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,
[2] Department of Information Science
[3] ATLAS Institute,
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

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. With our work on *in situ* prototyping of interactive environments, we present MRCAT (Mixed Reality Content Authoring Toolkit). We discuss initial design of MRCAT and iteration after a study ($N = 14$) to evaluate users' abilities to craft interactive environments with MRCAT and with a 2D prototyping tool. We contextualize our system in a case study of museum exhibit curation, identifying how existing ideation and prototyping workflows could be bolstered with the approach offered by MRCAT. With our exploration of *in situ* AR prototyping, we enumerate key aspects both of environment design and targeted domains that provide a way forward for research on AR prototyping tools.

Keywords: Augmented reality · Prototyping · Multimodal interaction.

1 Introduction

Other display media have established iterative design workflows, where designers prototype at increased level of fidelity to elicit feedback prior before the application is developed [16]. To elicit early feedback, designers will often employ simple sketch-based prototyping, but with continued iteration and increased fidelity, prototypes should increasingly look appear in the target display media (i.e. a browser window for a web application or touchscreen for a mobile application). Despite a fair amount of research in AR content creation [27, 36, 5, 23, 40], AR application prototyping does not have an established workflow.

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 to the idea in the designers mind in that they do not properly represent the AR application in mind. Considering AR content creation as a design tool, designers can place the virtual content in tandem with the physical environment. This *in situ* approach to prototyping environments allows designers and domain experts to truly capture the idea in mind, but *in situ* AR prototyping tools do not see usage in application design today. 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* environment prototyping called the Mixed Reality Content Authoring Toolkit (MRCAT) and discuss 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. Designers can manipulate transforms by resizing, translating, rotating, annotating, and recoloring objects *in situ*. We discuss the design of MRCAT in the context of common guidelines for prototyping tools and of the results of a preliminary study exploring the needs of in situ prototyping tools compared to AR application prototyping in 2D environments. Through these efforts, we enumerate ways to increase usability of AR prototyping tools and key aspects of *in situ* environment design for AR prototyping tools to design for going forward.

2 Related Work

As AR technology becomes more accessible, domain experts will increasingly need to be part of the application design conversation. Participatory design allows for ideation within what is called a "third space"—a hybrid space between technologists and domain experts that includes ideas novel to both fields through co-creation, and a mutual learning and challenging of assumptions [29]. Within AR specifically, previous systems have prototyping tools for domain experts such as educators [21] and museum exhibit curators [43] where they edit video streams in a 2D AR browser. Generally, AR prototyping systems have either removed virtual content from physical context [27] or used some adapted form of sketching with ubiquitous materials like cardboard and glass [6]. This greatly limits the fidelity to what the end-user AR experience would look like. Alternatively, prototypes that offer higher fidelity to the intended experience often require programming knowledge in order to build [34]. With our work, we explore at how *in situ* prototyping can allow domain experts to build the application design in mind, directly in the target environment. In reviewing relevant literature, we consider AR content creation, particularly *in situ*, and user interfaces (UIs) that will make AR prototyping workflows feasible to a broad range of domain experts.

2.1 AR Content Creation

Research in AR content creation tools has explored different approaches to increase participation with novel AR technology without the need to program every aspect of the AR scene. Tools like DART [27] and ComposAR [36] allow users to build augment video and picture representations of the physical environment. Other content creation tools allow users to customize which 3D models are associated with different fiduciary markers in a tangible AR application [5, 23, 40]. With headsets having 6 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*.

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 a seamless blend of virtual content and the physical space. For example, SceneCTRL allows users to edit arrangements of physical and virtual both by placing new objects and visually deleting existing physical objects. [46]. 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 [47]. Work in AR museum presentation authoring explores scene editing on a web browser [40] and is then extended to use of a mobile phone to create and edit virtual models for a museum exhibit directly in the space [33]. 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*. 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.

Maybe come back to me.

2.2 AR Multimodal Interaction

UIs for prototyping tools require consideration of a number of tasks. Effective design of UIs for *in situ* AR prototyping is further complicated by the fact that there are not standard interaction metaphors and best practices for UI design. Integral to the success of AR prototyping tool is consideration how to enable fluid interactive design in AR. AR systems commonly make use of freehand gestures to manipulate object transforms [7, 15] since the metaphor to grab and manipulate a virtual object maps to manipulation of physical objects. Freehand gestures have also been used to annotate [10, 24], sketch [1, 44], navigate menu systems [12, 30] and update descriptive characteristics such as color [32]. Alternatives to gestural interaction include using mediating devices such as tangible markers [23], secondary tablet/phone displays [28, **?**] and video game controllers [41, 42]. This disparate exploration of different modalities for interaction in AR makes in difficult to pin down specific best practices when crafting an AR interface.

Research in multimodal interaction in AR 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 define the object manipulation [8, 11, 39]. 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 [19, 31] or to change the color or shape of an object [26]. These multimodal interactions can also provide *mutual disambiguation*, where input from multiple modalities probabilistically provides greater precision than either input on its own [22]. In AR prototyping, these multimodal approaches could provide greater accuracy and speed in interactions than individual modalities could achieve on their own.

