An Authoring Tool for Creating Interactive AR User Tutorials by Demonstration

Junhan (Judy) Kong

CMU-CS-20-116

May 2020

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Jeffrey P. Bigham (Chair)
Patrick Carrington

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science.

Keywords: Human-Computer Interaction, Augmented Reality

Abstract

Using complex interfaces can be quite challenging, especially for those who do not enjoy exploring unfamiliar interfaces through trial and error. Existing tutorial systems can be cumbersome, and sometimes difficult to use. While augmented reality has been adopted to help users learn to use some interfaces, these AR user manuals are usually designed specifically for one single interface, so creating AR user manuals for a variety of interfaces requires a substantial amount of work, and the authoring tools for these manuals are usually not friendly to users without background knowledge in related areas such as programming and 3D modeling. To solve this problem, we present a system that automatically generates interactive AR user tutorials by demonstration. Our system first guides experienced users to add references that help our system understand the interface and to demonstrate steps to complete certain tasks. Then our system automatically detects user interactions and asks users for supplemental information on the interactions. Based on this information, our system generates an action sequence for each task, and provides a combination of text instructions, visual and audio guidance for new users to access the interface.

Acknowledgments

I would like to thank my amazing advisor, Prof. Jeffrey P. Bigham, for guiding me to find directions in my research, providing lots of great advise to help me form ideas of the project and better write about and present my work, and for funding my master's study. He's the one who led me into accessibility research, and this invaluable experience made me determined to continue a path in HCI research and accessibility.

I would also like to thank my committee member, Prof. Patrick Carrington, for providing lots of great feedback for my work and helped me shape my project into where it it now.

And I would like to thank my mentors, Amy Pavel and Anhong Guo, for helping me form the idea of this project, having lots of great discussions that inspired many design decisions in this project. Working with them, I have also learned so much in how to consolidate research ideas, iterating on them, and documenting my work along the process.

I would like to thank other people in the BigLab and FigLab for sharing their thoughts and experiences in research and helped me through the times when I was confused or unsure about research and decisions moving forward, or getting stuck.

Additionally, I want to thank Prof. Justine Sherry for her amazing class which systematically introduced me to academic writing and substantially improved my skills and build my confidence in writing.

Finally, I would like to thank my family and friends who have always been so supportive along the way.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Usability of a user interfaces can greatly affect the user experience when someone interacts with the interface. Good user interfaces are expected to be simple, natural, and consistent [16], and it can be quite challenging for a user to interact with interfaces that are not user friendly. Although the design and development of some existing artifacts follow a user-centered design process, lots of currently existing interfaces are still not well-designed and hard to use. In the meantime, current user manuals and tutorial systems can be cumbersome and inconsistent for different types of machines and interfaces. Prior research has shown that predictability of usage and same patterns of interaction are considered crucial in making interfaces universally accessible [11]. Thus, designing an intuitive tutorial system that's consistent across different types of interfaces becomes crucial in making existing complex user interfaces easier to use.

While augmented reality (AR) has been adopted in creating user manuals and assisting with certain tasks, most of these existing AR tools are focused on one single interface or task and could not be generalized. This makes the development of AR tools supporting a larger variety of tasks challenging and time-consuming. On the other hand, although lots of work has been done on AR authoring, most current AR authoring platforms are usually facing towards developers, and lots of times require users to have background and knowledge in programming and 3D modeling. Thus, these authoring platforms are not friendly and easy to use for a general user. Additionally, these platforms are mostly based on desktop and web environments, which can be a little too heavyweight for non-expert users.

1.2 Key Contributions

We propose an authoring tool for creating interactive AR user tutorials by demonstration based on the Apple ARKit platform [1]. In the authoring mode of our system, an experienced user is guided to first add static locations of a machine interface as anchor points, then add reference images of each step in completing a task, so that our system can locate the machine interface and know the current step of a user in the process. By tracking finger locations relative to the anchor point when the experienced user interacts with the interface, our system automatically

detects and records the actions taken by the experienced user at each step, and then asks the user for verification and supplemental information. In the access mode, our system automatically recognizes reference images appearing in the camera view and identifies the current step of the user. Then it reproduces a simulation of finger movements relative to the anchor points in the AR scene on the phone screen, thus guiding the new user to interact with the interface step by step. Our system also uses supplementary input entered by experience users for text and audio instructions in the access mode, as an auxiliary guidance in addition to the AR visual guidance.

Through design iterations and preliminary user studies, we increased the usability and generalizability of our system. In the end, we provide a full end-to-end demonstration of our system, from an experienced user creating the tutorial to a new user accessing the interface using the tutorial. We also tested our novel anchor point selection mechanism with multiple different interfaces, and proved our proposed design effective across different interfaces.

1.3 Thesis Organization

In this document, we discuss the background and related work of the thesis in Chapter 2. In Chapter 3, we describe the system design and implementation in detail, including how our system guides an experienced user to create AR user tutorials in the authoring mode, and how it automatically generates guidance to allow new users to access an interface in the access mode. Then we present our design iterations in Chapter 4, going from our initial prototypes to iterative refinements we made to the design. In Chapter 5, we present our results by showing an end-to-end demo of the system and testing critical features with multiple different interfaces, then discuss the successes, trade-offs and limitations of our work. Finally, we summarize our work and discuss future work of this thesis in Chapter 6.

Chapter 2

Background and Related Work

2.1 Augmented Reality (AR)

Although the definition varies in existing literature, an augmented reality (AR) system can be broadly described as the physical real world enhanced by virtual components. Milgram et al. [14] defined AR in terms of a continuum relating purely virtual environments to purely real environments, to be a middle ground between these two worlds. In contrast to virtual reality (VR) which immerses a user in the virtual environment, AR allows users to see the real world, and virtual objects superimpose upon or composite with the real world [2]. In addition to the visual augmentations to the physical real world, other aspects of reality such as audio, motion, haptics, taste/flavor, and smell [18] can also be potentially incorporated in an augmented reality (AR) or more broadly, mixed reality (MR) system. Applications of AR span across a variety of different fields including medical visualization, maintenance and repair, annotation, entertainment, etc. [2]

2.2 Step-by-Step Tutorial Systems

Step-by-step tutorial systems have been shown to be effective for complex user interfaces and complicated tasks, especially to certain user groups who do not prefer trial and error [12]. Research has been done in creating step-by-step tutorial systems, such as MixT by Chi et al. which segments screen capture videos, applies video compositing techniques and highlights interactions through mouse trails. StateLens by Anhong et al. [10] used a state diagram to represent a user interface and guide visually-impaired users to interact with touchscreens step by step, and their work inspired the modeling of user interactions with interfaces into action sequences in this work. Additionally, Black et al.'s study [3] found that minimalistic approaches to designing instruction manuals are most effective and that immediately involving the user in realistic tasks helps them learn faster.

2.3 AR User Manuals and Task Scaffolding

In recent years, Augmented Reality has been widely adopted to solve challenges and assist with tasks in a variety of areas. Lots of work has been done in using AR to scaffold interactions with interfaces or completing certain tasks. Blattgerste et al. 's work [4] compared conventional and AR instructions for manual assembly tasks and found that compared to using the paper instructions, users made fewer errors with AR assistance. AR tools have been developed to assist with tasks in a variety of different areas, for example, helping patients to test their blood at home [8] and simulating sudden cardiac arrest emergencies [7]. While these tools were shown to be effective in assisting with the specific tasks they were designed for, each of these tools is focused on one single application area, and thus developing such tools for a variety of purposes would takes substantial time and effort.

2.4 AR Authoring Tools

To make AR authoring easier for non-expert users, research has been done in creating AR authoring tools facing users without background knowledge in programming and AR development. For example, ComposAR by Seichter et al. [17] provides a python-based GUI for non-programming users to create AR and MR projects with little 3D modeling knowledge; Gimeno et al.'s work [9] on easy-to-use AR authoring tools for industrial applications simplifies development of AR applications for the execution of industrial sequential procedures. While these efforts effectively lowered the barriers of AR authoring, they were still based on desktop GUIs and aimed for general-purpose AR development, so the software could still be too sophisticated for general users trying to create an AR tutorial.

One prior work that relates especially closely to our work is ASMIM by Nguyen et al. [15], an augmented reality authoring system for mobile interactive manuals. ASMIM allows trainers (or expert editors) to design the step-by-step interactive manuals and provide visual instructions such as interest regions, text, and action animations, so that trainees (or users) can later follow these instructions. Although our work is similar to ASMIM in that our system also follows a trainer-trainee structure, our work does not require manual selections of points of interest on the interfaces, which is more automated compared to prior work. Moreover, our work supports a larger variety of user gestures by tracking users' fingers instead of selecting from a pool of predefined actions.

