exactly how [they] wanted." *(P12)*. Among the specific operations, object rotations were most often noted as problematic (6 our of 14 participants). We built on feedback provided by participants to revise our transform manipulation model (Sect. 6).

Another salient theme from open-ended feedback was frustration with the headset itself. Responses indicated that the hardware likely contributed to higher frustration and lower usability scores in the Likert and SUS measures. One participant noted that "the weight of the HoloLens on my head...discouraged me from looking upwards" *(P7)*. Limited field of view was also cited as a possibly confounding factor, with a participant pointing out that the "HoloLens was...difficult to use, not because of the complexity, but because of the limited vision" *(P14)*. Though these factors are difficult to disentangle from inefficiencies in MRCAT and will likely be mitigated with future AR headsets, they are worth considering in design of AR prototyping tools going forward.

6 Design Iteration

Open-ended feedback illuminated several opportunities to improve the design of MRCAT. We specifically identified issues participants had with understanding the system's current state for object interaction and their mental model of how

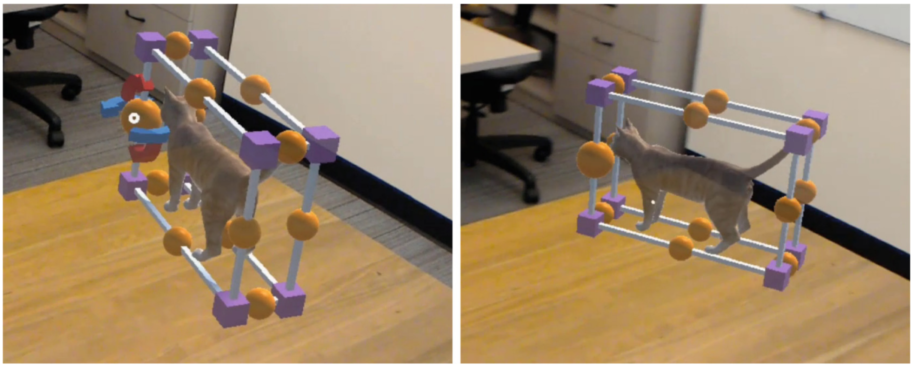

Fig. 5. The revised MRCAT interface uses wireframe cubes for transform manipulation. These cubes allow for continuous translation (dragging the object), scaling (dragging a corner cube), and rotation (dragging a sphere on an edge). When the user first engages with a rotation sphere, two sets of arrows indicate which axes the user can rotate (left). An initial movement to the left begins rotation about the vertical y-axis (right).

their input affects the prototype. In light of the feedback on transform manipulations with MRCAT, we add an additional design guideline to those in Sect. 3:

D4: Visual and Continuous Transform Manipulation. Though 2D prototyping tools make use of explicit mode-switching to map multiple functions to mouse dragging in 2D, our study found that this paradigm did not work as well in AR. AR prototyping tools should consider how 3D visual interfaces can provide users with an intuitive interface to manipulate objects continuously, without explicitly stating which mode they wish to be in. This increased continuity and system transparency should help reduce frustration and time to perform multiple transform manipulations. Iterating on MRCAT's UI design, we focused on allowing users to navigate to different perspectives and manipulate objects with more visual guidance on how their gestures will impact the selected objects.

To address this, we implement a 3D wireframe cube (Fig. 5) explored in prior AR studies [9,10]. Selecting an object with the gaze-tap toggles the wireframe cube around that object. Like the red outline in the previous iteration of MRCAT, the wireframe cube indicates engagement with objects for transform manipulations, color changes, and annotations. Performing a gaze-drag on the object translates all selected objects. Grabbing and dragging a blue box on the wireframe's corner uniformly scales all selected objects. Grabbing and dragging a sphere on one of the wireframe's edges allows the user to rotate the selected objects along one axis at a time. As with the preliminary implementation's "Rotation" mode, initial hand displacement determines which direction the object rotates. Engaging with one of the wireframe's 12 spheres provides users arrows indicating two possible rotation directions.