The particular task can also guide the best ways to interact with a system. For example, picking particular items in a data visualization may be well-suited to gestural interaction while higher-level commands like creating a new data visualization would be well-suited to voice interaction [2]. The high agreement scores in elicitation of gestural interaction for translation, rotation and scaling suggest that freehand gestures are intuitive for transform manipulations [32]. 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 transforms [15, 7, 38, 8] and voice commands for descriptive characteristics [26, 25] and interface-level commands [39, 46].

3 Design Guidelines

With this work, we first consider design guidelines that will enable AR systems to go beyond efficient content creation and toward interactive AR environment prototyping. While prior work in 2D and in situ AR prototyping has primarily presented the ability edit 3D models, we consider how an *in situ* approach changes prototyping tool usage. MRCAT offers an extended suite of prototyping functionality, including directly placing/manipulating virtual objects and saving/loading scenes. In developing MRCAT, we synthesized design guidelines for *in situ* prototyping tools that extend traditional prototyping guidelines:

D1: Full Experience Prototyping. To effectively create design artifacts, prototyping tools should allow designers to capture the intended experience and different application designs [3]. AR prototyping tools should consider interactions of virtual and physical elements giving designers the ability to enumerate relationships and other information not covered by virtual models loaded into the environment. In MRCAT, we implement this guideline through combined model integration, transformation and text-based annotation (§4.2).

D2: Intuitive UI. Creating interactive AR applications requires disparate design tasks (e.g., positioning models, mocking interactions, annotating models). We guided interaction mappings with prior literature (§2.2), using freehand interactions for model manipulation, and voice commands for abstract operations such as deleting objects, changing color and saving (§4.2).

D3: Constrained Interactions. Users should clearly understand how their input will manipulate the environment. Prior AR tools achieve this by mapping the same gesture to different functionality [37]. Explicit mode switching—implemented via voice commands and menu options—ensures users know what manipulation they are performing (e.g. translation, rotation, scaling) (§4.2).

4 MRCAT Preliminary Design

In our exploration of AR prototyping, we built MRCAT to allow users to create and edit prototypes *in situ*. This system instantiates the design guidelines laid out in §3, providing full experience prototyping *(D1)*, an intuitive UI *(D2)* and constrained interactions *(D3)*. In this section, we discuss the preliminary design and implementation of MRCAT.

4.1 System Overview

MRCAT is a prototyping tool built for the MS HoloLens[1] that allows users to place and manipulate objects in the environment. Prior to running MRCAT, users can load custom 3D models into the project folder in order to tailor the 3D models to the specific domain for which they are designing. When MRCAT first starts a main menu appears that shows the functionality available to the user. MRCAT allows users to enable different modes that determine how their interaction will affect the selected objects. The modes available to users are "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 the associated voice command (i.e. "MRCAT `mode`"). Selecting menu options is done through the built-in gaze-tap gesture, where a cursor raycasted from the center of the user's gaze indicates which specifies which item to select with and a freehand tap gesture acts as a click. To engage with an object, the user selects that object with the same gaze-tap gesture, and subsequent interactions will affect that object. 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.

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 (Fig. 1). For example,

[1] https://docs.microsoft.com/en-us/hololens/hololens1-hardware

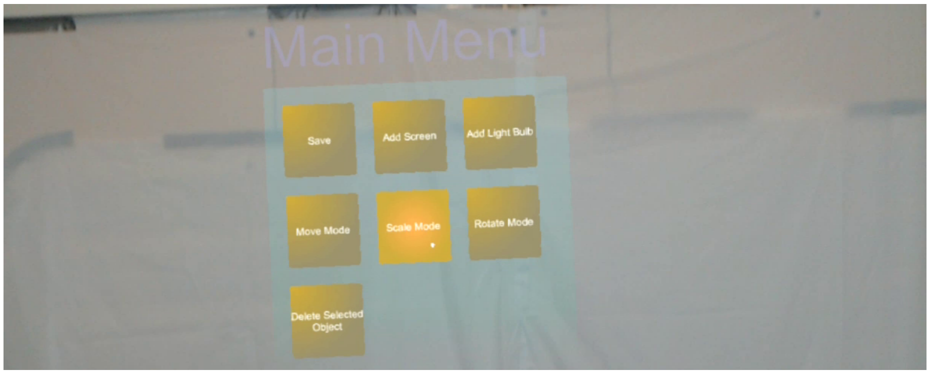

Fig. 1. MRCAT's main menu. Users can enter different interaction modes through the menu interface or through equivalent "MRCAT `mode`" commands. For example, the user can select "Scale mode" from the menu, or say "MRCAT Scale."

to enter rotation mode, for example, the user selects all objects they would like to rotate using the selection gesture. Those objects are then given a red border around them to indicate their selection. The user can then either give the voice command "MRCAT `mode`" to enter that mode (ex: "MRCAT Rotate") or select the corresponding menu option. This "MRCAT" initiation is similar to familiar voice-interaction with assistants such as Apple's Siri[2] and Amazon's Alexa[3] *(D2)*. Then the user performs the interaction to apply the change to all selected objects. All menus and notes situated in the environment are billboarded such that they always face the user, allowing the user to engage with UI elements from anywhere in the environment.

