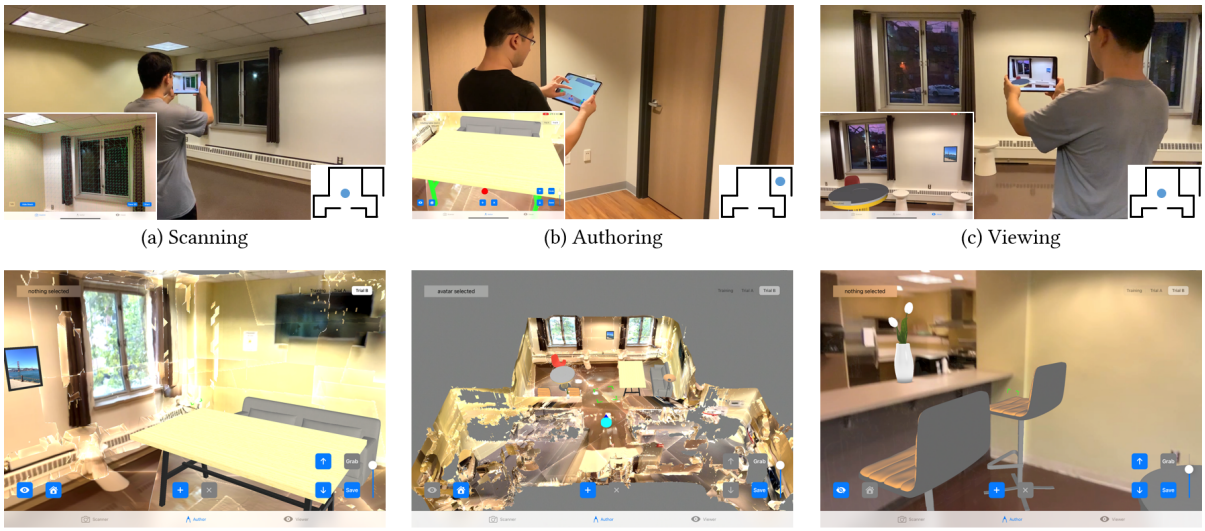
DistanciAR: Authoring Site-Specific Augmented Reality Experiences for Remote Environments

Zeyu Wang
Yale University
zeyu.wang@yale.edu

Cuong Nguyen
Adobe Research
cunguyen@adobe.com

Paul Asente
Adobe Research
asente@adobe.com

Julie Dorsey
Yale University
julie.dorsey@yale.edu



(a) Scanning (b) Authoring (c) Viewing

(d) PORTAL mode (e) DOLLHOUSE mode (f) PEEK mode

Figure 1: The DistanciAR pipeline (top) and the three modes in its authoring interface. (a) Capturing a remote environment with a LiDAR-equipped tablet. (b) Authoring an AR experience from another location. (c) Viewing the created AR experience on site. The screenshot in the bottom left corner shows what the tablet user sees. The floor plan in the bottom right corner indicates the location of the tablet user. (d) PORTAL mode that simulates the experience of using in-situ AR authoring tools. (e) DOLLHOUSE mode where the author can see a bird's-eye view. (f) PEEK mode that retrieves a captured image based on the pose of the AR camera. Red chair, picture, and vase models: © 2020 Apple Inc.

ABSTRACT

Most augmented reality (AR) authoring tools only support the author's current environment, but designers often need to create site-specific experiences for a different environment. We propose DistanciAR, a novel tablet-based workflow for remote AR authoring. Our baseline solution involves three steps. A remote environment is captured by a camera with LiDAR; then, the author creates an AR experience from a different location using AR interactions; finally, a remote viewer consumes the AR content on site. A formative study revealed understanding and navigating the remote space as key challenges with this solution. We improved the authoring interface by adding two novel modes: DOLLHOUSE, which renders a bird's-eye view, and PEEK, which creates photorealistic composite images using captured images. A second study compared this improved system with the baseline, and participants reported that the new modes made it easier to understand and navigate the remote scene.

CCS CONCEPTS

• **Human-centered computing** → **Mixed / augmented reality**.

KEYWORDS

augmented reality, remote authoring, spatial design, 3D scanning

ACM Reference Format:
Zeyu Wang, Cuong Nguyen, Paul Asente, and Julie Dorsey. 2021. DistanciAR: Authoring Site-Specific Augmented Reality Experiences for Remote Environments. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3411764.3445552

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05…$15.00
https://doi.org/10.1145/3411764.3445552

1 INTRODUCTION

With advances in devices and algorithms, augmented reality (AR) has become a popular and powerful tool in the design process.

Current AR authoring applications let designers author AR experiences for their current environment. However, with the modern workforce becoming more globalized and time-distributed, designers may need to create site-specific AR experiences from another location or at a different time. Creating AR content for a remote environment is difficult because designers do not have immediate access to its context. For example, interior designers can use AR tools when they are in the client's space to decorate, but after they leave, it becomes difficult to adjust the design and assess how well it fits the space. Better support for remote AR authoring would be useful, especially during a global pandemic, since it would allow creating AR experiences regardless of the physical location.

When AR is created in situ, authors can see, understand, and navigate the physical environment around them. They can also interact with the AR graphics rendered in that space. These affordances are critical to AR design because it lets authors integrate their creations with the physical environment [14]. More importantly, it lets authors experience the AR as the consumers will. When authoring from another location, authors are limited to desktop tools to create and test new AR experiences. Some tools, like Reality Composer [4], let authors preview in AR, but they support only generic contexts like tabletops, walls, and floors. Authors cannot create site-specific experiences that use a remote space's full context, including its size, layout, and contents. Other tools, like Unity [34], can support the full context, but do not let authors experience their creation as if they are in the remote site, so they cannot discover and correct usability issues. The key question that drives our research is how to bridge this gap. We want to let authors understand and use the full context of a remote space and to experience their content the same way as the final consumers.

We propose DistanciAR, a novel tablet-based workflow for authoring site-specific remote AR experiences. With DistanciAR a local user captures a 3D scan of the remote environment, then an author performs AR authoring tasks with that scan as if they were in the remote space. To investigate how well DistanciAR supports remote AR authoring, we adopted an iterative development approach. We first developed a baseline version of DistanciAR with three steps. First, a person captures the remote environment using an iPad Pro with a LiDAR scanner (Fig. 1a). The capture consists of a 3D mesh, a video, and AR tracking data at each frame. An author then creates an AR experience for the remote environment using our authoring interface in another location (Fig. 1b). The author can see, navigate, and interact with a 3D rendering of the remote space using standard AR interactions such as moving the device to change viewpoint and selecting objects using ray casting. Essentially, the authoring interface simulates the experience of AR in the remote space, letting the author design for the remote space without actually being there. Once authoring is finished, a remote viewer can consume the created AR experience on site (Fig. 1c).

We conducted a formative study to gather initial feedback from professionals. We found that our baseline system supported remote AR authoring well, but the authoring interface needed improvements to help the authors better understand and navigate the 3D scan of the remote environment.

Based on our findings, we developed an improved version of DistanciAR by adding two novel modes: Dollhouse (Fig. 1e) and Peek (Fig. 1f). Dollhouse mode lets the author transition from a first-person view to a bird's-eye view, facilitating an overall understanding of the remote space without requiring a large physical space. The author can add virtual objects to the dollhouse, view the design from different angles, and specify the position and orientation of a virtual avatar for teleportation. Peek mode addresses limitations in 3D reconstruction quality by replacing the 3D rendering with a photorealistic composite of the virtual objects and a frame captured during the scan. It chooses the frame by maximizing overlap between the portion of the mesh visible in the current AR camera pose and that visible in the recorded camera pose. The author can move the iPad to view a series of captured frames and validate the design against the real environment. We also improved the baseline interface and call this version Portal mode (Fig. 1d).

We conducted a user study to evaluate the effectiveness of Dollhouse and Peek modes and found that participants preferred the full version offering all three modes to a basic version offering only Portal mode. Participants found it significantly easier to understand and navigate the remote environment in the full version. The three modes were used in complementary ways. Dollhouse mode was used mainly for large-scale tasks such as navigating the space and creating an initial layout. In contrast, Portal and Peek mode, which provide an AR-like view of the scene, were used more for fine-scale adjustment. Peek mode was used to validate the design against the real environment and finalize the AR experience.