To mitigate issues with the HoloLens' gaze-drag gesture and limited field of view when manipulating objects, we added visual feedback to indicate lost

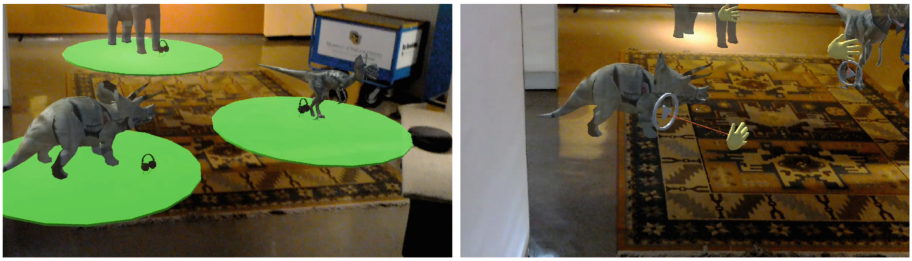

Fig. 6. Example usage of MRCAT in prototyping an interactive dinosaur exhibit. Here the designer prototypes proxemic interactions that trigger audio clips (left) and prototypes gestural interaction to trigger animations (right).

hand-tracking. An inherent limitation of freehand gestures relative to mouse-and-keyboard ones is that the user does not necessarily know when the their hand is outside of the headset's tracking area. In the preliminary evaluation, this lost hand-tracking caused participants to sometimes "[drag their] hand outside the screen several times on accident" *(P1)*. Since freehand gestures cannot provide any haptic feedback like a vibration or the sensation of a released button, we supplement MRCAT with a subtle visual flash when hand-tracking is lost. When the user moves their hand outside the field of view while manipulating an object, MRCAT tints the scene to a dark gray. The gray tint slowly fades away over the next two seconds. This visual feedback on system state is employed rather than audio because the user is engaging with a visual interface to manipulate objects, rather than audio feedback used when saving and loading scenes. This capability allows the user to understand that the headset is no longer processing their input and that they will need to restart the gaze-drag in order to continue.

Informal user feedback from MRCAT's revised design suggests that the updated UI significantly improves several usability issues from our study. This feedback provides promising evidence that many of the usability drawbacks in our initial study reflect a lack of design knowledge for effective *in situ* AR prototyping tools rather than limitations of *in situ* approaches generally. While we hope to elicit additional insight into these challenges in future studies, we evaluated our revised MRCAT approach through a case study in museum studies.

7 Case Study: Museum Exhibit Prototyping

We anticipate that AR prototyping will bolster AR integration into a number of domains. However, AR has not gained traction in many domains at least in part due to the lack of tools for domain experts to develop applications. We see *in situ* prototyping as a key component of overcoming this limitation and demonstrate the utility of *in situ* AR prototyping through museum exhibit development. We worked with an exhibit developer at a Natural History museum to build grounded insight into how *in situ* prototyping may benefit domain experts.

We worked with the exhibit developer to build foundational knowledge of existing ideation, collaboration, and prototyping workflows used to create new exhibits through a series of discussions about the potential for technology to support exhibit development and how MRCAT could support these practices. We began by characterizing existing workflows in museum exhibit development, followed by a brief demonstration of MRCAT, and concluded with how *in situ* AR prototyping may supplement existing practices. Most of the discussion centered around an in-progress exhibit at a heritage center, where the exhibit's design challenges were in utilizing a historic building to share contemporary and historic perspectives in accessible ways for diverse audiences.

Ideation: The initial ideation phase consists of brainstorming possible solutions and aggregating to common themes. Exhibit development teams typically brainstorm ideas on post-it notes, which then required clustering, organizing and digitizing. The museum team typically wants to first brainstorm design ideas then critically reflect on their options conceptually and graphically. For example, teams could test possible configurations to get at questions of intended experience: Is it hands-on? Are there videos? Is it a sensory experience? Fluid AR prototyping would enable them to try different experiences quickly and at minimal time and material cost. The ability to rapidly edit and reload scenes in MRCAT could enable curators to quickly explore these different representations.

Building Prototypes: The exhibit developer saw the most notable value of *in situ* prototyping as enabling people to visually communicate exhibit ideas more concretely to allow stakeholders to more accurately respond the approaches the exhibit development team is exploring. Currently, to move towards a final design, the exhibit team will create sketches, blueprints, CAD renderings, color and font palettes, and preliminary graphics and labels. The exhibit team may print prototype materials on cardboard to build physical prototypes. Relative to this approach, MRCAT would allow for higher fidelity prototypes that are significantly closer to what the end experience may look like.