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 speaks a command that says "Add `item name`". The user can then control the placement of a virtual object by moving the object around using their gaze. The object sits 2 meters in front of the middle of the user's forward 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. If the object collides with another virtual object or physical surface such as a floor, wall or table, the object temporarily rests on the surface it collided with. The yellow sphere provides a visual cue that there has been a collision and that the user may need to move the object elsewhere (Fig. 2). Once the user is satisfied with an object's location, they select the object 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, circling it in a red outline. They can then drag the selected objects by moving their hands to drag the object to different points in the

[2] https://apple.com/siri/
[3] https://developer.amazon.com/en-US/alexa

environment. We employ hand gestural interaction for position refinement than for initial object positioning to allow for more precise object placement. That is, users are free to position objects in the environment through whatever trajectory makes sense, rather than having objects locked 2 meters from their forward gaze and needing to crane their neck in order to precisely move an object. To engage with objects, the user can select objects to toggle a red outline on and off. 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.

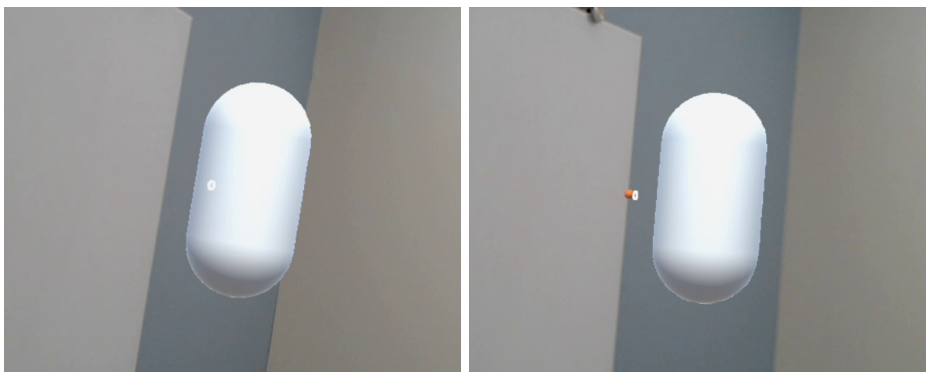

Fig. 2. When placing and moving objects, a small yellow sphere indicates attempted displacement. Here the user is trying to move the virtual capsule through a wall, but the capsule collides with and renders against the wall.

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 users initial hand movement is primarily along the X, Y or Z axis, and locks the object to rotate to one axis. If the initial hand displacement is along the X axis relative to the user, the object rotates about the Y axis. Similarly, hand displacement along the Y axis maps to rotation about the X axis, and hand displacement along the Z axis maps to rotation about the Z axis. 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 to change object rotation (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. Dis-

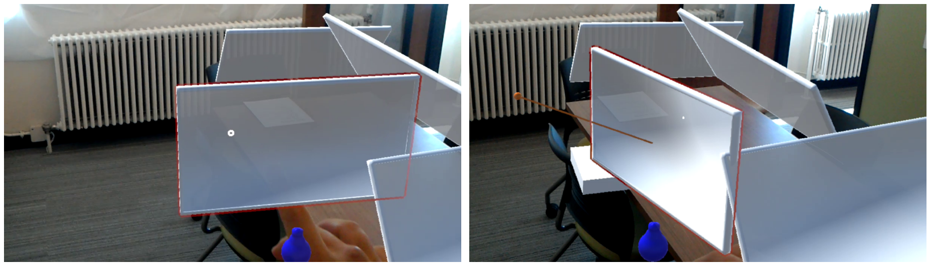

Fig. 3. User rotating a virtual screen about the Y axis via hand displacement along the Y axis. When the user begins dragging, a virtual joystick appears to give the user an interaction metaphor of pulling a joystick back and forth.

placement 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[4] and SketchUp[5]), 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 the material `material name`. For simplicity, we limited materials to the colors of the rainbow, black, white and ghost (for transparency).

Annotation: MRCAT allows users to textually annotate objects in the scene, such that it accurately represents the design in the user's mind *(D1)*. This is an important prototyping interaction as these annotations can indicate relationships between objects, fill in gaps where the 3D model design may fall short or as a means to provide feedback to prototypes *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). 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 seconds. The user says "MRCAT Post" to render a note that appears as a sticky-note style panel with a "Remove" button. The note then renders above the object, dynamically adjusts to its associated object's transform. To avoid occlusion, the note renders above its associated object if the user is looking down at the object and will render below the associated object if the user is looking up or directly at the object.

System-Level Interactions: Saving and loading arrangements is implemented through an audio input-output interface. Users can export prototyped scenes to

[4] https://unity.com
[5] https://sketchup.com

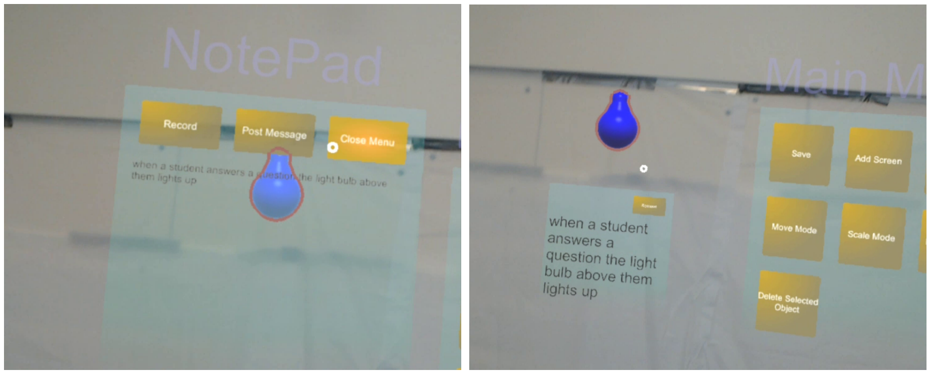

Fig. 4. Interface for recording user dictation as a note (left) and posting it to a virtual object (right). After pulling up the Notepad interface with the "MRCAT Note" command, users select the "Record" button to begin recording. Recording ends after two seconds of silence, at which point users can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

XML files using the command "MRCAT Save" 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.