In summary, we demonstrated that DistanciAR is an effective workflow for remote AR authoring and that adding Dollhouse and Peek mode further improved the authoring experience. They effectively mitigate the disconnect between the authoring context and the experience context. Our contributions are as follows:

- A novel tablet-based workflow for remote AR authoring that involves three steps: scanning, authoring, and viewing. Our integrated workflow lets the author create site-specific AR content for a remote environment through an intuitive authoring experience close to in-situ authoring tools and the consumer's viewing experience.
- An improved authoring interface with two new modes Dollhouse and Peek, informed by challenges identified in a formative study. Dollhouse shows a bird's-eye view and Peek creates photorealistic images of the remote AR environment. These modes facilitate navigation and spatial understanding and mitigate the disconnect between the remote space and the authoring space.
- A user study to evaluate the improved authoring system. Our study shows the benefits of the improved system over the baseline solution and reveals complementary usage patterns for Portal, Dollhouse, and Peek modes.

2 RELATED WORK

Authoring systems for AR. Mobile devices that are capable of AR sensing and rendering, such as head-mounted displays (HMD), tablets, and smartphones, are becoming more available and affordable. The availability of consumer AR devices has led to the emergence of AR authoring tools in both research and industry [7]. Current tools are being used either on an AR device or a desktop computer, neither of which can support contextual, site-specific

remote AR authoring. AR tools such as Adobe Aero [2] and CAPturAR [37] let authors create directly in AR. The authoring activity is contextual, but the author cannot continue after leaving the environment. Desktop tools such as DART [22], Unity [34], Spark AR Studio [13], Reality Composer [4], and Volumetric Capture [25] support content creation for a remote location. However, it is difficult to design AR *for* a particular physical environment if the author does not have immediate access to it. Without seeing the physical environment during the design process, the author might make mistakes. Their AR content might be misplaced or misaligned upon deployment. Worse, usability issues that could arise when testing the AR experience in the target environment cannot be fixed immediately. These hurdles create a tedious feedback loop for AR design, hinder creativity, and prevent good decision making.

DistanciAR combines the benefits of mobile and desktop AR authoring tools. Users can create AR experiences within a 3D capture of an environment. The capture serves as a proxy for the real environment and provides spatial context for AR design including room layout, ceiling height, and furniture arrangements. Users can design in context without being tied to any specific physical location.

Environment capture in Mixed Reality. Capturing contextual information of an environment and showing it to an author can assist the design process. A recent user study on collaborative AR with the Blocks system [14] pointed out that the spatial context of a physical environment plays an important role in AR creation. Several study participants adapted their AR design to match or interact with the physical environment even without explicit instruction.

Advances in scanning technology [8, 27] have made it easier to create and use a 3D model of an environment for authoring. Previous work on 2D desktop interfaces has explored using 3D reconstruction to assist in architectural sketching [24] and 3D prototyping [18]. A prominent direction in mixed reality research is to capture and manipulate the 3D reconstruction while the user is wearing an HMD to alter the user's perception. For example, Remixed Reality [19] lets users freeze time, erase objects, and teleport to viewpoints, none of which are normally possible in real life. Another example is SceneCtrl [41], which lets users erase objects from the physical scene. Capturing 3D data of a remote environment also has applications in remote instruction [30, 33, 35] and telepresence [16, 23]. These systems focus on visualizing remote users' activities in the current environment and facilitating effective communication among users. DistanciAR similarly lets a user interact with a 3D reconstruction of the environment. However, our work is different in that we use the 3D captured data to help users understand and navigate a virtual remote space. Supporting these tasks makes DistanciAR more suitable for remote AR authoring.

Image-based capture, such as photos and videos, has also been explored in AR authoring. Freeze-Frame [15] and AniCode [38] let users take a photo of a space and add AR annotations and animations. Pronto [17] extended this idea to video, letting a user capture a video to mock up AR experiences. DistanciAR adds the new use case of remote AR authoring to this line of research. Our capture is richer, containing both a video and a 3D reconstruction. The former provides a faithful visual representation of the scene, while the latter provides full spatial context.

Navigation in AR. AR navigation is typically done by moving the AR device. The AR camera updates based on the device's movement and tracking. Boom Chameleon [36] lets users similarly navigate a rendered 3D environment. The AR device is used as a *"window on a virtual world."* DistanciAR is similar in that both let users use an AR device to view and navigate a 3D scene, but our system focuses on 3D reconstructions of real-world scenes, which are usually larger in scale and demand more navigation effort from users. To augment this form of navigation, we provide Dollhouse mode, which shows a 3D map of the environment similar to World in Miniature [31] and Dollhouse VR [32]. Dollhouse mode extends this concept to AR. It helps users navigate a large 3D space while continuing their AR design activities.

Exploring photo collections in 3D. Most VR and AR systems rely on a 3D model or reconstruction. While a 3D environment supports viewing from an arbitrary position, it is challenging to construct a high-quality environment using a consumer device for scanning. Even with state-of-the-art LiDAR-equipped devices, the capture often has areas with coarse geometry and appearance. LiDAR scanning also has difficulty capturing glossy, specular materials. A structured browsing experience using captured images can convey the environment more clearly. Using photogrammetry, Snavely et al. [29] constructed a point cloud of a site from images collected from the internet, enabling Photo Tourism [28], which lets people browse images with a 3D virtual camera. The desire to combine a 3D rendering with captured images also motivated Immersive Trip Reports [9], which aligns photographs with a digital terrain model and presents them in an immersive experience. While neither system was designed for AR authoring, they inspired Peek mode, which retrieves a captured image that overlaps most with the view from the pose of the AR camera. Peek mode lets the author navigate freely and view captured images while retaining the feeling of using AR. This makes it easier to refine the design when the environment capture is of low quality.

3 BASELINE SOLUTION

The goal of our baseline solution was to create an end-to-end integrated solution in which the authoring experience was as close to an in-situ AR authoring experience as possible. It involves three steps. First, someone in a remote site captures the environment using a scanner; then, the author creates an AR experience from a different location using AR-inspired interactions; finally, a remote viewer consumes the experience at the original site. Our implementation uses an iPad Pro 2020 model because it is a powerful, readily-available tablet with a LiDAR scanner that enables state-of-the-art results for a consumer device. All three steps are incorporated in an iOS application implemented in Swift with ARKit [3], RealityKit [5], and SceneKit [6] for tracking, scanning, and rendering.

3.1 Scanning Interface

Our workflow relies on a captured 3D model of the remote environment. The scanning interface uses the LiDAR scanner on the iPad to construct a 3D mesh while recording a video with the RGB camera. The system constructs the mesh in real time, overlaying it on the viewed scene in wireframe, and the mesh expands as the user points the iPad at different areas. It concludes when the user has scanned the full environment. Scanning results in a set of files that describe the environment, including a 3D model, a recorded

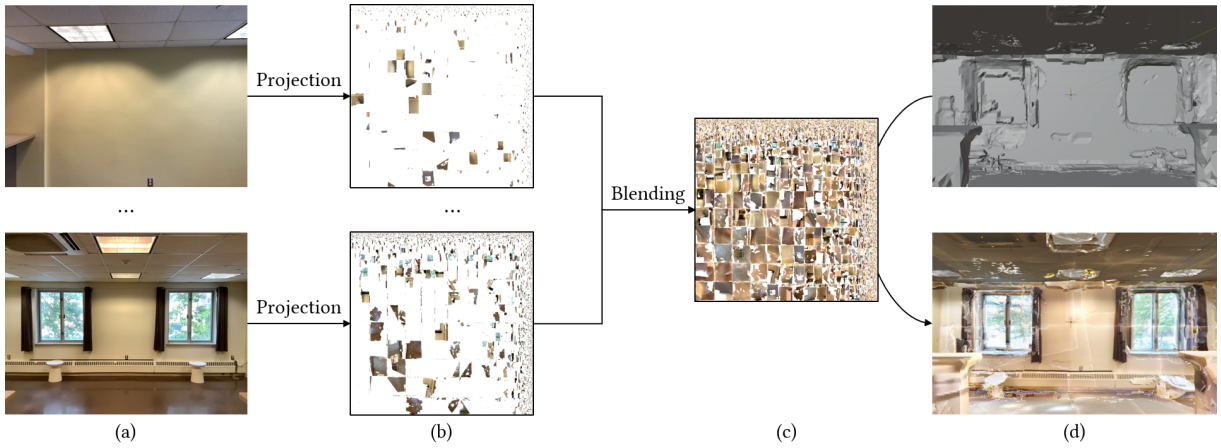


Figure 2: Texturing the captured mesh. (a) Captured image. (b) Partial texture maps. (c) Full texture map. (d) Renderings of the 3D model before and after texture mapping.

video, a world map for relocalization, and intrinsic and extrinsic camera parameters for each frame of the recording.