Collaboration: Collaborating on museum exhibits requires input from a number of key stakeholders. In this case, the stakeholders included tribal leaders, University students, teachers, anthropologists, and a core group of museum employees including our expert. While some collaboration is in-person, a majority is done over phone calls. In both scenarios, higher fidelity prototypes built in AR would likely scaffold conversations better than sketches and descriptions. Even with a greater initial investment of time, our expert mentioned that they could communicate their entire vision to large groups of people at once, but that any such system would require technological training. With MRCAT's ability to save and load XML prototypes, users can review, build on and annotate prototypes for a more collaborative design workflow.

Our demonstration of MRCAT for museum exhibit development provided exciting opportunities for future systems. Our expert envisioned that AR prototyping done correctly could allow developers to explore a wide range of ideas, populating virtual galleries to reflect a diversity of options but requiring models that span diverse possible content. Our interviews revealed that *in situ* prototyping workflows could enable more rapid exploration of different designs, increased fidelity and improved local and remote collaboration.

8 Discussion

MRCAT explores how *in situ* prototyping can positively impact design workflows for AR. We derive and refine guidelines for building *in situ* prototyping tools and show in a use case how these tools could benefit a broad set of domains.

8.1 Key Aspects for AR Prototyping

Our discussions with a museum exhibit developer exemplifies how *in situ* AR prototyping could immediately benefit design workflows. Pairing these discussions with study feedback suggests two key benefits for in situ prototyping tools.

Constraints of the Physical Environment: *In situ* AR prototyping emphasizes the importance of environmental constraints. Our participants and domain expert noted that an idea that proposes virtual augmentations in tandem with the physical environment itself is better represented by a prototype built *in situ* than one made through decontextualized techniques. In designing museum exhibits, prototyping *in situ* gives designers the ability to understand the scale of the proposed exhibit and the interplay with existing infrastructure, while more traditional techniques like sketching do not allow for high fidelity illustration of the proposed exhibit and do not consider the environmental constraints.

In cases where the look, feel and scale of the virtual objects in the target space are critical, *in situ* AR prototyping can provide a means to design for the interplay of the physical and virtual environment. In the preliminary study, users could directly manipulate virtual objects in the real world and leverage interactions between virtual and physical objects. For example, users would place virtual lightbulbs directly in the physical space and resize virtual screens such that they were positioned on the physical tables and walls. In designing exhibits, developers wanted to test out configurations for the allotted space. Prototypes built directly in the target environment enable designers to seamlessly blend physical and virtual content.

Co-located and Remote Collaboration: *In situ* prototyping can enable new means of collaboration. By saving and loading XML representations of scenes within the target environment, multiple designers can iterate on prototypes by adding, reconfiguring and annotating saved arrangements. Extending MRCAT to allow multiple AR headsets to view the same scene would allow for synchronous, multi-user editing and design review within the environment.

Collaboration was identified as a significant potential benefit of AR prototyping in our discussions with museum experts. The higher fidelity prototype built *in situ* would provide collaborators a better sense of the idea in each designer's mind. MRCAT allows prototypes to be loaded into VR headsets or 2D displays, making prototypes more portable, but removing virtual content from its physical context. By extending MRCAT's XML files with depth and RGB camera capture, prototypes could be exported with a representation of the physical environment for remote viewers in VR or on a desktop to view the prototype in reconstructed contexts.

In addition to museum exhibit development, *in situ* prototyping may also benefit a number of other domains. For example, prior work has explored AR as a UI for connected smart environments [14,22]. AR prototyping would allow designers to propose configurations of integrated smart environments and craft integrated device interfaces. AR has also shown promise for improving education [4,15,20]. However, successfully integrating these platforms into classrooms requires empowering educators with control over interface design to engage more fluidly with lesson plan design and educational content. This would allow educators to understand holistically how students may interact with the technologies to facilitate better educational outcomes. Theater sets increasingly integrate technologies to deliver unique performances [35]. *In situ* prototyping allows set designers and directors to more fluidly integrate novel interactive components and experience different designs from a variety of perspectives.

8.2 Usability and Longitudinal Study

Our preliminary study of MRCAT revealed the need for thorough consideration of effective multimodal interaction in AR. After our preliminary study, we prioritized increased visual feedback to enable fluid transform manipulation *(D4)*. Use of the 3D wireframe cube instead of explicit mode switching allowed for more fluid object manipulation and increased understanding of how exactly the performed gesture will affect virtual objects. Continued study of these interactions will improve interface efficiency for MRCAT and other AR prototyping tools.