Chapter 3

System Design and Implementation

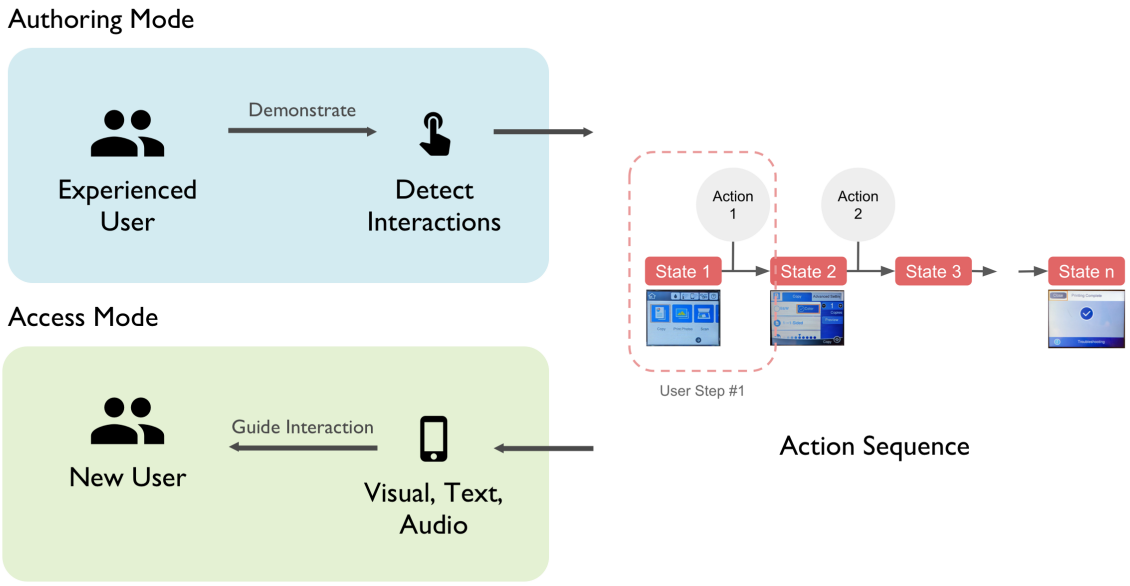

Figure 3.1: System Design

Our system includes two modes, the authoring mode and the access mode. In the authoring mode, an experienced user is guided to add references that help the system understand the machine interface, and then asked to specify a set of tasks and demonstrate each task step by step, for example, copying a document using a printer. The system generates a sequence of user actions needed to complete the task by processing the captured demonstration in real-time and asking for supplemental user input on the interaction. In the access mode, a new user selects which task they want to do with the interface, and our system then guides the user to interact with the interface step by step with a combination of AR visual guidance and text and audio instructions.

3.1 Understanding an Interface

In both the authoring mode and the access mode, our system uses a set of anchor points to track machine location and thus provide AR guidance on top of the machine in the phone application. Our system also uses a set of reference images to determine the status of an interface and a user's current tutorial step.

3.1.1 Modeling Tasks

Our system models a task on a user interface as an action sequence. Each action sequence contains a sequence of "states" of the interfaces that can be uniquely identified by a corresponding reference image, which we will explain in more detail in the next paragraph. An "action" is required between two neighboring states in the sequence, and makes the transition from the previous state to the next. The action is usually some user interaction with the interface, for example, pressing a button, opening or closing parts of the machine, etc. As a user interacts with an interface, the action sequence keeps proceeding to the next state as it reaches the terminating state of the sequence, which is usually something state with a reference image like a success message displayed on the interface screen.

A reference image is a unique status of a machine that helps the system identify the current user step. For example, an image of the digital display of a printer interface usually contains messages on the current status of the machine such as ready to print, copying in process, or paper stuck, etc. and thus provides information that helps our system understand where the user is in the process.

3.1.2 Locating an Interface in 3D Space

An anchor point is a static location on the interface, for example, the machine logo, the control panel, etc. In order for our system to robustly and accurately track interfaces in both the authoring mode and the access mode, an anchor point is expected to be within the same camera view as any user interactions, for example, button presses. The anchor point needs to be trackable, which means that it has enough feature points and is large enough for the application to recognize. The anchor point should also not be blocked from the camera view by any user interaction.

For example, as shown in Figure 3.3, the front side of the printer interface is divided into two anchor points - the left anchor point (the control panel) and the right anchor point (the printer logo); the microwave interface is also divided into two anchor points - the top anchor point (upper part of the control panel) and the bottom anchor point (lower part of the control panel); the door intercom interface is divided into three anchor points, the left anchor point (the talk button), the middle anchor point (the listen button), and the right anchor point (the open-door button).

Our system uses Apple ARKit to recognize saved reference images, anchor points, and track the 3D locations of any reference images that we might use during the process. The set of reference images and the set of anchor points are both inputted by an experienced user in the authoring mode, which we will explain in more detail in the next section. With this information, our system is able to track the location and status of an interface fairly robustly and accurately.

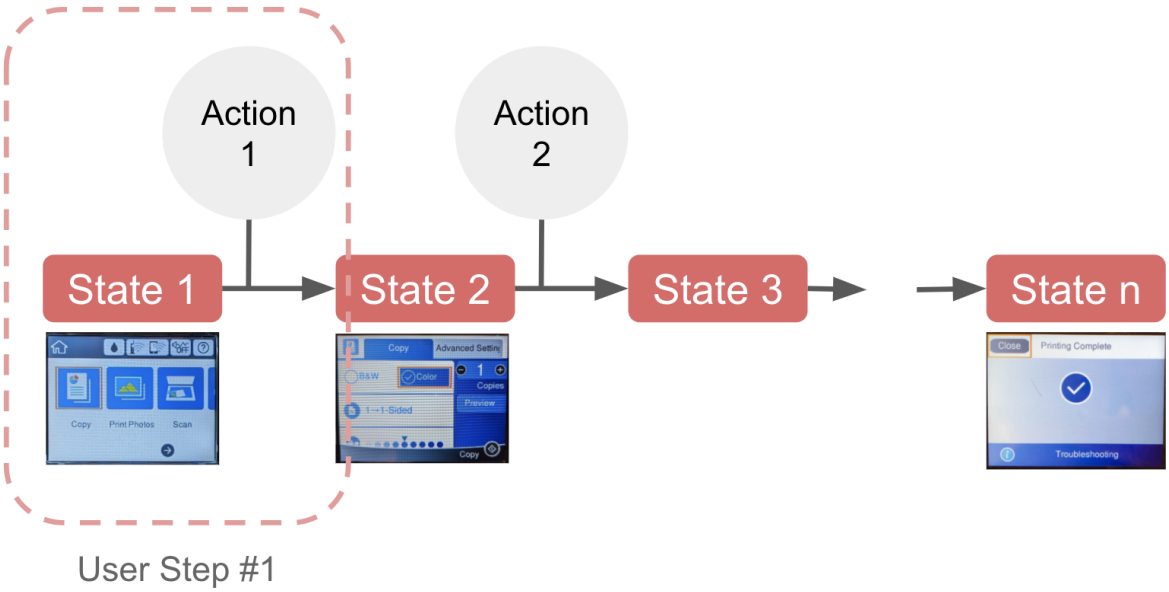

Figure 3.2: Modeling a task on a user interface as an action sequence. An action sequence contains a sequence states (each uniquely identifiable by a reference image) and user actions connecting the states. Each state-action combination is considered a user step.

3.2 Authoring Mode

The authoring mode allows an experienced user to demonstrate multiple different tasks one can perform with the interface, such as copying documents, faxing documents, etc. with a printer. Our system automatically processes the demonstration and asks users for supplemental information of their interactions, thus generating an action sequence of user actions to take for each task on the machine.

3.2.1 Adding Anchor Points

Whenever a user wants to create AR user manuals for a new machine our system has not seen before, the user is asked to take a picture of the entire static area of the front side of the machine. For example, for a printer with digital displays and a control panel with physical buttons, the picture should include the control panel and optionally the machine logo next to the control panel, but not the digital displays. Our system then divides the taken picture of the static area of the front side of the machine into multiple images, each large enough and has enough feature points to be recognized and tracked as an anchor point of the machine. These images will then be used as anchor points to track the machine location for all future interactions with the machine interface. By adding multiple images, our system is always able to identify at least one anchor point in the camera view regardless of a user's hand and finger locations, as long as the camera is pointing to the front side of the machine.

Figure 3.3: Example anchor points for different user interfaces. (a) The printer interface is divided into two anchor points, the left anchor point and the right anchor point; (b) The microwave interface is divided into two anchor points, the top anchor point and the bottom anchor point; (c) The door intercom interface is divided into three anchor points, the left, the middle, and the right anchor points.

3.2.2 Adding Reference Image of Machine State

To record a user's demonstration step by step, our system needs to identify the status of the machine with a unique image of the machine that can only be seen in this current user step, for example, an image of the digital display with a success message indicating task completion, an image of an opened printer top when placing documents to copy with a printer.

After the anchor points are added, the user is asked to take a picture of the machine that's unique to this current step. Our system will then add this image as a reference image of the current step in the current task.

3.2.3 Detecting User Interaction

Once the reference image of the current step is added, the user is asked to proceed with demonstrating the interaction with the interface. Our system then detects and records the user interaction by tracking the user's fingers seen in the phone camera view. Whenever any of the user's fingers start to appear in the camera view, our system starts to record the finger locations relative to the the anchor point in the camera view, i.e. relative to the machine location in the real world. Note that as the user's hand and finger locations change, the anchor point used in the process might also change as different parts of the machine might be blocked by the user's hand during the process. Thus, our system also records the anchor point used for tracking in each frame to reproduce the finger movements in the access mode in the future.

More specifically, for the user action in each step in the authoring mode, our system, frame by frame, stores a sequence of

$$(S_i, A_i, F_i)$$

where

$$A_i = (x_{anchor,i}, y_{anchor,i}, z_{anchor,i})$$

$$F_i = (F_{finger_1,i}, F_{finger_2,i}, ...)$$

and

$$F_{finger_j,i} = (x_{finger_j,i}, y_{finger_j,i}, z_{finger_j,i})$$

For the i-th frame in the user interaction, S_i is the name of the anchor point used in this frame. $A_i = (x_{anchor,i}, y_{anchor,i}, z_{anchor,i})$ is the 3D coordinates of the anchor point in this frame and $F_i = (F_{finger_1,i}, F_{finger_2,i}, ...)$ is the 3D coordinates of user fingers relative to the anchor point in this frame. Note that for each F_i, a maximum of five fingers can be recorded, while not all finger locations are required. As long as one finger appears in the view, the finger location is recorded.