Apple's ARKit provides only an untextured mesh, but preliminary tests showed that this did not give enough context for the author to understand the space. While 3D scanning is not the focus of our work, we need to provide a usable representation for authoring. To texture the mesh, the system sends the files describing the environment to a web service that parses the recorded video and camera parameters at each frame. The service is implemented in JavaScript and Python using Vue.js [40], three.js [10], and Flask [26]. It begins by calling a Blender script that creates a UV map using default settings, resulting in each vertex in the 3D mesh having a UV coordinate for texture mapping.

To obtain a texture map, we project captured images onto the mesh from recorded camera poses to assign a color to each UV coordinate. We sample a keyframe every 30 frames and place a virtual camera according to the recorded camera parameters, aligning the viewport with the selected frame. Then, for each UV coordinate, we retrieve the corresponding 3D point and connect it to the virtual camera to check whether it is included in this frame. If so, we compute a color for the UV coordinate by bilinearly interpolating pixel colors in the captured image. Each frame results in a partial texture map that contains colors for covered areas on the mesh. Since ARKit does not allow camera exposure to be fixed during scanning, we blend these partial texture maps to obtain the full texture map. We use the pyramid blending algorithm [1] because it can generate a full texture map with smooth borders while avoiding blurring effects caused by averaging. It creates a six-level image pyramid for each partial texture map and blends them to obtain the final texture map (Fig. 2).

Our texturing service runs on a MacBook Pro with a 2.9 GHz quad-core Intel i7 CPU. It takes about 30 minutes to process a scan of a 430-square-foot shared kitchen in a university dormitory. This scan consists of a five-minute video, around 200 keyframes for texture projection, and a mesh with 231k vertices and 426k triangles. The partial UV maps and the blended UV map are 2048×2048 pixels.

3.2 Authoring Interface

In the second step of our workflow, we present the textured 3D scene to the author on their iPad to let them add and manipulate virtual objects and create an AR experience for the remote environment. The interface is similar to those provided by current AR authoring tools. It provides a first-person view, but the author sees a rendering of the remote environment instead of the local one. Camera tracking lets the rendered view update smoothly and realistically as the author moves through their local space and repositions the iPad. Since the authoring environment can be very different from the remote environment, we included an arrow button that virtually moves the author's viewpoint forward along the front vector of the camera while the author presses the button. This lets the author navigate a large remote environment in a smaller physical space. It is the only interaction that goes beyond the AR-authoring metaphor.

The virtual-object placement and manipulation interface is based on those used in current AR authoring tools. A focus square appears on the surface of the model in the center of the screen, and the size and 3D orientation of the square update based on the distance between the virtual camera and the model and the direction of the surface normal there. Tapping an add button shows a list of virtual objects authors can add, and tapping a virtual object in the list adds it to the scene at the focus square (Fig. 3). The virtual object's up direction is aligned with the surface normal.

The author can select virtual objects with a tap and manipulate them using two standard on-screen gestures: pan and a combined rotate and pinch. The author moves the object using a one-finger pan gesture, and its orientation updates to match the surface normal of the changing location. They can rotate the virtual object around the surface normal using two-finger rotation and scale the virtual object relative to its local origin using two-finger pinch. A remove button removes the selected virtual object. A text field shows the active selection and operation in the corner of the interface. The entire scene is saved as a `.scn` file once authoring is complete.

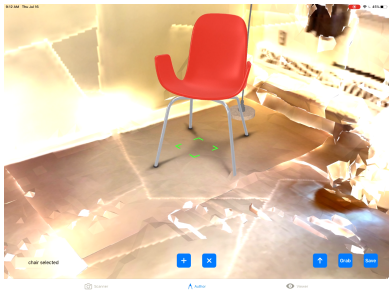


Figure 3: Authoring interface in our baseline solution. The author adds a chair at the focus square on the floor.

3.3 Viewing Interface

Finally, viewers in the remote location can consume the AR content created by the author. In the viewing interface, the viewer roughly scans the environment to relocalize the AR camera to match the world map saved in the scanning process; this normally takes less than a minute. The system then renders all added virtual objects at their intended positions in the space without noticeable error and the viewer can explore the content in AR.

4 FORMATIVE STUDY

We carried out a remote formative study to observe how authors used our baseline solution to create AR experiences for a remote environment and identify challenges that they faced. Understanding the challenges brought by the disconnect between the authoring environment and the experience environment lets us gain insight into remote AR authoring.

Participants. We recruited five participants FP1–FP5 using mailing lists of AR designers. Four were male and one non-binary, and their average age was 33.4. FP1, FP3, and FP4 identified themselves as experience/interaction designers and FP2 and FP5 as HCI researchers. On a scale of 1–7, FP3–FP5 described their familiarity with AR as 5, FP1 as 7, and FP2 as 3. All except FP2 had used some AR authoring tools, and had at some point wanted to create an AR experience for a remote environment. The study lasted one hour and each participant was compensated with a $25 gift card.

Task. The study was done remotely through video conferencing. Participants were asked to furnish an empty kitchen in a university dormitory using our baseline authoring interface after we captured the remote space. The university had removed all furniture from the shared kitchen to avoid gatherings during the pandemic. The design goal was to spend 15 minutes decorating the kitchen with virtual objects to design a space that people could share when the pandemic is over. We provided two pictures of the kitchen before the furniture had been removed as references, but we did not specify a particular layout for the participants to replicate so they could explore the authoring interface with creativity in an open-ended design task. Before the design session started, participants watched an introductory video containing a description of the design task, a panoramic view of the empty kitchen, and a tutorial on functions and interactions in the authoring interface. We asked participants to record the authoring session and save the created scene so we could analyze how they used the authoring interface and could view the created AR experience on site.

Interview. We conducted an in-depth interview with each participant after their session to collect qualitative feedback on the baseline authoring interface and identify challenges in remote AR authoring. First, we asked the participant to comment on the overall experience and whether they could create the AR content that they intended. Then, we asked them to discuss how well the rendering of the captured remote scene functioned as a design context and to describe challenges in navigating to desired viewpoints and manipulating virtual objects. Finally, we collected their suggestions for future improvements and possible use cases based on their experience using the baseline interface and existing AR tools.

Findings. Participants thought the overall experience was positive and fun. They were mostly able to create the content that they intended, although FP1 wanted a larger library of assets. FP4 had previously thought that AR authoring could only be done in the author's current environment and commented that creating AR experiences for remote environments would be a big opportunity. Participants expected remote AR authoring to have many use cases, including remote tutorials, interior design, collaborative content creation, and asynchronous AR authoring. In addition to the positive comments, they also identified challenges with the baseline system and suggested possible improvements.

Navigating in a different environment. Participants acknowledged the disconnect between the authoring and the experience environments as a source of difficulty. Their physical spaces were usually smaller than the remote space and had different layouts. For example, FP5 participated in the study in a small apartment and pointed out at the beginning that, *"the apartment is pretty crowded right now, so it is going to be interesting."* He commented at the end, *"in terms of navigating to a desired view, I would hit my physical constraints about how far I could move. I quickly ran into a wall."* He also suggested a solution—*"I did not want to walk through the space. I just wanted to teleport from one spot to another."*

Understanding the remote environment. All participants were able to understand most content in the remote environment by moving through and looking at the 3D rendering, although it took longer for some than others. FP4 commented, *"pretty quickly it felt very natural to understand the environment,"* whereas FP1 spent more time *"trying to understand more precisely how big the space is."* However, they also expressed concerns about the quality of the reconstruction and texturing as they made understanding the remote environment more difficult. FP1 cared about aesthetics and pointed out that, *"the environment was overexposed and blocky; the lighting was high key and everything was bright, so it became hard to separate surfaces; because of the quality, I could not trust any data about lighting, color, and texture."* Some participants relied on the panoramic view from the video instruction. FP1 commented, *"I focused more on remembering the images in the tutorial. I tried to keep them in my head in terms of the aesthetics of the room."* Similarly, FP5 commented, *"if you had not shown me that panoramic video before, I think I probably would have had some difficulty contextualizing what I was seeing."* They suggested that reference images would be helpful when the 3D reconstruction has poor quality.