4.3 Design Decisions

Multiple Object Manipulation: We see significant value in being able to selectively scale any subset of objects in an environment, and have included the ability to do so in MRCAT. The user selects multiple objects in any mode and the manipulation is applied to all objects. Given the relatively large amount of information participants needed to learn in order to perform the user study, we did not explicitly tell users about multi-object manipulation. In piloting, we found that the amount of functionality presented to participants was near the threshold of what is reasonable to expect of a participant in an hour long session. Additionally, due to physical fatigue of the heavy headset, we wanted to minimize training time for participants.

Dictation: Recording notes and saving environments requires user dictation to input freeform text. Due to the already prominent use of voice interaction and open nature of HMD-based text entry as a research area [45], 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-record, rather than editing individual words. However, since freeform text in this system should be relatively short, re-recording the intended text would be inexpensive.

Billboarding Menus: With our system, the only way to view contents of menus and notes is to look directly at them. That is to say, there is no hierarchy view as in Unity and many of the desktop-based prototyping solutions discussed in section Related Works. For this reason, at every frame the headset localizes itself within the environment, the normal vectors of the NotePad, the Main Menu, and the notes themselves adjust to face the user.

Portability: When the user saves a file, the scene saves as an XML file type, which contains all relevant information about Unity GameObjects created by the user such that MRCAT can load those files. Files are relatively small (1.6MB on average in the user study), and can be loaded into MRCAT's scene manager to populate the scene at runtime. By changing the headset camera prefabricated element (prefab), users can view the prototype in any headset. To verify this functionality, we tested out the ability to load a prototypes onto a desktop display and to a Gear VR display.

Though prototypes are best viewed in the HoloLens, as a blended mixed reality experience, future technological developments could make portability of a prototypes an important feature for improved collaboration. With 360 depth camera capture integrated with AR headsets, a high fidelity mesh of the physical environment could be included with the developed prototype. This capture of the physical environment alongside proposed virtual augmentations would allow remote users to view the prototype on a desktop computer or in VR. The ability to view high fidelity prototypes on different platforms would extend the benefits of *in situ* prototyping from improved co-located collaboration to also improve remote collaboration.

4.4 Desktop Prototyping

For our user study, we use a subset of functionality from Unity Game Engine to parallel functionality of MRCAT. Desktop content creation tools [36, 40] 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.

5 Evaluation

To evaluate MRCAT against a 2D alternative, we conducted a 2 (prototyping tools) x 2 (scenario) mixed factors user study with 14 participants (12M, 2F). Both involving prototyping the setup of a connected room of smart devices, an example use case, a new design constraint introduced, an alternate configuration, and a variation of the example use case, leaving the four tasks for each scenario freeform. Participants were given time to train with each tool, during which we walk them through the interactions and allow them to practice. We then let participants perform the scenerio's tasks, exiting the environment after giving each instruction. We administered a questionnaire after use of each tool and a comparisons and demographics questionnaire after the participant saw both tools.

5.1 Scenarios

We contrived two scenarios for participants to design, one with MRCAT, and one with the 2D prototyping alternative. In a smart conference room scenario, we gave participants tasks that require them to prototype a wirelessly connected conference room. We first asked participants to add primary and secondary displays to the room, and smart lightbulbs associated with each of the four people that can connect in this conference room. Participants were then asked to illustrate an example use case of meeting members connecting to the smart displays, using notes to explain where necessary. We then instructed participants to illustrate a different configuration to propose to administrators. Lastly as an alternate configuration, we ask participants to explore use of two displays of equal priority, rather than a primary and secondary.

In the learning room scenario, we gave participants tasks to prototype a room with a tabletop display that facilitates learning objectives. We first asked participants to design a tabletop display with lightbulbs associated with each student. We then explain that the system can be used for quiz questions, and ask participants to illustrate this example use case. We then introduce a new design constraint—that students should have tablets on the table in front of their seats. Lastly, we ask participants to create a new configuration with the display on the wall instead of the table.

5.2 Results

With our preliminary evaluation of MRCAT, we explored its ability to serve the needs of *in situ* prototyping. With the nature of the exploratory nature of the study tasks, we focused primarily on subjective metrics and user feedback, rather than objective metrics such as accuracy. We administered three questionnaires to participants—one after each prototyping tool and a comparisons questionnaire at the end of the study. These questionnaires all had Likert-scale questions and gave the opportunity for open-ended feedback.

Quantitative Measures Participants generally took longer with MRCAT ($\mu =$ 28.3 min, $\sigma = 11.66$ min) than with the 2D prototyping tool ($\mu = 25.78$ min, $\sigma = 10.75$ min). Participants reported higher System Usability Score (SUS) for the 2D prototyping tool ($\mu = 72.1$) than for MRCAT ($\mu = 51.9$). Participants also typically preferred the 2D tool when asked to directly compare the two for particular tasks. Specifically, they preferred the 2D prototyping tool for object placement, translation, rotation, scaling and annotating, while preferring MRCAT only for object recoloring.