By default, ARKit captures the 3D coordinates of fingers in the absolute coordinate system from camera view, say $F'_{finger_j,i} = (x'_{finger_j,i}, y'_{finger_j,i}, z'_{finger_j,i})$. Thus, our system computes the relative locations of user fingers to the anchor points as

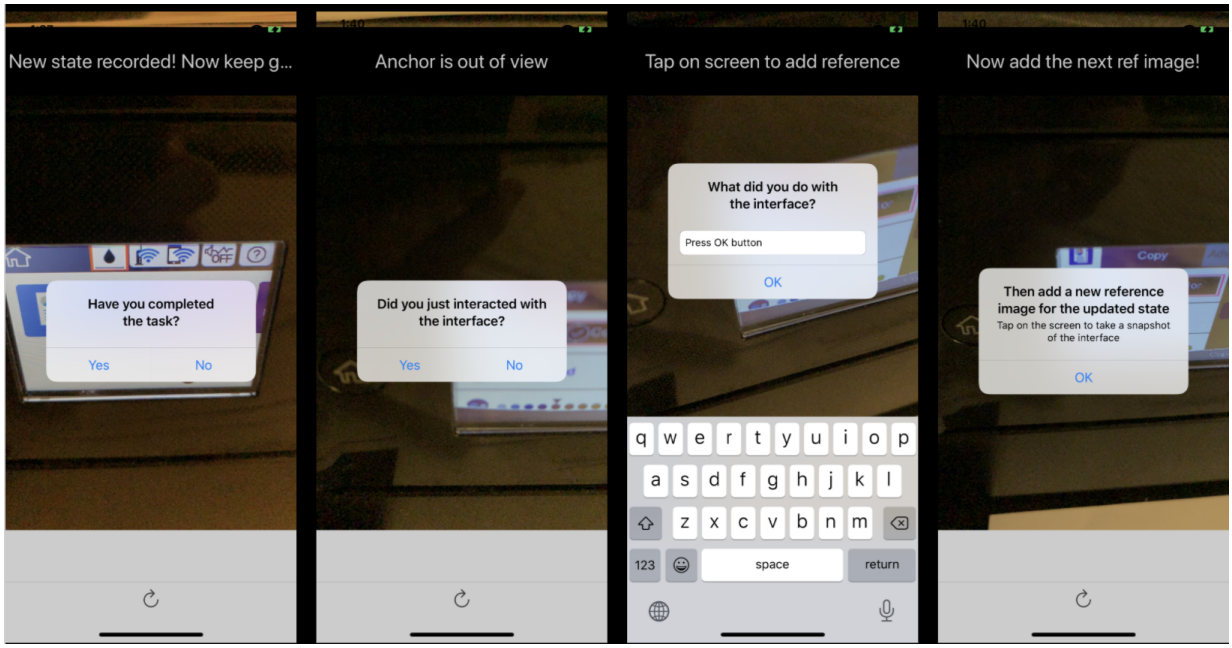

Figure 3.4: The app asking a user for verification of the detected action and asking the user to enter supplementary information describing the interaction.

$$F_{finger_j,i} = F'_{finger_j,i} - A_i$$

Our system can thus reproduce the finger locations later in the access mode by reversing the computation

$$F'_{finger_j,i} = A_i + F_{finger_j,i}$$

3.2.4 Asking for Verification and Supplementary Information

Whenever all the fingers disappear from the camera view, our system infers that the current step of interaction has been completed and asks the user for verification. If the user confirms that the action is completed, they are asked to also input a brief summary of what they did with the interface, for example, pressing the OK button, opening the top cover of the machine, etc, as shown in Figure 3.4. The sequence of finger locations, the corresponding anchor point for each recorded finger location, and the supplementary summary text entered by the user are then stored under the current step (identifiable by the unique reference image the user added) in our system.

Once this is done, our system proceeds to adding a reference image of the updated machine status for the next step in the tutorial, and repeats the process of adding a reference image and detecting user interaction until the user indicates that the task has been completed.

Figure 3.5: The system records the 3D coordinates of the finger locations relative to the anchor point location in the authoring mode, and reproducing the finger location in the access mode based on current 3D coordinates of the anchor point location.

3.3 Access Mode

After generating the sequence of reference images and actions for each task by processing experience users' demonstrations in the authoring mode, our system is then able to guide a new user of the interface to perform these tasks step by step.

3.3.1 Determining Current User Step

Once a new user selects among the saved tasks of the machine interface, our system identifies the current step of the user by detecting existing reference images for the selected task using ARKit. As the reference images have a 1-to-1 mapping to all the steps of the sequence of actions, there should always exist at most one reference image in the camera view. For example, whenever our app sees the home screen of the digital display of the printer, it knows that the user is in the first step of a task. Thus, our system determines the current step with the identified reference image.

3.3.2 AR Simulation of Finger Movements

As discussed in Section 3.2.3, our system stores a sequence of finger locations frame by frame as the user action corresponding to each user step. After our system determines which step the user is currently on, it can then retrieve the corresponding sequence of finger locations, and displays a 3D visual simulation of the recorded finger movements in the AR view on the user's phone screen.

To do that, our system takes the recorded sequence of finger movements, anchor points. For each frame in the sequence, it looks for the specific anchor point for that frame. Also as mentioned in Section 3.2.3, once that anchor point is detected, it reproduces the absolute finger location in the current camera view using the finger coordinates relative to the anchor point, by computing

$$F'_{finger_j,i} = A'_i + F_{finger_j,i}$$

where A'_i is the 3D coordinates of the anchor point S_i in the current camera view, and $F_{finger_j,i}$ is the stored relative finger location in the sequence.

3.3.3 Text and Audio Instructions

As mentioned in Section 3.2.4, in addition to the 3D visual guidance, our system also stores the supplementary text information entered by the experienced user in each user step, for example, ”press OK button” or ”open top cover of printer”. Then in the access mode, our system retrieves the text corresponding to the current step and displays and announces the description text for this step. Thus, our system provides supplementary text and audio guidance in assistance to the AR visual guidance.

Whenever the user completes the action to take for a step, the machine status should change accordingly, thus a new reference image corresponding to the next step of the task should appear in the camera view. Our system automatically captures this change and proceeds to the next step, and repeats the process until the task is completed.

Chapter 4

Design Iterations

4.1 Initial Prototypes

In the early stages of the project, we brainstormed possible uses of augmented reality in assisting users with understanding and interacting with unfamiliar interfaces. We categorized user interaction with interfaces into four different major types: buttons and toggles, such as selecting an option or pressing a key; touchscreen gestures, such as swiping down a touchscreen menu; interactions involving real physical objects, such as inserting bills or swiping credit cards on vending machines. We designed different visual indicators to guide users for these different types of interactions and with different interface elements, including buttons, toggles, sliders, and other physical interactions.

4.1.1 Visual Indicators for Different Interface elements

Buttons and toggles that require a simple click by the user are among the most widely used interface elements, both for static and dynamic interfaces. For the click interaction, we designed an animated circle displayed around the target button or toggle to clickin the AR scene (Figure 4.1a).

User interactions in some interfaces, especially touchscreen interfaces, may also involve gestures. A slider (or swipe action) requires users to tap on certain parts of an interface and then move their finger. For the dragging or swiping interaction, we designed an animated circle repeatedly appears at the initial tap location and then moves towards the target direction (Figure 4.1b).

Some tasks would also require actions other than interacting with interface elements, such as inserting bills or swiping a credit card on a vending machine. In these scenarios, we designed the target area to be highlighted in the scene to raise the user's attention (Figure 4.1c), and then supporting text displayed on the phone screen to specify what the user needs to do, for example, "insert a dollar bill" or "swipe your credit card."

4.1.2 Interface Overlay

Aside from providing feedforward to guide users to perform interactions, we also designed overlays on top of existing interfaces in the AR scene to transform the interface in reality into one

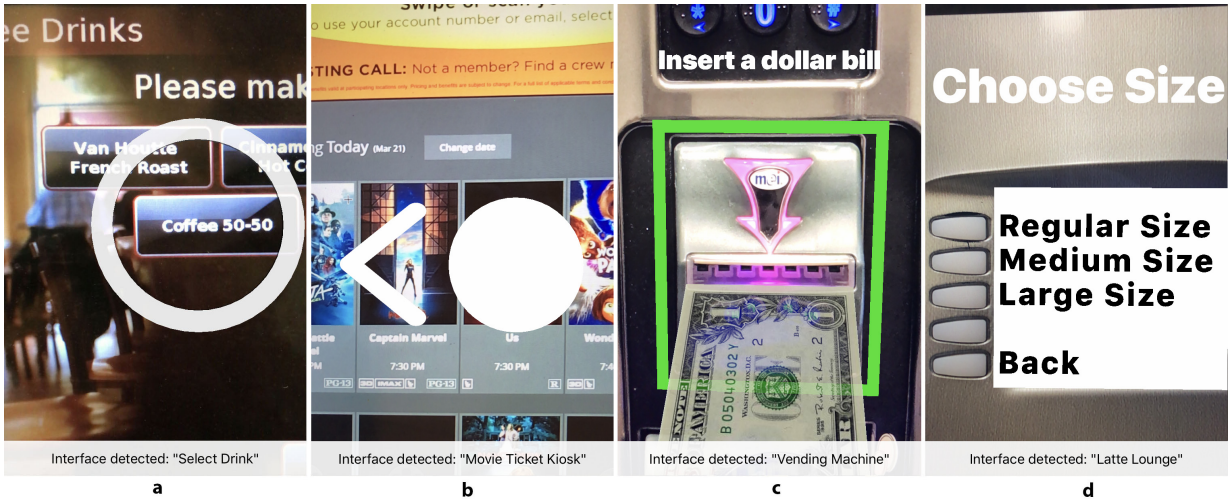

Figure 4.1: Design of visual guidance for different types of interactions: *(a)* On a coffee machine interface, a floating circle is displayed around the target button "coffee 50-50"; *(b)* On a movie ticket kiosk, an animated circle moving towards the target swipe direction is displayed to guide the user to swipe to the next page of movie list; *(c)* On a snack vending machine interface, an animation of inserting bills is displayed with the area highlighted, in order to guide the user to complete the payment; *(d)* On a text-heavy coffee machine, an overlay with bigger fonts and higher contrast is displayed on top of the original interface to make it more readable.

that's more easily readable and understandable by certain user groups. For example, we designed interface overlays using bigger fonts and higher contrast for older adults (Figure 4.1d), covering irrelevant sections to allow users to focus on the parts to interact with, adding descriptive icons or text descriptions, etc.