Manipulating virtual objects. Our system supports standard touch-based object manipulation gestures. Participants found the gestures

to be intuitive and easy to use. Participants suggested improvements to the interface, such as highlighting selected objects and showing their local origins. They also wished to group the objects in order to manipulate them together. Some participants had difficulty placing virtual objects on a wall that was bumpy because of poor 3D reconstruction; they wished that the wall had been flat.

Design considerations. We formed three design considerations from the findings to improve our system.

- It should let authors freely navigate a large remote environment, even if they are in a small physical space.
- It should help authors understand the remote environment's layout and appearance.
- It should continue to let the authors experience their creation as if they were using AR on site.

5 IMPROVED SYSTEM

We sought to address the challenges identified in the formative study with an improved authoring interface. It includes an improved baseline that we call Portal mode and two new modes, Dollhouse and Peek. Dollhouse mode shows a bird's-eye view of the scene, helping the author understand the remote environment and letting them navigate by manipulating a virtual avatar. Peek mode replaces the rendered scene with a captured image aligned with the AR view, allowing the author to see a photorealistic composite of the virtual content with the actual environment. We also improved the user experience for navigation and object manipulation based on feedback from the formative study.

5.1 Portal Mode

Portal mode (Fig. 1d) is identical to our baseline system, augmented with the system-wide improvements described in Section 5.4.

5.2 Dollhouse Mode

A major challenge in remote AR authoring is the disconnect between the captured remote environment and the author's current environment. The remote space usually differs from the current space in size, layout, entry position, and the presence of obstacles such as furniture. The formative study showed that this disconnect made it difficult for the author to navigate and understand the remote space. For example, participants could not use the baseline solution while sitting down, and they complained that it was difficult to navigate a large captured environment in a small physical space. To address the disconnect, we created Dollhouse mode (Fig. 1e). It lets the author smoothly transition between the first-person view and a bird's-eye view of the entire reconstructed scene.

Two techniques let the user see into the Dollhouse view without obstruction. First, we modified the rendering to use a single-sided material that becomes invisible when viewed from behind and does not occlude more distant content. However, flaws in scanning can result in faces with incorrect orientations and these can remain visible. This is especially distracting when they are part of the ceiling, so we also remove all faces higher than an elevation threshold.

User interactions. We initialize the camera for Dollhouse mode to point toward the center of mass of the model from a fixed distance. The zenith angle is set to 45° and the author can use a rotation gesture to orbit the camera around the center and view the model from different directions. In Dollhouse mode, the author adds a virtual object by moving the focus square to the desired position with a pan gesture. The author can select and manipulate virtual objects using the same set of interactions used in Portal mode, including the new functions described in Section 5.4.

We render a virtual avatar in Dollhouse mode using a circle and an arrow to represent the author's position and orientation in the model. When the author moves the iPad, the position of the circle remains fixed while the direction of the arrow updates based on the orientation of the iPad. Pan and rotate gestures reposition and rotate the avatar, which determines the first-person view when switching back to Portal mode. This interaction lets the author easily move the Portal mode viewpoint across a large space without physically moving. They can switch into Dollhouse mode, reposition the avatar, and switch back. They can similarly make large viewpoint angle changes without physically rotating their body, for example when sitting in a chair.

The author switches between Portal mode and Dollhouse mode by tapping a button, and the camera transitions smoothly between the two modes (Fig. 4). For the transition into Dollhouse mode, we compute an azimuth angle so that the new and current front vectors will be on the same plane. This determines the pose of the Dollhouse mode camera because its zenith angle and distance to the center are fixed. We animate the camera by linearly interpolating between the original and new poses using their positions and look-at vectors. For a transition back into Portal mode, we first orbit the camera by changing its azimuth angle so that its current front vector and the target front vector specified by the avatar are on the same plane. Then we animate the camera in a similar fashion by linearly interpolating between the two poses. The author can smoothly zoom in and out without drastic changes to the context.

5.3 Peek Mode

The other major challenge in remote AR authoring is that the limited quality of the 3D reconstruction and texturing hinder the author's understanding of the spatial layout and appearance of the remote environment. While an accurate 3D model with detailed textures would be an ideal representation of the remote environment, creating one requires an expensive laser scanner, plenty of computing power, or extensive effort in manual modeling. The latest iPad Pro with its LiDAR scanner gives state-of-the-art results for consumer-grade devices, but the resulting mesh is coarse and contains many inaccuracies. This can interfere with understanding and visualizing the remote scene. To address this, we invented Peek mode (Fig. 1f), where the view in the author's iPad is dynamically replaced by the frame from the recorded video that most closely matches the current view (Fig. 5). The virtual objects are added to the scene using the camera pose saved when the frame was recorded, creating a photorealistic composite that is very close to what the consumer will see. This helps the author understand the remote environment and lets them iterate on their design more effectively. The displayed frame does not usually match the current camera pose exactly, because there is rarely an exact match among the captured frames. We update the frame dynamically as the author moves the iPad, but the motion cannot be smooth because of this limitation. To manage

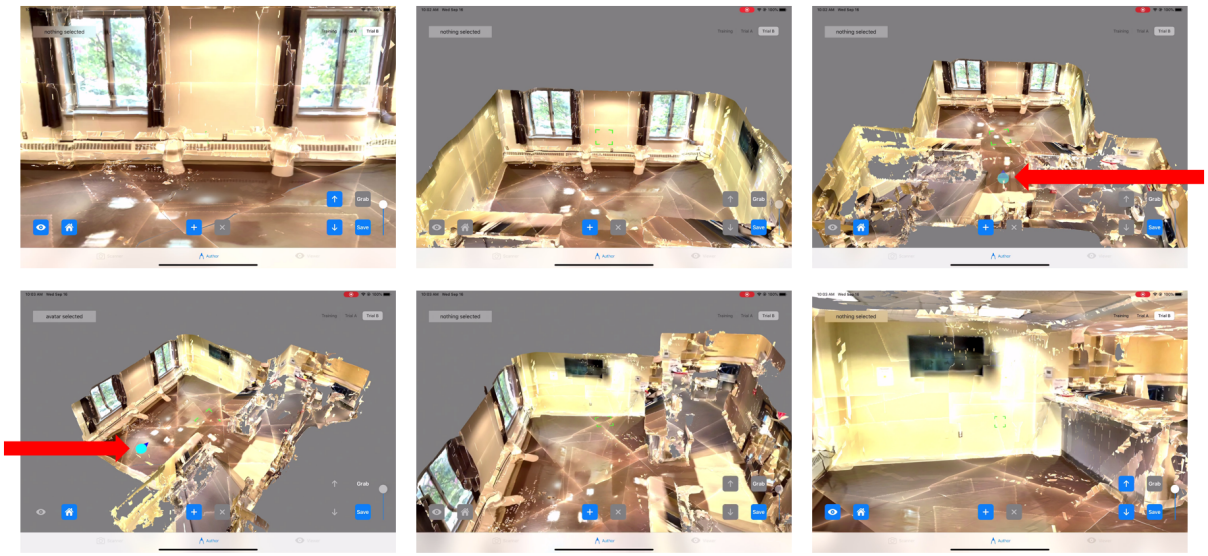


Figure 4: Top: Transition from Portal mode to Dollhouse mode. The blue circle on the floor represents the author's location. Bottom: Transition from Dollhouse back to Portal mode after the author rotates the dollhouse and specifies a new viewpoint. We animate the camera to smoothly zoom out and back in.