Open Feedback While the quantitative measures indicated strong preference toward the 2D prototyping tool, open feedback illuminated some possible reasons for this preference and the potential for *in situ* prototyping going forward.

Open feedback elicited key benefits of prototyping an interactive environment *in situ*. One benefit was with editing and navigating the proposed environment in parallel. This manifested as participants 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 "see what [they] built from multiple angles easily." *(P3)*. Another aspect of *in situ* prototyping was "working the actual room was useful...to get a better sense of scale" *(P1)* and "getting a true feel for environment" (P5). In line with this heightened sense of scale was perhaps the most promising feedback regarding *in situ* prototyping—the ability to prototype to higher fidelity. This aspect was best summarized by participants saying "it definitely allowed for the user to better visualize how the room would look in reality, which is a pretty significant advantage over [the 2D tool]. 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 [they] 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 called out as problematic by 6 out of 14 participants, more than any other operation. We consider object transform manipulations as the most pressing issue with MRCAT and a point of focus for iteration.

Another salient theme from open feedback was frustration with the headset itself. Likely impacting frustration and usability, a 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 should be mitigated with future AR headsets, we note that these hardware limitations are worth considering in design of AR prototyping tools going forward.

5.3 MRCAT Iteration

Based on this feedback from participants, we came to understand that the trade-offs of *in situ* prototyping against decontextualized prototyping on a 2D system were likely conflated by hardware and UI issues. In our iteration of MRCAT's UI, we consider issues participants had with understanding the system's current state and how their input is affecting and will affect the prototype. This is in response to the sentiment toward the UI that "the interface is somewhat unresponsive" and "a bit choppy" *(P10)*. While we cannot necessarily address the issue of imprecision in freehand gestures, we iterate on MRCAT's UI design in an attempt to build a more fluid user experience. In light of the feedback on transform manipulations with MRCAT, we add an additional design guideline:

D4: Visual & Continuous Transform Manipulation to supplement the guidelines from §**??**. With this guideline we prioritize providing the user with visual feedback to manage transform manipulation continuously, rather than explicitly switching between "Rotation", "Translation" and "Scale" modes. Despite this explicit mode switching effectively mapping multiple functions to mouse dragging in 2D, visual cues can enable a more fluid experience in 3D.

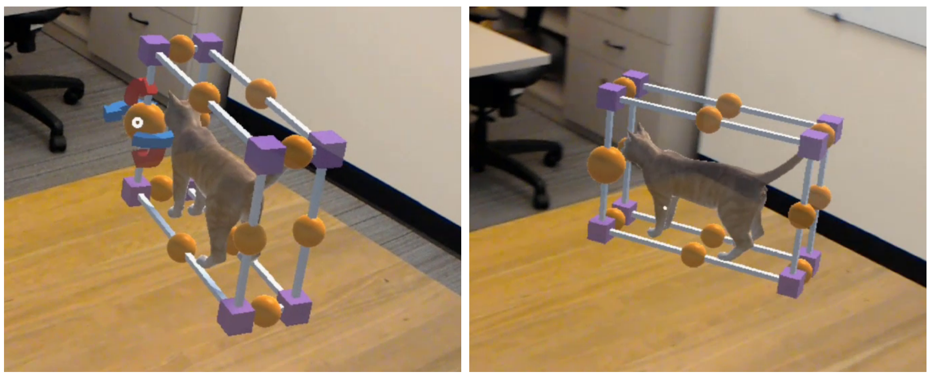

Fig. 5. Updated MRCAT interface using wireframe cubes for transform manipulation. This allows for continuous translation (dragging the object), scaling (dragging a corner cube), and rotation (dragging a sphere on an edge). When the user first engages with a rotation sphere, two sets of arrows indicate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

To address this, we implement a 3D wireframe cube (Fig. 5), as has been done in previous AR systems [9, **?**]. Selecting an object with the gaze-tap toggles the wireframe cube around that object. As did the outline around objects in the previous iteration of MRCAT, the wireframe cube indicates engagement with objects for transform manipulations, color changes and annotations. Hovering the gaze cursor on the object itself and dragging 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 sides allows the user to rotate the selected objects with one degree of freedom at a time. As with the "Rotation" mode in the previous iteration of MRCAT, initial hand displacement determines which direction the object is to rotate. Engaging with each of the wireframe's 12 spheres provides users with 2 possible rotation directions, which render when the user engages with the sphere. Depending on the location of the spheres on the wireframe, the sphere allows the user to manipulate along either the X- and Y-axes or the X- and Z-axes.

To mitigate issues with the HoloLens' gaze-drag gesture when manipulating transforms, we implement visual feedback to indicate lost hand-tracking. An inherent limitation of freehand gestures relative to haptic interfaces 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 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. This 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.

6 Case Study: Musuem Exhibit Prototyping

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

Long term, we anticipate that AR prototyping will see broad usage in a number of domains. As HMDs become increasingly accessible, the need to prototype interactive environments will emerge as an important component of AR application design. As it stands though, AR does not see widespread usage across domains. In this section, we discuss museum exhibit curation as an exemplary domain that could see more immediate benefits to existing prototyping workflows from *in situ* AR prototyping. We discuss museum exhibit curation as a case study, scaffolded with a formal interview with a curator at an on-campus Natural History museum. Considering museum exhibit curation as an example, we identify aspects that make it sensible for *in situ* AR prototyping and how it and other domains could pave the way for novel AR prototyping workflows.