4.1.3 End-User AR Tutorial with Pre-Specified Action Sequence

In order to put together the different types of visual indicators in an AR user tutorial and test out how well they can guide an end user to complete the tasks, we created an AR tutorial for a printer interface with the action sequence pre-specified. Using the Apple ARKit platform, we developed an AR user tutorial that helps a user to complete a task of copying a document with the printer.

The printing task is composed of the following steps: pressing the OK button to select the copy option from home page; opening the top cover of printer; placing the document to copy; closing the top cover of printer and press the start button; and optionally pressing the HOME button to go back to the home page once the printing job is completed. As shown in Figure 4.2, each of these steps can be uniquely identified by a reference image of the digital display or some other part of the printer, and contains a user action that connects to the next step in the sequence.

Our prototype pre-coded the action sequence in the application, including the reference images, corresponding AR visual indicators and text instructions for each step. As shown in Figure 4.3, at each step, whenever a reference image is found in the camera view, i.e. a state of the printer is detected, our prototype displays the visual indicators in the AR scene on the user's phone camera, and announces the text instruction describing the current step as an auxiliary

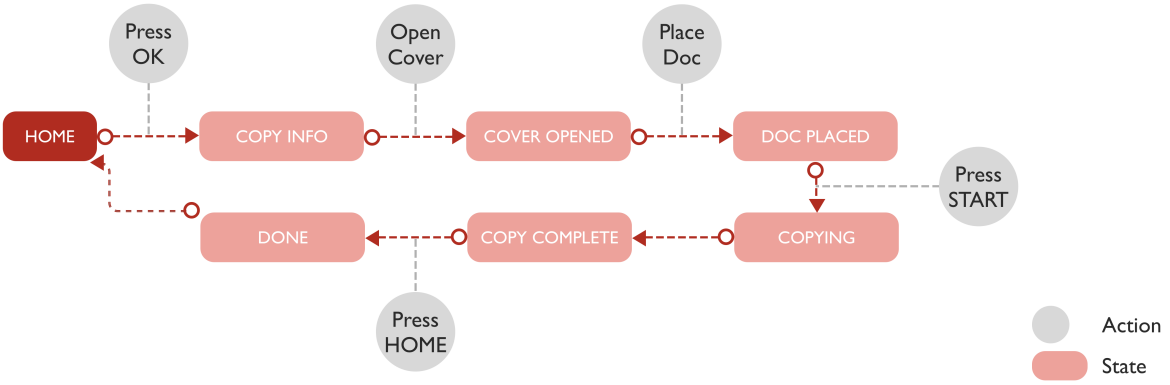

Figure 4.2: Example action sequence containing the states and actions connecting the states for task of copying a document with a printer interface.

guidance for the user. Once the user completes an action, the machine transitions to the next state and thus the reference image for the next step in the task should be detected and the visual indicator is updated accordingly. This process is repeated until the final state is reached, i.e., the user successfully completed the task.

4.2 Preliminary User Study

4.2.1 Think Alouds

With our end-user AR tutorial with a pre-specified action sequence, we ran preliminary user study with 3 users who have not used the printer interface before. We asked each user to complete the task of copying a document, following the AR visual indicators and text and audio instructions in each step. During the process, we asked the users to "think aloud" and verbalize their thought process while interacting with the printer interface using the AR user tutorial.

4.2.2 Feedback and Observations

One major finding of our preliminary user study was that some users got confused by the visual indicators in certain steps of the task of copying a document. More specifically, since some steps of the task involved user actions of opening and closing parts of the printer, our prototype displayed a 3D simulation of the expected movement of these parts of the machine in the AR scene - for example, when a tray of the printer needs to be pulled out, our prototype displayed an animated 3D box simulating the tray moving out from the machine. However, a few users were confused by this visual indicator and indicated that they didn't know which exact part of the machine to interact with.

Another common feedback we received from the users was that holding a phone in their

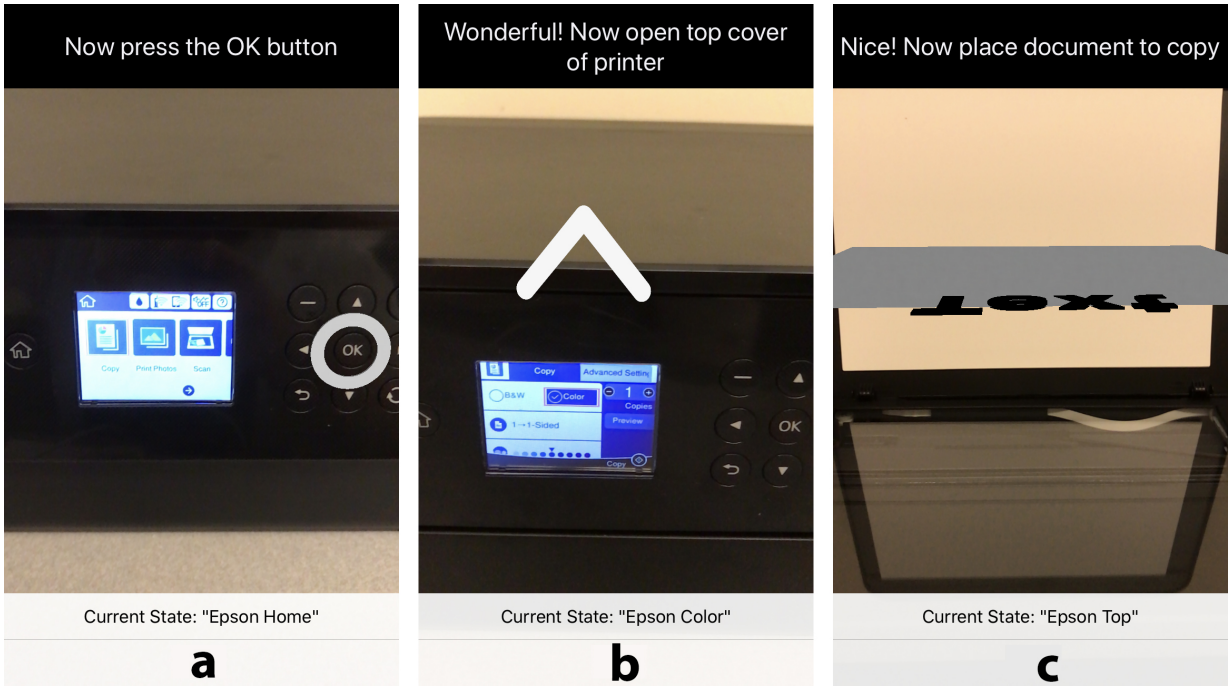

Figure 4.3: Prototype of user tutorial for a printer interface with pre-coded action sequence of copying a document using the printer. (a) A circle is displayed around the button to press in the AR scene when the user needs to press the OK button in the first step; (b) An arrow is displayed around the edge of the top cover of the printer in the AR scene when the user needs to open the top cover in the second step; (c) An animation of a 3D object indicating a piece of paper with text facing down is played in the AR scene when the user needs to place the document to copy. The text instructions on the screen also change accordingly for each step.

hand while interacting with the printer interface was a little overwhelming, especially that they needed to look at the phone screen for visual indicators in the AR scene, while interacting with the machine in the real world. Although we were not able to resolve this in our future prototypes, we will expand on potential solutions to this issue in our discussion and future work section.

4.3 Redesigning Visual Indicators for A More Generalized Authoring Process

4.3.1 Using Simulated Finger Movements as Visual Indicators

Although our prototype of the end-user AR tutorial with pre-specified action sequence received overall positive feedback, the prototype was developed based on a fixed task with the action sequence, reference images and corresponding visual indicators manually inputted. The prototype was also developed especially for the printer interface, thus the visual indicators in that prototype could hardly be authored in a generalizable way.

Because of that, we needed to re-design the visual indicators in our prototype, so that the visual indicators can be automatically generated given user input of the actions. As we hope our authoring tool to work with both physical interfaces and touchscreen interfaces, both flat interfaces as well as 3D machines, designing visual indicators based on machine parts seemed quite challenging as their shape, position, etc. vary a lot across different machines and interfaces.

However, we realized that although there are a variety of different types of user interactions with different types of interfaces, one thing in common is that all of these interactions require a user's hand in the process. Thus, instead of displaying the movement of machine parts in the AR scene as visual indicators, we could record the finger movements of an experienced user interacting with the interface, and then replay the finger movements to guide a new user to access the interface.