Another important consideration for AR prototyping is how particular modalities map to different design tasks. We designed our interface based on mappings commonly used in prior literature (Sect. 2.2), but continued study of which tasks will be better suited to different interaction modalities will improve the usability of *in situ* prototyping tools. For example, while freehand gestures are commonly used for transform manipulations [7,9,17], they should be evaluated against multimodal gaze plus voice interactions [21,33] or video game controllers [43,44] to empirically confirm the decision. We employed voice for text entry, prioritizing speed over accuracy [19]. This decision matched participants' desires to use the annotation interface for relatively short, post-it style annotations, such as "the screen does not have to be a smart table. It can also be a secondary wall screen" *(P6)* and "the question gets displayed on all of the tablets in the students input their answers" *(P7)*. With continued research on HMD text entry, the basic dictation recognition used in MRCAT should be substituted with future work on text entry optimized for HMDs.

Longitudinal study of AR prototyping tools will be critical to effective adoption of *in situ* prototyping workflows. Our discussion of museum exhibit development identified the potential for developers to employ AR when prototyping exhibits. However, integration of *in situ* AR prototyping into museum design workflows and observing how the tools shape design practices usage will reveal further guidelines for effective AR prototyping tools.

Longitudinal study could also reveal how to best extend the breadth of prototyping features available to designers. For example, MRCAT currently does not integrate interactive components such as responsive objects as there are not well-documented standards for prototyping rich interactions without scripting or complex menu structures. In the preliminary study, users employed annotations to describe interactivity and in Fig. 6, we demonstrate how virtual models can visually depict interactivity. Further extensions of MRCAT could enable designers to mock the interactions to build fully interactive prototypes, allowing them to conduct user studies with interactive AR prototypes. With increased comfort from continued use of AR prototyping tools, we could test this extended functionality to make interactive prototypes better than with a usability study.

9 Conclusion

In situ prototyping allows designers to build and ideate on interactive mixed reality applications within a target environment. We implement a tool for building *in situ* prototypes called MRCAT, through which we identify guidelines for design of such prototyping tools. Through a preliminary user study, we identify trade-offs between *in situ* and decontextualized 2D prototyping. We consider several aspects of *in situ* prototyping: system design, multimodal user interaction and key domains that may benefit from an *in situ* approach. This work provides a roadmap for *in situ* AR prototyping research, such that AR prototyping tools could enable higher fidelity and more collaborative design in AR.

Acknowledgements. We thank the CU Senior Projects program for facilitating development of MRCAT. This work was supported by NSF Award #1764092.

References

1. Arora, R., Habib Kazi, R., Grossman, T., Fitzmaurice, G., Singh, K.: SymbiosisSketch: combining 2D & 3D sketching for designing detailed 3D objects in situ. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York (2018)
2. Badam, S.K., Srinivasan, A., Elmqvist, N., Stasko, J.: Affordances of input modalities for visual data exploration in immersive environments. In: 2nd Workshop on Immersive Analytics (2017)
3. Beaudouin-Lafon, M., Mackay, W.E.: Prototyping tools and techniques. In: Human-Computer Interaction, pp. 137–160. CRC Press (2009)
4. Beheshti, E., Kim, D., Ecanow, G., Horn, M.S.: Looking inside the wires: understanding museum visitor learning with an augmented circuit exhibit. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 1583–1594. Association for Computing Machinery, New York (2017)