Museum exhibit curation, or the design and construction of installations in a museum, presents an interesting case study for *in situ* AR prototyping. In conversation with a local museum exhibit curator at the University of Colorado

Natural History Museum, we gained knowledge of existing ideation, collaboration and prototyping workflows. Here we summarize findings from this interview about existing ideation, prototyping and collaboration workflows in museum exhibit curation. In this 90 minute interview, we began with a discussion of existing workflows in museum exhibit curation, followed by a brief demonstration of MRCAT, and how *in situ* AR prototyping may supplement existing prototyping practices. Most of the discussion was centered around an in-progress exhibit at the Mesa Verde National Park heritage center, which focuses on use depicting Ancestral Pueblo life.

Ideation: As with any design project, museum exhibit curation requires an initial ideation phase before resources are poured into implementation of the exhibit. Artifacts from this ideation phase are typically in the form of post-it notes, which then required clustering, organizing and digitizing. The museum team typically wants to "test out some [options] and choose a few different ways to graphically interpret". Testing these possible configurations could get at questions of intended experience: "Is it a hands-on experiences? Is it a video experience? Is it a sensory experience? Let's come up with a few different visitor experiences and try them out in a low-cost way." With these early-stage questions in mind, it becomes important test and iterate on these ideas.

Building Prototypes: Within the domain of curating museum exhibits, the greatest benefit of *In situ* AR prototyping is "trying to communicate the vision in [the curator's] head to people who may not perceive it." Currently, artifacts of the prototyping stage are typically in the form of "pre-schematics" which are a combination of sketches, blueprints and text. In some instances, the group will "print stuff cheaply, using cardboard before [they] start designing and fabricating fully finished exhibits". Relative to this approach, *in situ* AR prototyping would allow for higher fidelity to what the end experience may look like.

Collaboration: Collaborating on museum exhibits requires input from a number of key stakeholders. In the described project, this included tribal leaders, University students, teachers, anthropologists, and a core group of museum employees including our interviewee. While some collaboration is in person, a good bit is done over video chat, and in both scenarios, the higher fidelity of prototypes built in AR would scaffold conversations better than sketches and descriptions would. Even with a greater initial investment of time, our interviewee mentioned that "if they are trying to get 60 stakeholder to understand the vision that [they are] thinking of, it would take more time to create those visions but there might be real value to using AR/VR to communicate what the full space might feel like."

With these potential improvements to ideation, prototype building and collaboration workflows, *in situ* AR prototyping tools such as MRCAT could see usage in museum exhibit prototyping, even before AR becomes a ubiquitous display medium.

7 Discussion

Our work on MRCAT explores how an *in situ* prototyping approach can positively impact design workflows both when AR headsets become more ubiquitous and today, in targeted domains that could benefit from improved environment prototyping. In this section, we discuss domains that could see immediate usage of AR prototyping tools, and important future work for AR prototyping tools be be successful.

7.1 Key Aspects for AR Prototyping

Our interview identified ways in which museum exhibit serves as an exemplary domain where *in situ* AR prototyping could immediately benefit design workflows. In this subsection, we generalize two key aspects of museum exhibit curation that make it well-suited to an *in situ* approach to prototyping, With these aspects enumerated, future AR prototyping systems can design for particular objectives, and future work can identify other domains for collaboration and research.

Constraints of the physical environment: Perhaps the most important aspect of a domain that would make it sensible for *in situ* AR prototyping is the importance of environmental constraints. An idea that proposes changes and augmentations in tandem with the physical environment itself is better represented by a prototype built *in situ* than one made through hand-drawn sketches or other 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.

In cases where the look, feel and scale of the augmentations in the target space is particularly important to the success of the prototyped environment, *in situ* AR prototyping can provide the means to design for this interplay of the physical and virtual environment. In the preliminary study, this manifested as the ability directly manipulate the mixed reality environment. For example, users would place virtual lightbulbs directly in the physical space and resize virtual screens such that they made sense on the physical tables and walls. In curating museums, this was the ability to test out configurations that would make sense for the allotted space. This seamless blend of physical and virtual content is effectively represented by prototypes built directly in the target environment.

Co-located and remote collaboration: In situ prototyping of environments can enable new ways of collaboration in environment design. The ability to save and load XML representations of scenes within the target environment enables multiple designers to iterate on prototypes by adding, reconfiguring and annotating saved arrangements. Extending MRCAT to enable multiple AR headsets to view the same scene would allow for synchronous, multi-user design and review within the environment.

Though out of scope for the preliminary study, collaboration was identified as a particularly useful benefit of AR prototyping in our interview. The higher fidelity prototype built *in situ* would provide collaborators a better sense of the idea in the 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 to with depth camera capture, prototypes could be exported with a representation of the physical environment such that remote viewers in VR or on a desktop could have a comparable experience.

Other Possible Domains for AR Prototyping

While we focus in depth on museum exhibit curation as a domain of interest, future research in AR prototyping can focus on a number of other domains that could see immediate benefits from an *in situ* prototyping workflow. We call out these specific domains as additional testing grounds for continued AR prototyping research.