4.3.2 Using Anchor Points in Finger Tracking

As we developed prototypes to test out how well simulated finger movements work as visual indicators, we found that naive finger tracking didn't work as the phone camera might move around when an experienced user is demonstrating the interaction. As the 3D finger locations recorded by ARKit are relative to the phone camera position, the accuracy of tracking and replaying user finger movements highly depends on whether users could keep the phone in their hand still.

To resolve this issue, we introduced an anchor point to improve finger tracking stability. By also tracking a static location on the machine, our prototype computes a user's finger positions relative to the static machine location during the authoring process. When a new user is accessing the interface, our prototype tracks the same static location on the machine, and computes the 3D coordinates of simulated finger movements using the relative coordinates saved earlier. In this way, we were able to record and reproduce user finger movements fairly robustly regardless of how the phone camera moves.

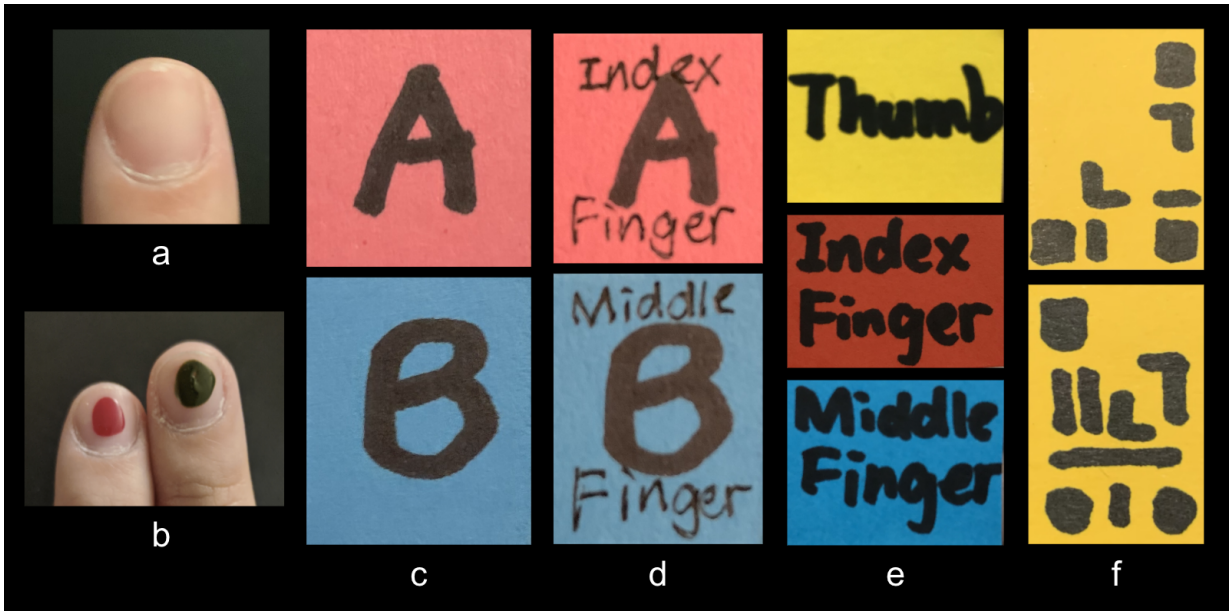

Figure 4.4: Different methods of finger tracking and prototypes of different types of patterns used on finger labels. (a) Bare finger; (b) Finger with nail polish; (c) Single letters; (d) Single letters with small text; (e) Large text; (f) Pseudo QR code.

Pattern Type	Whether or Not Working	Required Size of Label	Robustness
Single Letters	No	-	-
Single Letters with Small Text	Yes	Regular (about nail size)	Not robust
Large Text	Yes	Large (about twice the nail size)	Robust
QR Code	Yes	Regular (about nail size)	Not robust
Pseudo QR Code	Yes	Regular (about nail size)	Robust

Table 4.1: Different types of patterns used on finger labels and their performance in finger tracking

4.4 Prototyping Finger Tracking

We experimented with multiple different finger tracking methods, such as detecting convex hull using OpenCV [5], hand detection using OpenPose [6] and MediaPipe [13], or transfer learning on top of trained object detection models.

However, these methods have a few common drawbacks that make them not ideal for our use case:

- Only providing 2D coordinates by processing the 2D camera view
- Requiring the entire user's hand to be in the camera view, in order to detect all the junctions for accurate finger tracking
- Relying heavily on colors or shapes of fingers and can easily have false positives

Thus, we decided to image tracking of user finger tips for finger tracking in our prototypes. Although this method might not be as user-friendly compared to more intelligent finger tracking methods, the robust image tracking of ARKit allows us to fairly stably track and record finger locations in our prototypes.

We first experimented with using images of bare fingers and fingers with nail polish as reference images for fingers. However, as the image tracking of our chosen platform only works with flat 2D images, this method didn't work out.

Then we experimented with adding flat "labels" to user fingers and tried these patterns for the finger labels (as shown in Figure 4.4 and Table 4.1):

- **Single letters, such as "A", "B", etc.** This did not work out, as the patterns are too simple to be detected by ARKit.
- **Single letters with some small text surrounding it** This worked when the phone camera is always close enough to user fingers. However, as the camera moves a little far away, the texts become too small to be clearly seen by the phone camera.
- **Large text of short phrases, such as "index finger", "middle finger", etc.** This worked quite well and the detection was very stable. However, the labels needed to be made very large in order to include all the text.
- **QR Code** This worked when the phone camera is always close enough to user fingers. However, as the camera moves a little far away, the pixels in the QR code become too small to be clearly seen by the phone camera.
- **"Pseudo QR Code"** We finally experimented with drawing a QR-code-like pattern to the labels using black markers. This worked quite well and can allow user fingers to be accurately and stably tracked even when the phone camera moves a little farther from the fingers.

4.5 Refining Anchor Point Selection

Our prototype first used a single anchor point for finger tracking throughout the process. Specifically for the printer interface, we used the printer logo next to the control panel on the front side of the machine. However, we found that the anchor point constantly disappears from the camera view as we tested out the prototype. The two main reasons of the single anchor point disappearing include the following:

- When an experienced user demonstrates the interaction, their hand might hover over the anchor point, thus blocking the anchor point from the camera view.
- When an experienced user interacts with some part of the machine that's far away from the anchor point, they move the phone camera to follow their hand movement, so the anchor point gets out of the view.

To resolve this issue, we decided to use multiple anchor points instead of one single anchor point to assist with finger tracking. By dividing up the entire control panel of an interface into grids large enough for recognition and uses them as multiple anchor points, our prototype can then guarantee that there always exists some anchor point in the camera view as long as users'

hands don't move too far away from the front side of the machine. During the authoring process, our prototype uses any identified anchor point to compute the relative finger positions, and records that anchor point. When a new user is accessing the interface, our prototype looks for that anchor point in the camera view, and starts to reproduce the finger movements once that anchor point is found in the view.

Chapter 5

Results and Discussion

5.1 End-To-End Demonstration

Due to COVID-19, we were not able to run in-person user studies in labs. Thus, to demonstrate our authoring tool, we recorded a video of an end-to-end demonstration of how our authoring tool works.

In this demonstration, an experienced user creates an interactive tutorial on how to copy a document with the printer using our phone app. Then a new user comes to the machine and copies a document following the guidance on the phone app. A full video of our demonstration can be found online: https://youtu.be/JXorzmC1xIM.

5.1.1 Authoring Mode

Figure 5.1 demonstrates one step of the authoring process: an experienced user adding the first step of the copying task with the printer to the AR interactive. At the beginning of the demonstration, the experienced user is asked by the text instruction on the phone app to ”Tap on the screen and add a reference image”. Following the instruction, the user takes a picture of the home screen on the digital display of the printer interface. This picture is then automatically added as a reference image, and the phone app pops up a message asking the user ”Have you completed the task?” to determine whether or not to terminate the authoring process. The user selects ”no” indicating that the task is not completed. Then the app guides the user to proceed with interacting the interface by updating the text instruction on the screen to ”New state added! Now keep going.”

The user then starts with the first step of the copying a document, which is selecting the copy option on the home screen of the printer. As this option is the default selection on the home screen, the user only needs to press the OK button to select this option. As the user moves their finger to press the button, our phone app records the finger movements. Once the user finger moves outside of the camera view, our phone app pops up a message asking the user ”Did you just interact with the interface?” to confirm with the user whether they’ve finished this step. The user then selects ”yes” to confirm with their action. After that, the phone app pops up a text entry box asking the user ”What did you do with the interface?”, and the user enters ”press OK