5. Billinghurst, M., Kato, H., Poupyrev, I.: The magicbook - moving seamlessly between reality and virtuality. IEEE Comput. Graph. Appl. **21**(3), 6–8 (2001)
6. Broy, N., Schneegass, S., Alt, F., Schmidt, A.: Framebox and mirrorbox: tools and guidelines to support designers in prototyping interfaces for 3D displays. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2037–2046. Association for Computing Machinery, New York (2014)
7. Buchmann, V., Violich, S., Billinghurst, M., Cockburn, A.: FingARtips: gesture based direct manipulation in augmented reality. In: Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, pp. 212–221. Association for Computing Machinery, New York (2004)
8. Carter, A.S., Hundhausen, C.D.: How is user interface prototyping really done in practice? a survey of user interface designers. In: IEEE Symposium on Visual Languages and Human-Centric Computing, pp. 207–211, September 2010
9. Chaconas, N., Höllerer, T.: An evaluation of bimanual gestures on the Microsoft hololens. In: IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 1–8 (2018)
10. Chakraborty, A., Gross, R., McIntee, S., Hong, K.W., Lee, J.Y., St. Amant, R.: Captive: a cube with augmented physical tools. In: Extended Abstracts on Human Factors in Computing Systems, CHI 2014, pp. 1315–1320. Association for Computing Machinery, New York (2014)
11. Chang, Y.S., Nuernberger, B., Luan, B., Höllerer, T.: Evaluating gesture-based augmented reality annotation. In: IEEE Symposium on 3D User Interfaces (3DUI), pp. 182–185 (2017)
12. Chang, Y.S., Nuernberger, B., Luan, B., Höllerer, T., O'Donovan, J.: Gesture-based augmented reality annotation. In: IEEE Virtual Reality (VR), pp. 469–470 (2017)
13. Dachselt, R., Hübner, A.: Three-dimensional menus: a survey and taxonomy. Comput. Graph. **31**(1), 53–65 (2007)
14. Garcia Macias, J.A., Alvarez-Lozano, J., Estrada, P., Aviles Lopez, E.: Browsing the internet of things with sentient visors. Computer **44**(5), 46–52 (2011)
15. Giraudeau, P., et al.: Cards: a mixed-reality system for collaborative learning at school. In: Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces, pp. 55–64 (2019)
16. Grossman, T., Chen, X.A., Fitzmaurice, G.: Typing on glasses: adapting text entry to smart eyewear. In: Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI 2015, pp. 144–152. Association for Computing Machinery, New York (2015)
17. Ha, T., Feiner, S., Woo, W.: WeARHand: Head-worn, RGB-D camera-based, bare-hand user interface with visually enhanced depth perception. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 219–228 (2014)
18. Hall, R.R.: Prototyping for usability of new technology. Int. J. Hum. Comput. Stud. **55**(4), 485–501 (2001)
19. Hoste, L., Dumas, B., Signer, B.: SpeeG: a multimodal speech- and gesture-based text input solution. In: Proceedings of the International Working Conference on Advanced Visual Interfaces, pp. 156–163. Association for Computing Machinery, New York (2012)
20. Ibáñez, M.B., Di Serio, Á., Villarán, D., Kloos, C.D.: Experimenting with electromagnetism using augmented reality: impact on flow student experience and educational effectiveness. Comput. Educ. **71**, 1–13 (2014)

21. Irawati, S., Green, S., Billinghurst, M., Duenser, A., Ko, H.: "Move the couch where?": developing an augmented reality multimodal interface. In: IEEE/ACM International Symposium on Mixed and Augmented Reality, pp. 183–186 (2006)
22. Jahn, M., Jentsch, M., Prause, C.R., Pramudianto, F., Al-Akkad, A., Reiners, R.: The energy aware smart home. In: 5th International Conference on Future Information Technology, pp. 1–8 (2010)
23. Jee, H.-K., Lim, S., Youn, J., Lee, J.: An augmented reality-based authoring tool for E-learning applications. Multimedia Tools Appl. **68**(2), 225–235 (2011). https://doi.org/10.1007/s11042-011-0880-4
24. Kaiser, E., et al.: Mutual disambiguation of 3D multimodal interaction in augmented and virtual reality. In: Proceedings of the 5th International Conference on Multimodal Interfaces, pp. 12–19. Association for Computing Machinery, New York (2003)
25. Lee, G.A., Nelles, C., Billinghurst, M., Kim, G.J.: Immersive authoring of tangible augmented reality applications. In: Third IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 172–181 (2004)
26. Lee, G.A., Teo, T., Kim, S., Billinghurst, M.: Mixed reality collaboration through sharing a live panorama. In: SIGGRAPH Asia 2017 Mobile Graphics and Interactive Applications. Association for Computing Machinery, New York (2017)
27. Lee, M., Billinghurst, M.: A wizard of Oz study for an AR multimodal interface. In: Proceedings of the 10th International Conference on Multimodal Interfaces, pp. 249–256. Association for Computing Machinery, New York (2008)
28. Lee, M., Billinghurst, M., Baek, W., Green, R., Woo, W.: A usability study of multimodal input in an augmented reality environment. Virtual Reality **17**(4), 293–305 (2013)
29. MacIntyre, B., Gandy, M., Dow, S., Bolter, J.D.: 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, pp. 197–206. ACM (2004)
30. Millette, A., McGuffin, M.J.: DualCAD: integrating augmented reality with a desktop GUI and smartphone interaction. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct), pp. 21–26 (2016)
31. Muller, M.J.: Participatory design: the third space in HCI. Chapter. In: The Human-computer Interaction Handbook, pp. 1051–1068. L. Erlbaum Associates Inc., Hillsdale (2003)
32. Ni, T., Bowman, D.A., North, C., McMahan, R.P.: Design and evaluation of freehand menu selection interfaces using tilt and pinch gestures. Int. J. Hum. Comput. Stud. **69**(9), 551–562 (2011)
33. Piumsomboon, T., Altimira, D., Kim, H., Clark, A., Lee, G., Billinghurst, M.: Grasp-shell vs gesture-speech: a comparison of direct and indirect natural interaction techniques in augmented reality. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 73–82 (2014)
34. Piumsomboon, T., Clark, A., Billinghurst, M., Cockburn, A.: User-defined gestures for augmented reality. In: Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J., Winckler, M. (eds.) INTERACT 2013. LNCS, vol. 8118, pp. 282–299. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40480-1_18
35. Ponto, K., Lisowski, D., Fan, S.: Designing extreme 3D user interfaces for augmented live performances. In: IEEE Symposium on 3D User Interfaces (3DUI), pp. 169–172, March 2016