Internet of Things (IoT) applications: With increased usage of connected smart devices, AR prototyping would allow designers to propose configurations of integrated smart environments within the target environment. Use of AR as a user interface for connected smart environments has been explored in prior work [13, 20]. As IoT-enabled devices become smaller and interactive components are shrunken or removed from the devices themselves, AR can provide the visual UI needed to interact with smart devices in the connected environment. As is done in the preliminary user study of MRCAT, designers can pick and place and manipulate models of smart devices and use annotation to help define relationships and usage scenarios.

Educational Content: Research on use of AR for delivery of situated educational content has indicated benefits of mobile and immersive AR displays, broadly categorized by Santos et al. as real world annotation, contextual visualization and vision-haptic visualization [35]. While prior systems show promise for AR as an educational display medium [4, 18, 14], continued use of AR in education should include educators in the iterative design process of the system and the lesson plan. By allowing educators to ideate and create in the classroom, AR lesson plans can better incorporate the physical environment.

Theater Set Design: Theater sets require design of physical interactive components that comply with the constraints of the particular stage. Creative use of the stage, lights and set-pieces allows directors to deliver unique performances. This creative process could see benefits from *in situ* prototyping on the stage itself, where directors and other stakeholders could collaborate on sets for each scene of the performance.

7.2 Usability and Longitudinal Study

Our preliminary study of MRCAT against a 2D prototyping tool revealed the need for thorough consideration of effective multimodal interaction in AR. For transform manipulations in particular, we see constrained interactions *(D3)* as a particularly important guideline when designing freehand gestural interaction. After our preliminary study, we prioritized increased visual feedback *(D4)*. Use of the 3D wireframe cube rather than explicit mode switching allows for both more fluid object manipulation and increased understanding of how exactly the performed gesture will affect virtual objects. Continued study of these interactions will serve as the basis for an efficient interface for MRCAT and other AR prototyping tools.

Another important consideration for multimodal interaction in AR is how particular modalities map to different tasks. We designed our interface based on mappings commonly used in prior literature (§2.2), but continued study of which tasks will be better suited to different interaction modalities will provide more grounded justification for design decisions. For example, while freehand gestures are perhaps most the commonly used modality for transform manipulations [7, 15, 8], this could be evaluated against a multimodal gaze + voice interface [19, 31] or a video game controller [**?**,42] to empirically ground the decision. We employed voice for text entry due to our prioritization of speed over accuracy (a trade-off identified in prior literature [17]). This decision made sense, as participants used 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 in put 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. As future research on best practices for AR interactions develops, AR prototyping tools should integrate the results of empirical interaction studies into multimodal user interfaces.

Longitudinal study of usage of AR prototyping tools will be critical to widespread adoption of *in situ* AR prototyping workflows. In our discussion of museum exhibit curation, there are clear fits between the described needs for design workflows and the opportunities presented by *in situ* AR prototyping. However, integration of MRCAT into these design workflows and observation of usage will validate long-term usage and identify further design guidelines for AR prototyping tools. Longitudinal study of environment design with AR prototyping tools will further establish a roadmap for successful design of AR prototyping tools.

Among other interesting design considerations that could be elicited from longitudinal study is how to could extend the breadth of prototyping features available to designers. With continued use and increased comfort with MRCAT, a next step for AR prototyping tools is to enable designers to prototype interactive components with the environment. In the preliminary study, users employed annotations to describe the interactivity of the environment and in Figure 6, we demonstrate how virtual models can be used to visually depict interactivity. However, extending MRCAT to allow designers to mock the interactions and build fully interactive prototypes could allow designers to test design of different interactions and even conduct user studies with interactive AR prototypes.

8 Conclusion

In situ prototyping provides designers the ability to build and ideate on interactive environments within the target environment. With our work on MRCAT, we propose and evaluate a tool for prototyping interactive environments *in situ*, identifying guidelines for design of such prototyping tools. Through a prelimniary user study, we identify tradeoffs between *in situ* and decontextualized 2D prototyping. We also consider domains that could see immediate improvements to existing design workflows, and thus should be targeted in future work. With our exploration of *in situ* prototyping, we provide a roadmap for AR prototyping research, such that AR prototyping tools could see increased usage from designers and domain experts toward greater participation in design of interactive, room-scale AR applications.

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, NY, USA (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. p. 15831594. Association for Computing Machinery, New York, NY, USA (2017)
5. Billinghurst, M., Kato, H., Poupyrev, I.: The magicbook - moving seamlessly between reality and virtuality. IEEE Computer Graphics and Applications **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. p. 20372046. Association for Computing Machinery, New York, NY, USA (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. p. 212221. Association for Computing Machinery, New York, NY, USA (2004)
8. Chaconas, N., Hllerer, T.: An evaluation of bimanual gestures on the microsoft hololens. In: 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). pp. 1–8 (2018)
9. Chakraborty, A., Gross, R., McIntee, S., Hong, K.W., Lee, J.Y., St. Amant, R.: Captive: A cube with augmented physical tools. In: CHI 14 Extended Abstracts on Human Factors in Computing Systems. p. 13151320. Association for Computing Machinery, New York, NY, USA (2014)