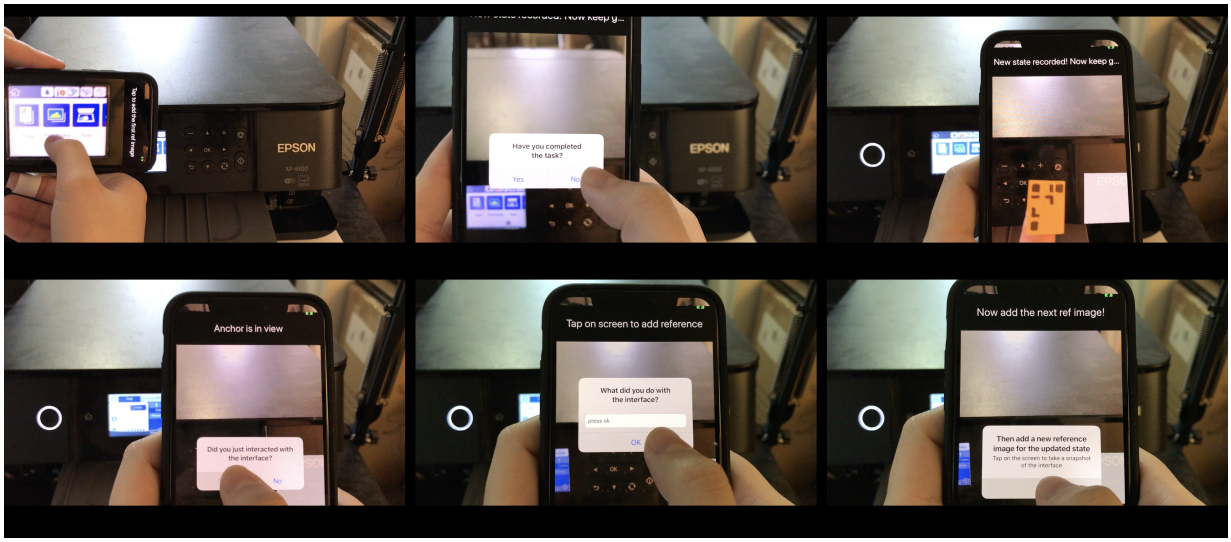

Figure 5.1: Adding a step to the AR user tutorial under the authoring mode. (a) The app asks the user to tap on the screen to take a picture as the reference image for the current step; (b) The app confirms whether the task is completed; (c) The user demonstrates the action of the step: pressing the OK button; (d) The app verifying if the user interacted with the interface; (e) The app asks the user for a short description of what they did; (f) The app asks the user to proceed to adding the next reference image.

button". Thus, this short summary is stored to our system corresponding to this current step, and can later be used as assisting text and audio instructions while guiding a new user in the access mode.

As the user presses the OK button, the digital display of the printer transitions to the next screen, which is the copy info page. After the short summary text is entered, our phone app pops up another message asking the user to "Add a new reference image of the updated state", and the user then takes a picture of the updated digital display screen of the printer. By this, our system knows that the tutorial is entering the next step, and then repeats the process of recording user finger movement, asking for verification of the user action and additional text description of the interaction, which is opening the top cover of the printer in this step.

Since the digital display in this step is not changed, the user then takes a picture of the inner structure through the glass from the top of the printer as the unique reference image of this step - this can only be seen when the top cover of the machine is open. Still, the user is asked to then interact with the interface, which is placing the document, then verifies the interaction and enters additional text describing the step.

Similar to the previous step, the user then takes a picture from the top of the printer - as the document is placed on the document, the glass is partially covered by paper, thus the reference image can be distinguishable from that of the previous step. Then the user proceeds to the next step and closes the top cover of the printer and presses the START button. This action is also recorded by our phone app, and it still pops up messages to verify the interaction and ask the user to briefly describe the step.

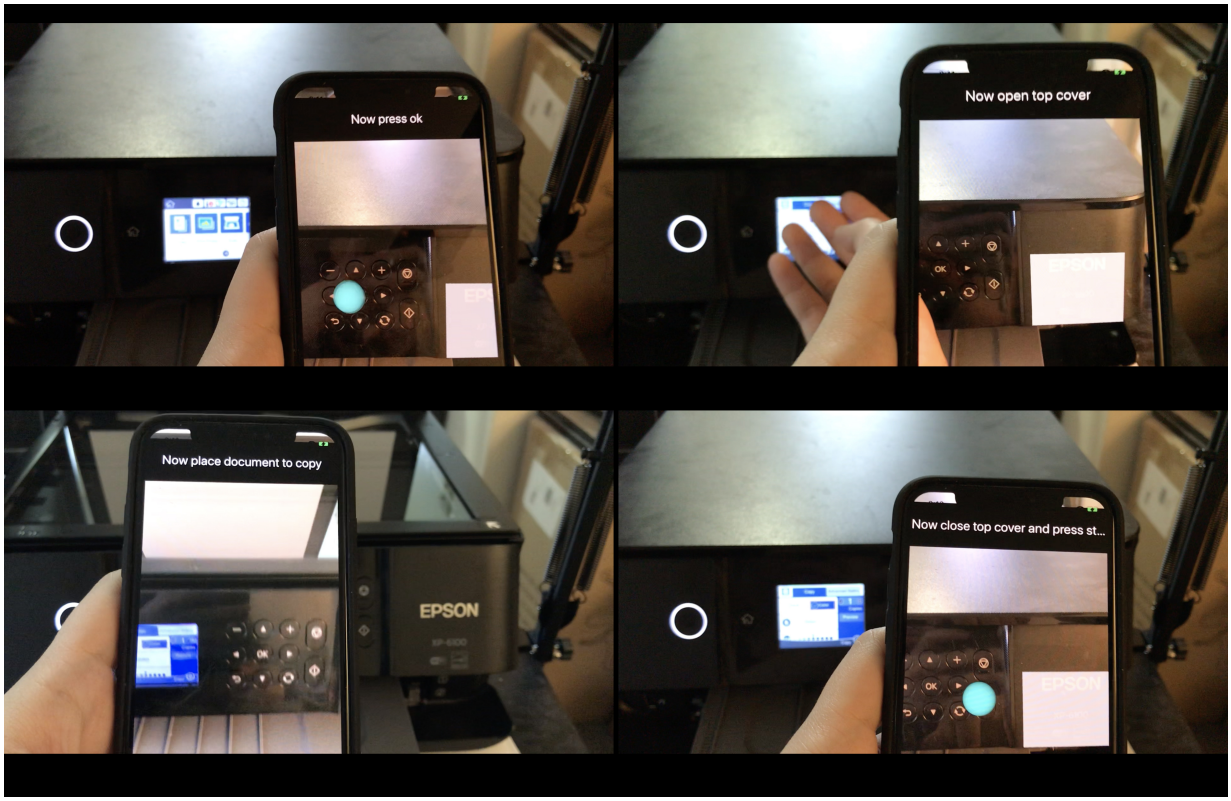

Figure 5.2: The app guiding a new user to copy a document with the printer under the access mode. (a) The app recognizes the home screen and displays a sphere simulating user finger pressing on button OK to go to the next step; (b) The app asks the user to open the top cover of printer; (c) The app asks the user to place the document to copy; (d) The app asks the user to close the top cover and press the START button, and displays a sphere simulating the user finger pressing on button START to start copying.

Finally, after the START button is pressed, the printer starts to copy the document. Once the document is successfully copied, the display screen of the printer changes to a success message indicating that the document is copied. The user then takes a picture of the updated display screen as the final reference image, and when our phone app pops up a message asking whether the task is completed, the user selects ”yes” indicating that the demonstration of all the steps in the copying task has been completed.

5.1.2 Access Mode

As the authoring process is completed, our phone app then transitions to the access mode, which can guide a new user to copy a document using this printer.

Figure 5.2 demonstrates one step of the app guiding a new user to complete a copying task with the printer using AR visual indicators and supportive text. As the user moves the phone camera close to the printer interface, the home screen of the digital display appears in the camera view. Our phone app thus knows that the user is on the first step of the process. Thus, it displays

the text descriptions of the step which was entered by the experienced user: ”Press OK button”. In the meantime, it announces this text aloud to the user. The phone app then keep displaying a text of ”Anchor point out of view” until the anchor point appears in the camera view. As the user moves around the phone camera and the anchor point appears, our phone app starts to display an AR simulation of the user’s finger pressing the OK button - a cyan sphere moving close to the OK button then moving away, which indicate the user finger movement in this step. The user follows this demonstration of finger movement as well as the text and audio instructions, and presses the OK button.

After the OK button is pressed, the digital display updates to the copy info screen, which is then captured by the phone camera. Our app thus knows that the user now completes the action in the first step and enters the second step of the process. Thus, it updates the text instruction on the screen to ”Open top cover of printer”, and announces the text. Our app also displays an AR simulation of the user’s fingers opening the top cover of the printer as the anchor points appear in the view, and the user follows the guidance and opens the top cover of the printer.

Similarly, the user moves the phone camera around and our app sees the opened top cover of the printer, thus knowing that the user enters the next step. Our app then displays and announces the corresponding text instructions, ”Place document to copy”, guiding the user to place the document to copy on the printer.

After the document is placed, the view from the top of the printer changes, and our app knows this as the camera view now matches with the reference image of the next step. The text and audio instructions are thus updated to ”Close top cover and press START.” As the user moves the phone camera back to the front side of the machine, the anchor points appear again in the view, thus our app starts to display an AR simulation of the user finger moving to the START button and the moving a way. The user follows the guidance, closes the top cover of the printer and presses the START button, so that the printer starts copying.

Finally, after the copying job is completed on the printer, the digital display changes to the success screen, which is also seen by our app from the phone camera view. Our app then knows that the final step is entered, and the user has successfully completed the copying task, thus tells the user ”Congratulations! You’re done.”

With that, our demonstration includes an end-to-end process going from an experienced user demonstrating the task creating the AR tutorial, to a new user accessing the interface following the guidance in the AR tutorial. By this demonstration, we show that our system can successfully create interactive AR user tutorials to guide new users to access complex user interfaces, such as the printer interface we chose for demonstration.