36. Rumiński, D., Walczak, K.: Creation of interactive AR content on mobile devices. In: Abramowicz, W. (ed.) BIS 2013. LNBIP, vol. 160, pp. 258–269. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41687-3_24
37. de Sá, M., Churchill, E.: Mobile augmented reality: exploring design and prototyping techniques. In: Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services, pp. 221–230. Association for Computing Machinery, New York (2012)
38. Seichter, H., Looser, J., Billinghurst, M.: ComposAR: an intuitive tool for authoring AR applications. In: 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, pp. 177–178 (2008)
39. Smith, J., Wang, I., Woodward, J., Ruiz, J.: Experimental analysis of single mode switching techniques in augmented reality. In: Proceedings of the 45th Graphics Interface Conference on Proceedings of Graphics Interface 2019. Canadian Human-Computer Communications Society, Waterloo (2019)
40. SyafiqahSafiee, N., Ismail, A.W.: Ar home deco: virtual object manipulation technique using hand gesture in augmented reality. In: Innovations in Computing Technology and Applications, vol. 3 (2018)
41. Turini, G., et al.: A Microsoft hololens mixed reality surgical simulator for patient-specific hip arthroplasty training. In: De Paolis, L.T., Bourdot, P. (eds.) AVR 2018. LNCS, vol. 10851, pp. 201–210. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95282-6_15
42. Walczak, K., Wojciechowski, R.: Dynamic creation of interactive mixed reality presentations. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, pp. 167–176. Association for Computing Machinery, New York (2005)
43. Walker, M.E., Hedayati, H., Szafir, D.: Robot teleoperation with augmented reality virtual surrogates. In: 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 202–210 (2019)
44. Whitlock, M., Harnner, E., Brubaker, J.R., Kane, S., Szafir, D.A.: Interacting with distant objects in augmented reality. In: IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 41–48 (2018)
45. Wojciechowski, R., Walczak, K., White, M., Cellary, W.: Building virtual and augmented reality museum exhibitions. In: Proceedings of the Ninth International Conference on 3D Web Technology, pp. 135–144. ACM, New York (2004)
46. Wolf, D., Dudley, J.J., Kristensson, P.O.: Performance envelopes of in-air direct and smartwatch indirect control for head-mounted augmented reality. In: IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 347–354 (2018)
47. Xu, W., Liang, H., Zhao, Y., Zhang, T., Yu, D., Monteiro, D.: RingText: dwell-free and hands-free text entry for mobile head-mounted displays using head motions. IEEE Trans. Visual Comput. Graphics **25**(5), 1991–2001 (2019)
48. Yu, C., Gu, Y., Yang, Z., Yi, X., Luo, H., Shi, Y.: Tap, dwell or gesture? Exploring head-based text entry techniques for HMDs. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 4479–4488. Association for Computing Machinery, New York (2017)
49. Yue, Y.T., Yang, Y.L., Ren, G., Wang, W.: SceneCtrl: mixed reality enhancement via efficient scene editing. In: Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, pp. 427–436. ACM, New York (2017)
50. Zauner, J., Haller, M., Brandl, A., Hartman, W.: Authoring of a mixed reality assembly instructor for hierarchical structures. In: Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality 2003, pp. 237–246 (2003)