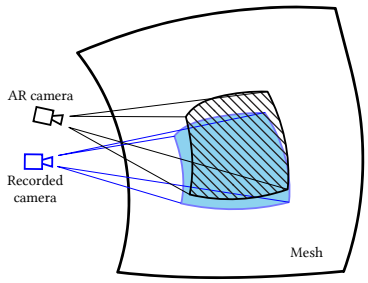


Figure 5: Mesh coverage. Peek mode finds the recorded frame that most closely matches the current AR view by comparing mesh coverage of the AR and recorded cameras.

user expectations, we use a slide-show like display in which one image quickly cross-fades into the next at a low frame rate.

The key question is how to choose a frame from the captured video given the recorded camera parameters and the AR camera parameters so that the frame most closely matches the current AR view. Formally, let c_{AR} denote the AR camera and $c_1, \cdots, c_n$ denote the recorded camera at each frame.

$$\text{Peek}(c_{\text{AR}}) = c_i, \text{ where } i = \underset{j=1,\cdots,n}{\text{argmin}}\ \text{cost}(c_{\text{AR}}, c_j).$$

Here $\text{cost}(c_i, c_j)$ measures the view difference between two cameras. Many properties could be used in the cost function, including position, orientation, time, image features, and mesh coverage. We explored a cost function based on position and orientation, but we found it challenging to determine how to weigh the distance term and the angle difference term. When we used different weights, the retrieved camera could look in a similar direction but be far away from the AR camera, or it could be close to the AR camera but look in a different direction. Using time difference as a metric is based on the assumption that frames that are close in the recorded video have similar coverage. It can help suggest likely new frames given a current one, but it cannot help to find one initially because the AR camera has no timestamp. Comparing image features could help find similar views, but the quality of the 3D reconstruction and texturing makes it difficult to find features in the model that match those in an image.

Through a series of empirical experiments, we found mesh coverage to be the most robust metric to use to define the cost function. When the two cameras cover similar areas on the mesh, these views will contain similar contents. Mesh coverage is determined by the camera pose and mesh geometry. For efficiency, we measure mesh coverage using downsampled 32×32 binary UV maps. Most pixels in the UV map have corresponding 3D points on the mesh. We set a pixel to 1 if the camera view includes its 3D point, and 0 otherwise. The cost function between two UV maps is defined as $1 - \text{IOU}(\text{UV}_i, \text{UV}_j)$, where IOU stands for intersection over union. We precompute the UV coverage map for each recorded camera and create a UV coverage map on the fly for the AR camera (Fig. 6).

Finding the frame with the best coverage has three steps:

(1) Because cameras aimed in very different directions cannot view the same portions of a scene, we begin by selecting the 2,000 frames that have the smallest difference in view angle from the AR camera. For a five-minute initial scan, this represents about one third of the frames.
(2) We consider the 32×32 binary UV maps as bit vectors and compute IOU for each using bitwise operations.
(3) We find the frame with the minimum cost, and consider all frames with costs within 5% of the minimum. We choose

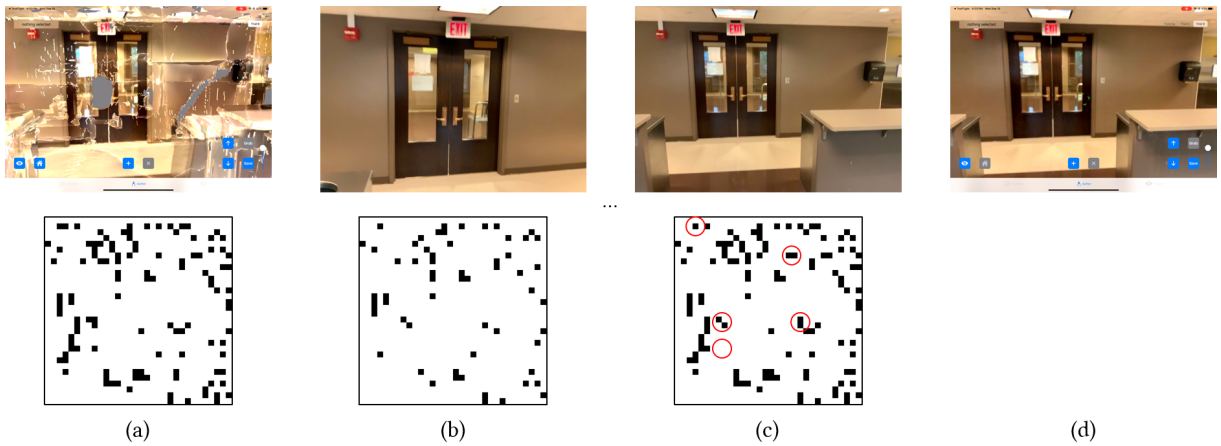


Figure 6: Retrieving a captured image in Peek mode. Top: Camera viewports. Bottom: Computed UV coverage maps. The author sees (a) in Portal mode. Our algorithm searches candidate captured images including (b) and (c), and retrieves (c) because its UV coverage overlaps most with the one in (a). The author then sees the retrieved image in Peek mode (d). The red circles highlight the difference between the UV coverage maps in (a) and (c).

the one whose camera position is closest to the actual AR camera position. This lets us choose a frame with coverage that is nearly as good as the best, but which is closer to the current camera pose.

Our algorithm takes less than half a second, letting us achieve a Peek mode frame rate of two frames per second. Peek mode is less smooth but more realistic than Portal mode.

User interactions. The author switches between Portal and Peek modes by tapping a button, and the interface shows the transition by blending the views. Peek mode supports the same set of interactions for adding and manipulating objects as Portal mode.

5.4 Additional Enhancements

We improved the user experience for navigation based on the feedback from the formative study. Several participants suggested that the viewpoint's elevation in Portal should not change when they used the arrow button to move their viewpoint forward. Portal and Peek modes now remove the vertical component of the camera's front vector when moving with the arrow, and provide a separate slider to control the elevation explicitly. We also added a second arrow button that moves the viewpoint backward so the author can back up without having to turn around.

We also improved the object selection and manipulation interface in all three modes by adding visual feedback and fine-tuning the gestural controls. We now assign a highlight material to the selection and render its origin to make it easier for the author to keep track. The author can select a group of objects by tapping on them sequentially; tapping on the background deselects everything. The author can then move, rotate, and scale the group as a whole.

6 USER STUDY

We evaluated the improved version of DistanciAR with a remote user study. We compared a basic system, having just Portal mode, to a full version having all three modes. The basic system was similar to the baseline used in our formative study, but included the improvements to navigation and object manipulation, making the comparison with the full version fairer. The goal was to understand whether Dollhouse mode and Peek mode helped address challenges identified in the formative study. We also wanted to observe how participants used the different modes to accomplish various tasks in remote AR authoring.

Participants. We recruited eleven participants (P1–P11) using university and company mailing lists. P4 had to be excluded from our analysis because technical errors caused some of his data not to be recorded. There were six females and four males in the remaining ten participants, with an average age of 29.2. Most identified themselves as designers or researchers, and none took part in the formative study. On a scale of 1–7, the average familiarity with AR authoring was 3.8. Five participants had one year or more of AR authoring experience: P1, P9, and P10 had 1 year, P5 had 3 years, and P7 had 8 years. The study lasted one hour and each participant was compensated with a $25 gift card.

Study design. We chose a within-subjects study design to compare the basic system, limited to Portal mode, with the full system having Portal, Dollhouse, and Peek modes. We used an open-ended design task to understand how participants would create an AR experience for the remote environment, which was again to decorate the empty kitchen for post-pandemic community living. While we did not give specific instructions to participants, we provided requirements to prompt a functional and purposeful design. Specifically, the design has to provide seating for ten people, not block the path between seats and the exit door, make use of the countertops on either side of the kitchen, and include decorations near the exit area. The order in which the systems were assigned to a participant was counter-balanced. To reduce learning effects, participants were asked to create two different AR experiences, one using the basic system and another using the full system. For the second trial, we encouraged participants to try a different design, arranging at least four objects differently.