10. Chang, Y.S., Nuernberger, B., Luan, B., Hllerer, T.: Evaluating gesture-based augmented reality annotation. In: 2017 IEEE Symposium on 3D User Interfaces (3DUI). pp. 182–185 (2017)
11. Chang, Y.S., Nuernberger, B., Luan, B., Hllerer, T., O'Donovan, J.: Gesture-based augmented reality annotation. In: 2017 IEEE Virtual Reality (VR). pp. 469–470 (2017)
12. Dachselt, R., Hbner, A.: Three-dimensional menus: A survey and taxonomy. Computers & Graphics **31**(1), 53 – 65 (2007)
13. 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)
14. Giraudeau, P., Olry, A., Roo, J.S., Fleck, S., Bertolo, D., Vivian, R., Hachet, M.: 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)
15. Ha, T., Feiner, S., Woo, W.: Wearhand: Head-worn, rgb-d camera-based, bare-hand user interface with visually enhanced depth perception. In: 2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 219–228 (2014)
16. HALL, R.R.: Prototyping for usability of new technology. International Journal of Human-Computer Studies **55**(4), 485 – 501 (2001)
17. 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. p. 156163. Association for Computing Machinery, New York, NY, USA (2012)
18. Ibez, M.B., ngela Di Serio, Villarn, D., Kloos, C.D.: Experimenting with electromagnetism using augmented reality: Impact on flow student experience and educational effectiveness. Computers & Education **71**, 1 – 13 (2014)
19. Irawati, S., Green, S., Billinghurst, M., Duenser, A., Ko, H.: "move the couch where?" : developing an augmented reality multimodal interface. In: 2006 IEEE/ACM International Symposium on Mixed and Augmented Reality. pp. 183–186 (2006)
20. Jahn, M., Jentsch, M., Prause, C.R., Pramudianto, F., Al-Akkad, A., Reiners, R.: The energy aware smart home. In: 2010 5th International Conference on Future Information Technology. pp. 1–8 (May 2010)
21. Jee, H.K., Lim, S., Youn, J., Lee, J.: An augmented reality-based authoring tool for e-learning applications **68**(2) (2014)
22. Kaiser, E., Olwal, A., McGee, D., Benko, H., Corradini, A., Li, X., Cohen, P., Feiner, S.: Mutual disambiguation of 3d multimodal interaction in augmented and virtual reality. In: Proceedings of the 5th International Conference on Multimodal Interfaces. p. 1219. Association for Computing Machinery, New York, NY, USA (2003)
23. 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)
24. 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, NY, USA (2017)
25. Lee, M., Billinghurst, M.: A wizard of oz study for an ar multimodal interface. In: Proceedings of the 10th International Conference on Multimodal Interfaces. p. 249256. Association for Computing Machinery, New York, NY, USA (2008)

26. 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)
27. 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)
28. Millette, A., McGuffin, M.J.: Dualcad: Integrating augmented reality with a desktop gui and smartphone interaction. In: 2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct). pp. 21–26 (2016)
29. Muller, M.J.: The human-computer interaction handbook. chap. Participatory Design: The Third Space in HCI, pp. 1051–1068. L. Erlbaum Associates Inc., Hillsdale, NJ, USA (2003)
30. Ni, T., Bowman, D.A., North, C., McMahan, R.P.: Design and evaluation of freehand menu selection interfaces using tilt and pinch gestures. International Journal of Human-Computer Studies **69**(9), 551 – 562 (2011)
31. 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: 2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 73–82 (2014)
32. 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.) Human-Computer Interaction – INTERACT 2013. pp. 282–299. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
33. Rumiński, D., Walczak, K.: Creation of interactive ar content on mobile devices. In: Abramowicz, W. (ed.) Business Information Systems Workshops. pp. 258–269. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
34. 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. p. 221230. Association for Computing Machinery, New York, NY, USA (2012)
35. Santos, M.E.C., Chen, A., Taketomi, T., Yamamoto, G., Miyazaki, J., Kato, H.: Augmented reality learning experiences: Survey of prototype design and evaluation. IEEE Transactions on Learning Technologies **7**(1), 38–56 (2014)
36. Seichter, H., Looser, J., Billinghurst, M.: Composar: An intuitive tool for authoring ar applications. In: 2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality. pp. 177–178 (2008)
37. 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, CAN (2019)
38. 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)
39. Turini, G., Condino, S., Parchi, P.D., Viglialoro, R.M., Piolanti, N., Gesi, M., Ferrari, M., Ferrari, V.: A microsoft hololens mixed reality surgical simulator for patient-specific hip arthroplasty training. In: De Paolis, L.T., Bourdot, P. (eds.) Augmented Reality, Virtual Reality, and Computer Graphics. pp. 201–210. Springer International Publishing, Cham (2018)

40. Walczak, K., Wojciechowski, R.: Dynamic creation of interactive mixed reality presentations. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology. p. 167176. Association for Computing Machinery, New York, NY, USA (2005)
41. Walker, M.E., Hedayati, H., Szafir, D.: Robot teleoperation with augmented reality virtual surrogates. In: 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI). pp. 202–210 (2019)
42. Whitlock, M., Harnner, E., Brubaker, J.R., Kane, S., Szafir, D.A.: Interacting with distant objects in augmented reality. In: 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). pp. 41–48 (2018)
43. 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, NY, USA (2004)
44. Wolf, D., Dudley, J.J., Kristensson, P.O.: Performance envelopes of in-air direct and smartwatch indirect control for head-mounted augmented reality. In: 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). pp. 347–354 (2018)
45. 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. p. 44794488. Association for Computing Machinery, New York, NY, USA (2017)
46. 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, NY, USA (2017)
47. 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)