5.2 Evaluating Multiple Anchor Point Selection for Different Interfaces

As described in Section 3.1 and Section 4.4, we used multiple anchor points to locate a machine in the 3D space. In the authoring mode, whenever one of the anchor points for a machine appears in the camera view, our system tracks and records user finger positions relative to this anchor point; In the access mode, our system creates a 3D simulation of user finger movements based

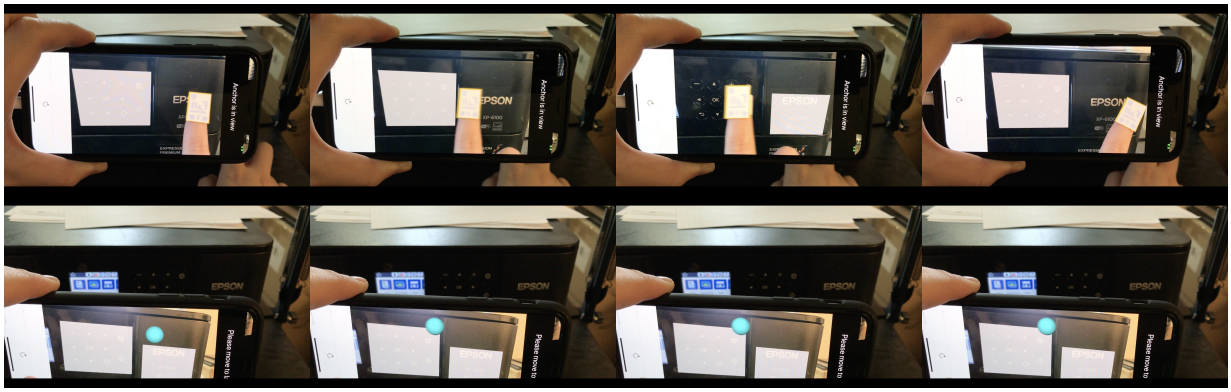

Figure 5.3: Printer interface: a sequence of snapshots when user moves finger around under the authoring mode and a sequence of snapshots when the app reproduces a simulation of the movement. The user first moves their finger from the right to the left, and then from the left to the right. The selected anchor point (marked by the white box) switches from the left anchor point to the right anchor point, and then back to the left anchor point.

on the current positions of these anchor points. In this way, our system guarantees that there always exists an anchor point in the camera view in both modes, and can thus accurately track and reproduce finger movements for the actions to take in each step.

To evaluate our anchor point selection mechanism, we tested our prototype with three different interfaces we found at home: the printer interface, the microwave interface, and the door intercom interface. All these three interfaces have a control panel with physical buttons, while they vary in size, shape and layout in the control panel.

For each interface, we moved one finger around to cover different parts of the interface, i.e. different anchor points. For both the authoring mode and access mode, we used a white rectangle to mark the detected anchor points in the camera view as well as the finger location. Then in the access mode, we used a sphere in the AR scene to simulate the finger movements. From the snapshots of the app (Figure 5.3, Figure 5.4, Figure 5.5) during our testing process, we can see that our method of using multiple anchor points of the machine successfully tracked and later reproduced finger movements as different anchor points were covered by user fingers.

5.3 Discussion

5.3.1 Multi-Modal Feedback

We have received very positive comments on the multimodal feedback in our AR interactive user tutorial during our preliminary user studies. As the text and audio feedback in the access mode provide auxiliary instructions to support the AR visual guidance, users found these feedback helpful in helping them understand the action to take in each step. These instruction especially helped when the finger movements were too vague for a new user to identify what action to take with the interface, or when the fingers move too far away from the anchor points, and thus can no longer to be tracked by the system.

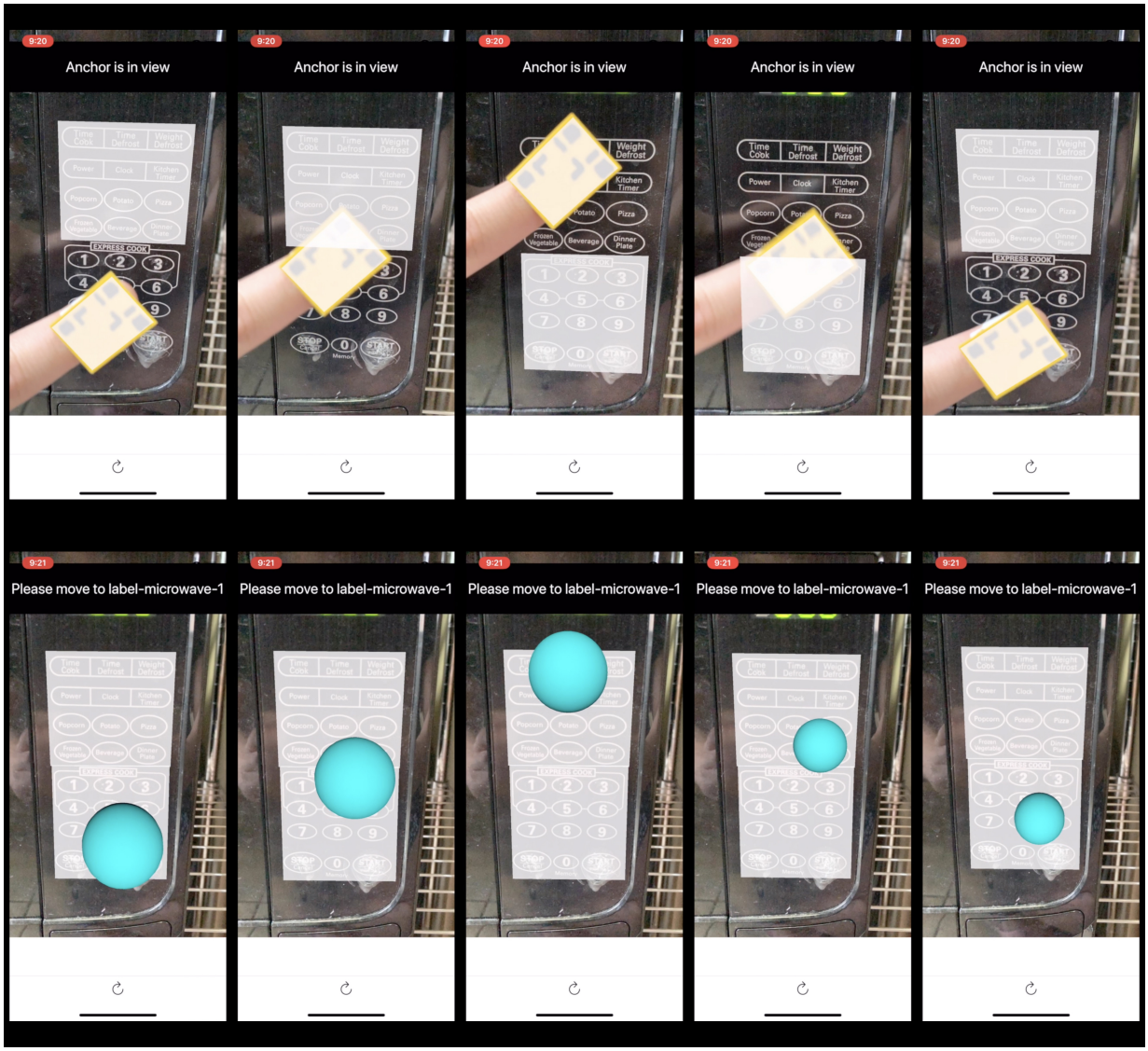

Figure 5.4: Microwave interface: a sequence of snapshots when user moves finger around under the authoring mode and a sequence of snapshots when the app reproduces a simulation of the movement. The user first moves their finger from the bottom to the top, and then from the top to the bottom. The selected anchor point (marked by the white box) switches from the top anchor point to the bottom anchor point, and then back to the top anchor point.