Procedure. The study was done remotely through video conferencing with participants. All participants used an AR-capable iPad device in the study. At the beginning of the study, participants watched an introductory video containing a description of the design task and screen recordings of user interactions. We then held a 10-minute training session for the participants to learn how to use the authoring interface. In this session, we presented a remote scene of a dormitory bedroom and asked participants to familiarize themselves with Portal mode, Dollhouse mode, and Peek mode by using these modes to navigate within the scene and decorate the room with 3D objects. During the study, we asked participants to follow the think-aloud protocol, so we could follow their thought processes. Participants were told to finish each design session in about fifteen minutes. They completed a survey after each design session and participated in an interview at the end of the study.

Measurement. We recorded the video meetings. We also asked participants to record their iPad's screen during the study. The survey that they filled out after each trial included five questions: "the application was easy to use" (Q1), "it was easy to understand the remote space" (Q2), "it was easy to navigate to a desired view" (Q3), "it was easy to place objects at a desired location" (Q4), and "it was easy to create my intended AR content" (Q5). The participants rated their agreement to each question on a scale of 1–7, where 1 stood for "strongly disagree" and 7 stood for "strongly agree." In the post-study interview, we asked them to comment on Portal mode, Dollhouse mode, and Peek mode, what each mode was useful for, features that they would like to have in a new version, and how this experience compared to previous AR authoring experiences. To gain more insight into participants' design activity, we also recorded the number of added virtual objects and total time spent in each design session.

7 RESULTS

We analyzed the differences in participants' ratings, time spent, and number of objects added using a paired-samples t-test. Effect size is reported as Cohen's d value. We used a Shapiro-Wilk test on all of the data to check for normality. We found that all data were normally distributed, except for participants' ratings in Q3. We decided to still carry on the t-test on this measurement and report all of our results below.

For the ratings (Fig. 7a) , the average ratings in all questions were higher for the full system compared to the basic system. The differences between ratings in Q2, "it was easy to understand the remote space," ($p = 0.003$, $t = -4.019$, $d = 1.434$) and Q3, "it was easy to navigate," ($p = 0.0004$, $t = -5.438$, $d = 1.58$) were found to be statistically significant.

Participants' behavior in the task were also different in both systems. We found that participants spent more time when using the full system (Fig. 7b). On average, the participants spent 10.7 minutes ($\sigma = 2.5$) in the basic system and 12.9 minutes ($\sigma = 3.0$) in the full system, and the difference was found to be statistically significant ($p = 0.014$, $t = -3.034$, $d = 0.806$). For the number of objects added (Fig. 7c), on average, participants added 23.3 objects ($\sigma = 6.1$) in the basic system and 19.8 objects ($\sigma = 4.3$) in the full system, and the difference was found to be statistically significant ($p = 0.0121$, $t = 3.1305$, $d = 0.6643$).

In order to gain more insight into how Portal, Dollhouse, and Peek modes were being used in the full system, we took the screen recordings of participants using the full system and labeled the usage of each mode. We visualized this data on normalized task timelines (Fig. 8a) and found that most participants frequently switched between Portal mode and Dollhouse mode during the task and used Peek mode primarily in the second half of the task. We computed the percentage of time spent in each mode. We analyze the differences (Fig. 8b) using a one-way analysis of variance (ANOVA) and found that the differences among the three modes were statistically significant ($p = 0.0002$, $F = 11.6466$, $\eta^2 = 0.8627$). Through a Tukey-Kramer posthoc test, we found that the differences between Peek mode and Portal mode, as well as between Peek mode and Dollhouse mode, were also statistically significant.

We also collected subjective feedback from participants about each mode, and how they were used together to help them accomplish the tasks. For Dollhouse mode, participants liked this mode because it provided global information that helped them better perceive the remote space (P3, P5, and P11). P5 commented on a benefit of Dollhouse mode, *"I liked the ability to contextualize by going in and out; I could work in a fairly small space and I felt like that I could do a lot in this virtual room."* P9 used it to understand depth, reporting, *"the task was a lot harder without Dollhouse mode because sometimes it was hard to get a sense of depth in Portal mode,"* and to navigate the space, saying, *"being able to shift viewpoints quickly with the avatar was really good."* Participants also suggested that Dollhouse mode should be used in combination with Portal mode. P10, a professional architect, commented, *"planning a basic layout in Dollhouse mode is usually what an architect would do; then I can view the design in Portal mode and slightly adjust the objects."* P11 commented that Portal mode had complementary benefits to Dollhouse mode when adding objects on the ceiling.

For Peek mode, P9 and P11 commented that it helps them validate their design in poorly captured areas in the environment, where they *"could not really see what was going on."* P7 and P8 also used Peek mode for the same purpose. Neither realized that there were two metallic refrigerators in the kitchen until they used Peek mode. Peek mode also helped P5 and P10 made better design decisions. P5 had intended to decorate the exit area by putting a vase next to the door, but she changed her mind when Peek mode made her realize that the door might knock over the vase when people were entering. Similarly, P10 mentioned using Peek mode to find and remove objects that did not look suitable in the actual environment.

8 DISCUSSION

Our user study demonstrated that participants had a clear preference for the full system with Portal, Dollhouse, and Peek modes. Participants were most positive about the full system's ability to help them understand and navigate the remote space. They were able to create useful AR designs shown in Fig. 9. These findings show that our improvements over the basic system, specifically Dollhouse and Peek modes, helped address the navigation and spatial understanding challenges identified in our formative study. However, they still spent substantial time in Portal mode, validating it as a useful way to do remote AR authoring.

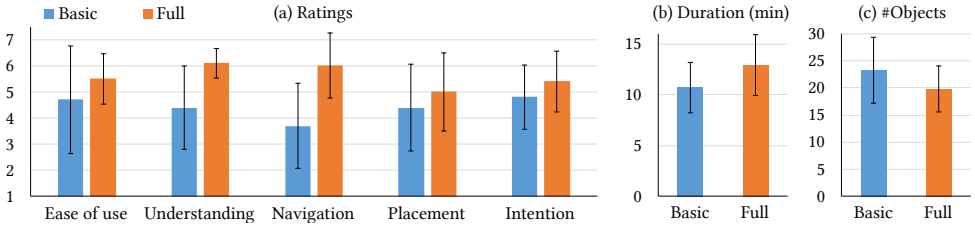


Figure 7: Comparison between the basic system and the full system. (a) Ratings of the two systems for Q1–Q5 (left to right). (b) Time spent using each system. (c) Number of objects added in each system.

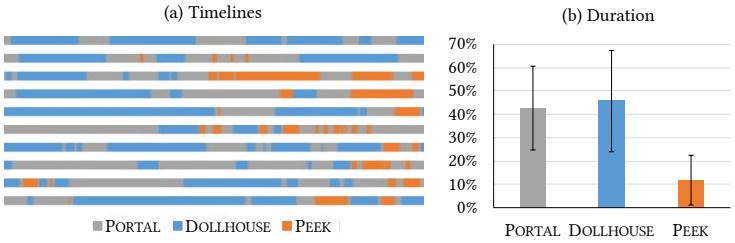


Figure 8: Comparison between Portal mode, Dollhouse mode, and Peek mode in the full system. (a) Mode usage over time for P1–P11 (top to bottom, excluding P4). (b) Time spent in each mode.

Adding more viewing modes into a system incurs a risk of making the experience more demanding. This is an issue that has been well-studied in 2D interfaces [39]. As shown in Fig. 8a, participants frequently switched among Portal, Dollhouse, and Peek modes. This mode-switching might account for participants spending more time in the full system. However, participants still preferred the full system even with its complexity. Subjective feedback from participants suggest that Dollhouse and Peek mode were used to complement Portal mode. Specifically, Dollhouse mode was used mainly for large-scale tasks such as getting an overview of the layout, navigating, and manipulating objects over a large distance. In contrast, Portal and Peek mode, which provide a more AR-like view of the scene, were used more for fine-scale adjustment. Peek mode was typically used toward the end of the task in order to validate the design against a realistic view of the environment. These results suggest that Dollhouse and Peek mode were considered useful by participants. They complement Portal mode well and improve the remote AR authoring experience.

One of our key design requirements is letting authors experience their design as if they were using AR on site. Participants spent more time in Dollhouse than in the other modes, but the experience of using this mode is not at all similar to AR. However, our results suggest that this mode is critical in a remote AR authoring system because it complements the shortcomings of AR-like modes like Portal and Peek. First, it can show an overview of the environment, which is important because it reduces the effort needed for users to understand the structure of the space and their location within it [11]. Second, it enables large-scale interactions that are normally difficult in AR, such as quickly moving groups of 3D objects from one room to another. Thus, although Dollhouse mode does not provide an AR-like experience, it provides necessary support for users to view and create in the remote environment.

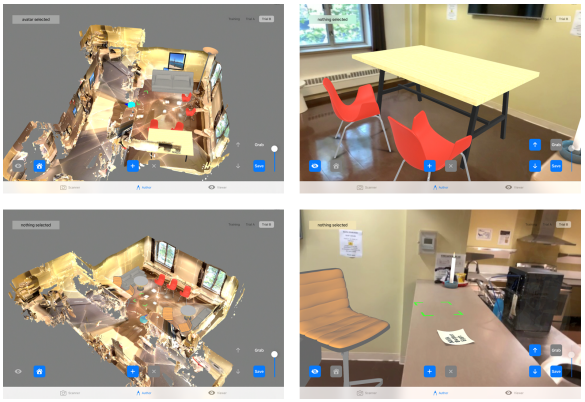


Figure 9: Representative AR designs created by P5 (top) and P9 (bottom). Left: Dollhouse mode. Right: Peek mode. P5 put chairs, a table, and a sofa all facing the TV. P9 put a candle and a sticky note on the countertop next to high chairs. Candle, sticky note, and cup models: © 2020 Apple Inc.

Limitations. We used an open-ended task in our user study. While our study validated the remote AR authoring workflow and provided insight into how participants use our system, it did not let us evaluate the differences in their designs. For example, the significant difference in the number of objects added could be due to unexpected behavior of a few participants. P3, P7, and P10 added an unusual number of sticky notes—6, 9, and 13, respectively, while no other participant added more than 3. This does not add real value to their design. Excluding sticky notes, we found that participants

added approximately the same number of objects in each system (an average of 19.6 in the basic system and 19 in the full system). We propose that an alternative study with stricter design requirements could more accurately evaluate their performances.

Our work provides a foundation for future studies on remote AR authoring. For example, comparing with desktop authoring would provide more insight into the value of an AR-like authoring experience. Comparing with in-situ AR authoring would highlight the potential physical limitations of DistanciAR. Comparing each mode on different tasks would determine the benefits of Dollhouse mode in helping users navigate a large AR scene and the perceptual difference between Peek and Portal modes. These studies are useful even beyond the scope of remote AR authoring and therefore would require more careful evaluation and analysis.

Our system requires that the user's AR device can capture and store a 3D scan of the environment. Although modern consumer AR devices such as the iPad LiDAR or the Hololens are starting to support 3D scanning off-the-shelf, the technology is still an active research topic and there are inherent artifacts in the resulting scans. While including Dollhouse mode and Peek mode made it easier for the author to understand and navigate the remote environment, object placement and manipulation are still affected by the quality of the scan. For example, 3D objects cannot stand upright if the scanned surfaces are not flat. In our study, most participants were not affected by the scan quality. However, P1 and P6 suggested that the experience of adding objects on uneven surfaces could be improved. Adding an additional mesh refinement step to the pipeline can be an immediate solution to this problem. For example, for regions that were poorly scanned, we could use a machine-learning image-based 3D reconstruction algorithm such as PlaneRCNN [20] to produce alternative results and combine these results to improve the existing scans.

The quality of our Peek mode compositing is hindered by the lack of depth information in our video capture. Captured frames only serve as backgrounds, which means that their contents cannot occlude rendered virtual objects. In our kitchen task, this was most evident with the counters between the kitchen and living areas. However, additional depth data could be captured either from the device's built-in API [12] or estimated from stereoscopic parallax [42]. Integrating depth data into Peek mode to enable realistic occlusion effect would be a good direction for future work.

Finally, our system is currently implemented on a mobile AR device. DistanciAR can potentially be ported to an AR HMD such as the Hololens. However, the experience might not be similar. The Hololens uses see-through optics, which means that users see AR graphics through an optical lens, not through a video composited image. Seeing the remote environment in a small HMD FOV overlaying the current environment might be disorienting. Developing a comparable remote AR experience in the HMD is an interesting direction and is left for future work.

9 CONCLUSIONS AND FUTURE WORK

We presented DistanciAR, a novel workflow for authoring site-specific AR experiences for remote environments. It involves three steps: capturing the remote environment, authoring from a different location, and viewing the created AR experience on site.

Our original hypotheses were that authors would prefer an immersive interface that closely simulated an in-situ AR authoring experience, and that a 3D reconstruction based upon the capabilities of consumer scanning technology would suffice. A formative study of a baseline system made us re-evaluate these hypotheses.

When an author does AR authoring in situ, they can understand the environment just by looking around, and they have access to the full space for navigation. However, when authoring for a remote environment, neither is true. We added the non-immersive Dollhouse mode to our system to provide a birds-eye view that helped authors understand the environment and allowed them to reposition the immersive viewpoint without moving. While this is a non-immersive interface, it does not invalidate our hypothesis about the value of an immersive interface—authors in our user study still spent slightly more than half of their time in immersive modes. The combination of immersive and non-immersive authoring modes proved to be very effective.

To address the reconstruction deficiencies, we added Peek mode, which allowed authors to view and modify their experiences using photorealistic captured frames.

There are several interesting directions for future work. Since DistanciAR demonstrates a workflow for remote content creation that mitigates the disconnect between contexts, it would be valuable to explore how collaboration fits into the workflow. It could be the basis for a collaborative system for remote AR authoring, where authors create AR experiences from different locations while keeping track of each other's changes.

We used interior design as a proof-of-concept use case to show the benefits of our remote AR authoring workflow, but the system can support other applications such as tutorial and navigation guidance. The authoring interactions supported in DistanciAR are similar to those in in-situ AR authoring. The DistanciAR framework could be easily extended to support additional AR authoring tasks like proximity triggers, animations, and interactive behaviors. However, authoring interactions with the actual environment would be more challenging due to the lack of physical access.

While we added Dollhouse mode to address specific deficiencies in the immersive authoring interface, it proved to be useful beyond that. Authors found it a convenient way to place and modify major elements like tables and chairs. We think that being able to interact with a bird's-eye view could also be helpful for in-situ authoring.

Peek mode's immersive experience was limited by being restricted to using captured video frames. We would like to explore stitching frames into panoramas [21] and then using portions of the results to create backgrounds that more closely match the AR camera view. We would also like to explore techniques that extract depth and planes from images [12, 20, 42] in order to enable occlusion and create smoother and higher-quality reconstructions.

10 ACKNOWLEDGMENTS

This research began during an internship at Adobe Research and was sponsored in part by Adobe Research. This work was partially supported by National Science Foundation award #1942257.

REFERENCES

[1] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. 1984. Pyramid Methods in Image Processing. *RCA Engineer* 29, 6