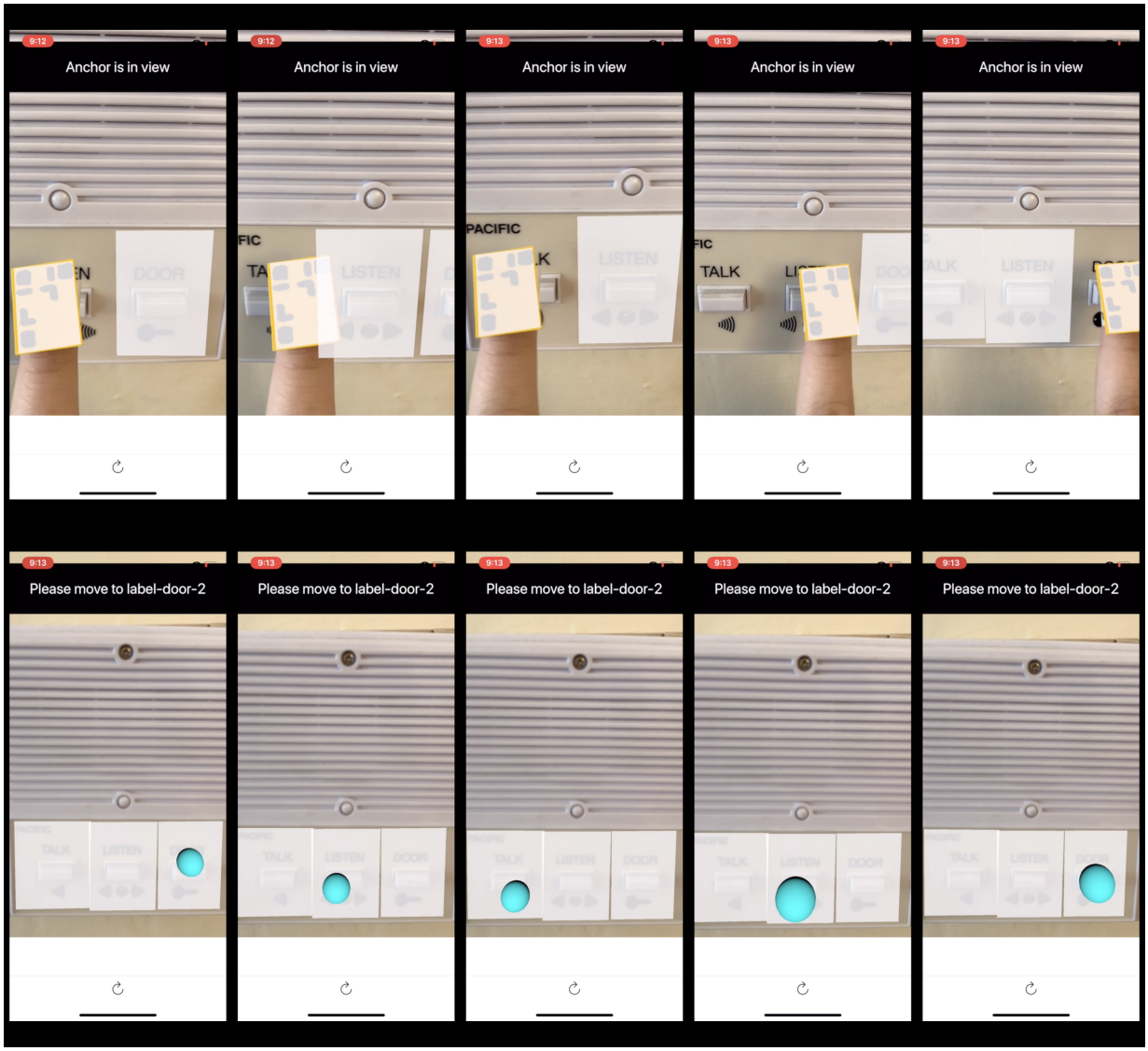

Figure 5.5: Door intercom interface: a sequence of snapshots when user moves finger around under the authoring mode and a sequence of snapshots when the app reproduces a simulation of the movement. The user first moves their finger from the right to the left, and then from the left to the right. As the camera moves, different subsets of the anchor points appear in the view. The selected anchor point (marked by the white box) switches from the right anchor point to the middle anchor point and then the left anchor point, then back to the middle and then right anchor point.

5.3.2 Trade-Offs in Finger Tracking

As the ARKit platform currently does not support finger detection in 3D, we are using a "pseudo QR code" to stick onto user fingertips to track user fingers in the authoring mode. Each finger has a unique "pseudo QR code" and these images are added to the reference images of the authoring tool and can be then detected and accurately located by the application. Due to the current limitations of the ARKit platform, our "pseudo QR codes" still need to be reasonably large to be stably detected and tracked by ARKit, and the larger the "pseudo QR codes" are, the more stable ARKit can track them. However, sticking large labels to user fingertips would affect the user experience, and we tried to size down these "pseudo QR codes" to as small as about the area of a user fingernail, so that they won't negatively affect the user experience too much.

5.3.3 Anchor Points for Interfaces without Static Components

Seemingly, our system only works with machines with static components on the interface, for example, a physical control panel or a large printed label. However, our system, with slight modifications, can also work on completely dynamic interfaces such as touchscreen interfaces. Although we are currently only using static components of the machine interface as anchor points, we could potentially also allow reference images of each user step to be used as anchor points. As the reference images must appear in the camera view for our system to recognize the current step, they can be seen as a temporary "static component" within the time frame of this user step. Thus, our system can track user finger locations relative to the reference image of the current step, which still allows our system to accurately track and reproduce user finger locations.

5.3.4 Limitations

As mentioned in Section 6.2, the current finger tracking method in our prototype requires paper labels to be stick to user fingers for accurate tracking. These labels (especially when they are larger than nail size) can affect the user experience, and make it not as natural as the user demonstrating the interactions with their bare fingers. However, we believe that this issue could be easily resolved as hand and finger tracking algorithms develop and more features supported in the Apple ARKit platform in the future.

Another limitation of our work is that the anchor point selection mechanism still requires the anchor points to be relatively close to the points of interactions on every step. For example, if we use the front side of a printer as the anchor points, then if the user fingers move too far to the top to open the top cover of the printer, most of the finger movements won't be captured by the camera. In our prototype, such limitation is partly made up by the multi-modal feedback in the access mode - the user can just follow the text and audio instructions when they don't see AR simulations of finger movements on the screen.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

We have presented an authoring tool for creating interactive AR user tutorials by demonstration based on the Apple ARKit platform. With the authoring mode and the access mode, our system guides experienced users to add anchor points of the machine and reference images for each step, then to demonstrate the action to take at each step. By tracking and recording finger locations in each action taken by the experienced user, our system is able to reproduce an AR simulation of the finger movements in the access mode, thus helping new users to access the interface. With our end-to-end demonstration of the authoring-access process and evaluations of our anchor point selection mechanism, we show that our system can be a useful tool in creating interactive AR user tutorials that make complex user interfaces easier to use for a general user.

6.2 Future Work

6.2.1 In-Person User Studies

Due to COVID-19, we were not able to run in-person user studies in labs, and thus used a video recording to demonstrate the effectiveness of our authoring tool. For the future, we would like to recruit real users to test out the authoring tool in person. We would like to more thoroughly evaluate the user experience under the authoring mode and the access mode, and the quality of the AR user tutorial created by participants - whether or not the AR interactive user tutorial is able to help new users successfully interact with a complex user interface they have not seen, and compare the AR interactive user tutorial with traditional tutorial systems such as user manuals.

6.2.2 More Automated and Multi-Modal User Action Detection

Moving forward, we would like to add more automation to the process of the user action detection. For example, instead of asking the experienced user to take pictures of reference images for each step, we can potentially parse the video stream to automatically select and crop the captured images from the camera view and add them as reference images.

We would also like to try multi-modal recognition in the authoring mode. Aside from recognizing reference images of each step, our system could potentially also determine the current step a user is on by capturing audio or other forms of feedback from the machine interface, such as a beeping sound after clicking a button.

6.2.3 Better Usability

One thing we would like to add to our system is error recovery. Although our system now supports creating an AR user tutorial by demonstration and guiding a new user to access the interface, a user cannot go back to modify already-entered demonstrations and added user steps if they were added by mistake. Moreover, if a user accidentally takes the wrong action which transitions the interface to a state our system has never seen before, our system can potentially also guide the user to revert the wrong steps and go back to the main sequence of actions in the task.

Additionally, we can potentially port the application from mobile to AR headsets, as currently users need to hold their phone in one hand while interacting with the machines. With AR headsets, users will have both hands free, and can potentially have a better experience when creating the AR user tutorial and using it.

Bibliography

[1] Apple Inc. Get ready for the latest advances in augmented reality., 2019. URL `https://developer.apple.com/augmented-reality/`. 1.2

[2] Ronald T Azuma. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, 6(4):355–385, 1997. 2.1

[3] John B Black, John M Carroll, and Stuart M McGuigan. What kind of minimal instruction manual is the most effective. *ACM SIGCHI Bulletin*, 18(4):159–162, 1986. 2.2

[4] Jonas Blattgerste, Benjamin Strenge, Patrick Renner, Thies Pfeiffer, and Kai Essig. Comparing conventional and augmented reality instructions for manual assembly tasks. In *Proceedings of the 10th international conference on pervasive technologies related to assistive environments*, pages 75–82, 2017. 2.3

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 4.4

[6] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018. 4.4

[7] Tom Djajadiningrat, Pei-Yin Chao, SeYoung Kim, Marleen Van Leengoed, and Jeroen Raijmakers. Mime: An ar-based system helping patients to test their blood at home. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, pages 347–359, 2016. 2.3

[8] Tom Djajadiningrat, Patray Lui, Pei-Yin Chao, and Christian Richard. Virtual trainer: A low cost ar simulation of a sudden cardiac arrest emergency. In *Proceedings of the 2016 ACM conference on designing interactive systems*, pages 607–618, 2016. 2.3

[9] Jesús Gimeno, Pedro Morillo, Juan Manuel Orduña, and Marcos Fernández. An easy-to-use ar authoring tool for industrial applications. In *Computer Vision, Imaging and Computer Graphics. Theory and Application*, pages 17–32. Springer, 2013. 2.4

[10] Anhong Guo, Junhan Kong, Michael Rivera, Frank F Xu, and Jeffrey P Bigham. Statelens: A reverse engineering solution for making existing dynamic touchscreens accessible. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 371–385, 2019. 2.2

[11] Michael Leitner, Özge Subasi, Norman Höller, Arjan Geven, and Manfred Tscheligi. User requirement analysis for a railway ticketing portal with emphasis on semantic accessibility for older users. In *Proceedings of the 2009 International Cross-Disciplinary Conference*

on Web Accessibililty (W4A), pages 114–122, 2009. 1.1

[12] Rock Leung, Charlotte Tang, Shathel Haddad, Joanna Mcgrenere, Peter Graf, and Vilia Ingriany. How older adults learn to use mobile devices: Survey and field investigations. *ACM Transactions on Accessible Computing (TACCESS)*, 4(3):1–33, 2012. 2.2

[13] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019. 4.4

[14] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Telemanipulator and telepresence technologies*, volume 2351, pages 282–292. International Society for Optics and Photonics, 1995. 2.1

[15] Tam V Nguyen, Bilal Mirza, Dorothy Tan, and Jose Sepulveda. Asmim: Augmented reality authoring system for mobile interactive manuals. In *Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication*, pages 1–6, 2018. 2.4

[16] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256, 1990. 1.1

[17] Hartmut Seichter, Julian Looser, and Mark Billinghurst. Composar: An intuitive tool for authoring ar applications. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 177–178. IEEE, 2008. 2.4

[18] Maximilian Speicher, Brian D Hall, and Michael Nebeling. What is mixed reality? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2019. 2.1