(1984), 33–41.
[2] Adobe. 2020. Aero. https://www.adobe.com/products/aero.html. Accessed Sep 12, 2020.
[3] Apple. 2020. ARKit. https://developer.apple.com/documentation/arkit. Accessed Sep 12, 2020.
[4] Apple. 2020. Reality Composer. https://developer.apple.com/augmented-reality/reality-composer/. Accessed Sep 12, 2020.
[5] Apple. 2020. RealityKit. https://developer.apple.com/documentation/realitykit. Accessed Sep 12, 2020.
[6] Apple. 2020. SceneKit. https://developer.apple.com/documentation/scenekit. Accessed Sep 12, 2020.
[7] Narges Ashtari, Andrea Bunt, Joanna McGrenere, Michael Nebeling, and Parmit K Chilana. 2020. Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.
[8] Matt Bell. 2020. Matterport. https://matterport.com/. Accessed Dec 17, 2020.
[9] Jan Brejcha, Michal Lukác, Zhili Chen, Stephen DiVerdi, and Martin Cadík. 2018. Immersive Trip Reports. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany). Association for Computing Machinery, New York, NY, USA, 389–401.
[10] Ricardo Cabello. 2020. three.js. https://threejs.org/. Accessed Sep 12, 2020.
[11] Andy Cockburn, Amy Karlson, and Benjamin B Bederson. 2009. A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. *Comput. Surveys* 41, 1 (2009), 1–31.
[12] Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces, Shahram Izadi, Adarsh Kowdle, Konstantine Tsotsos, and David Kim. 2020. DepthLab: Real-Time 3D Interaction with Depth Maps for Mobile Augmented Reality. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA). Association for Computing Machinery, New York, NY, USA, 829–843.
[13] Facebook. 2020. Spark AR Studio. https://sparkar.facebook.com/ar-studio/. Accessed Sep 12, 2020.
[14] Anhong Guo, Ilter Canberk, Hannah Murphy, Andrés Monroy-Hernández, and Rajan Vaish. 2019. Blocks: Collaborative and Persistent Augmented Reality Experiences. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 83 (Sept. 2019), 24 pages.
[15] Sinem Guven, Steven Feiner, and Ohan Oda. 2006. Mobile Augmented Reality Interaction Techniques for Authoring Situated Media On-Site. In *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE, 235–236.
[16] André Kunert, Alexander Kulik, Stephan Beck, and Bernd Froehlich. 2014. Photoportals: Shared References in Space and Time. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing* (Baltimore, Maryland, USA). Association for Computing Machinery, New York, NY, USA, 1388–1399.
[17] Germán Leiva, Cuong Nguyen, Rubaiat Habib Kazi, and Paul Asente. 2020. Pronto: Rapid Augmented Reality Video Prototyping Using Sketches and Enaction. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA). Association for Computing Machinery, New York, NY, USA, 1–13.
[18] Yuwei Li, Xi Luo, Youyi Zheng, Pengfei Xu, and Hongbo Fu. 2017. SweepCanvas: Sketch-based 3D Prototyping on an RGB-D Image. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. 387–399.
[19] David Lindlbauer and Andy D. Wilson. 2018. Remixed Reality: Manipulating Space and Time in Augmented Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada). Association for Computing Machinery, New York, NY, USA, 1–13.
[20] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. 2019. PlaneRCNN: 3D Plane Detection and Reconstruction from a Single Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4450–4459.
[21] Feng Liu, Yu-hen Hu, and Michael L. Gleicher. 2008. Discovering Panoramas in Web Videos. In *Proceedings of the 16th ACM International Conference on Multimedia* (Vancouver, British Columbia, Canada). Association for Computing Machinery, New York, NY, USA, 329–338.
[22] Blair MacIntyre, Maribeth Gandy, Jay Bolter, Steven Dow, and Brendan Hannigan. 2003. DART: the Designer's Augmented Reality Toolkit. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.* IEEE, 329–330.
[23] Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L. Davidson, Sameh Khamis, Mingsong Dou, Vladimir Tankovich, Charles Loop, Qin Cai, Philip A. Chou, Sarah Mennicken, Julien Valentin, Vivek Pradeep, Shenlong Wang, Sing Bing Kang, Pushmeet Kohli, Yuliya Lutchyn, Cem Keskin, and Shahram Izadi. 2016. Holoportation: Virtual 3D Teleportation in Real-Time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan). Association for Computing Machinery, New York, NY, USA, 741–754.
[24] Patrick Paczkowski, Min H. Kim, Yann Morvan, Julie Dorsey, Holly Rushmeier, and Carol O'Sullivan. 2011. Insitu: Sketching Architectural Designs in Context. *ACM Trans. Graph.* 30, 6 (Dec. 2011), 1–10.
[25] Benjamin Reynolds, Hisham Bedri, Valentin Heun, Anna Fusté, and Christian Vazquez. 2019. Remote Spatial Programming and Collaboration Using a Real-Time Volumetric Capture Space. In *ACM SIGGRAPH 2019 Virtual, Augmented, and Mixed Reality* (Los Angeles, California). Association for Computing Machinery, New York, NY, USA, Article 27, 1 pages.
[26] Armin Ronacher. 2020. Flask. https://flask.palletsprojects.com/en/1.1.x/. Accessed Sep 12, 2020.
[27] Aditya Sankar and Steve M. Seitz. 2017. Interactive Room Capture on 3D-Aware Mobile Devices. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada). Association for Computing Machinery, New York, NY, USA, 415–426.
[28] Noah Snavely, Steven M. Seitz, and Richard Szeliski. 2006. Photo Tourism: Exploring Photo Collections in 3D. *ACM Trans. Graph.* 25, 3 (July 2006), 835–846.
[29] Noah Snavely, Steven M Seitz, and Richard Szeliski. 2008. Modeling the World from Internet Photo Collections. *International Journal of Computer Vision* 80, 2 (2008), 189–210.
[30] Rajinder S. Sodhi, Brett R. Jones, David Forsyth, Brian P. Bailey, and Giuliano Maciocci. 2013. BeThere: 3D Mobile Collaboration with Spatial Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France). Association for Computing Machinery, New York, NY, USA, 179–188.
[31] Richard Stoakley, Matthew J. Conway, and Randy Pausch. 1995. Virtual Reality on a WIM: Interactive Worlds in Miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA). ACM Press/Addison-Wesley Publishing Co., USA, 265–272.
[32] Yuta Sugiura, Hikaru Ibayashi, Toby Chong, Daisuke Sakamoto, Natsuki Miyata, Mitsunori Tada, Takashi Okuma, Takeshi Kurata, Takashi Shinmura, Masaaki Mochimaru, and Takeo Igarashi. 2018. An Asymmetric Collaborative System for Architectural-Scale Space Design. In *Proceedings of the 16th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry* (Tokyo, Japan). Association for Computing Machinery, New York, NY, USA, Article 21, 6 pages.
[33] Franco Tecchia, Leila Alem, and Weidong Huang. 2012. 3D Helping Hands: A Gesture Based MR System for Remote Collaboration. In *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry* (Singapore, Singapore). Association for Computing Machinery, New York, NY, USA, 323–328.
[34] Unity Technologies. 2020. Unity. https://unity.com/. Accessed Sep 12, 2020.
[35] Balasaravanan Thoravi Kumaravel, Fraser Anderson, George Fitzmaurice, Björn Hartmann, and Tovi Grossman. 2019. Loki: Facilitating Remote Instruction of Physical Tasks Using Bi-Directional Mixed-Reality Telepresence. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA). Association for Computing Machinery, New York, NY, USA, 161–174.
[36] Michael Tsang, George W. Fitzmaurice, Gordon Kurtenbach, Azam Khan, and Bill Buxton. 2002. Boom Chameleon: Simultaneous Capture of 3D Viewpoint, Voice and Gesture Annotations on a Spatially-Aware Display. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology* (Paris, France). Association for Computing Machinery, New York, NY, USA, 111–120.
[37] Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Ke Huo, Yuanzhi Cao, and Karthik Ramani. 2020. CAPturAR: An Augmented Reality Tool for Authoring Human-Involved Context-Aware Applications. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA). Association for Computing Machinery, New York, NY, USA, 328–341.
[38] Zeyu Wang, Shiyu Qiu, Qingyang Chen, Natallia Trayan, Alexander Ringlein, Julie Dorsey, and Holly Rushmeier. 2019. AniCode: Authoring Coded Artifacts for Network-Free Personalized Animations. *The Visual Computer* 35, 6-8 (2019), 885–897.
[39] Michelle Q Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. 2000. Guidelines for Using Multiple Views in Information Visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces*. 110–119.
[40] Evan You. 2020. Vue.js. https://vuejs.org/. Accessed Sep 12, 2020.
[41] Ya-Ting Yue, Yong-Liang Yang, Gang Ren, and Wenping Wang. 2017. SceneCtrl: Mixed Reality Enhancement via Efficient Scene Editing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada). Association for Computing Machinery, New York, NY, USA, 427–436.
[42] Yinda Zhang, Neal Wadhwa, Sergio Orts-Escolano, Christian Häne, Sean Fanello, and Rahul Garg. 2020. Du^{2}Net: Learning Depth Estimation from Dual-Cameras and Dual-Pixels. In *European Conference on Computer Vision*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 582–598.