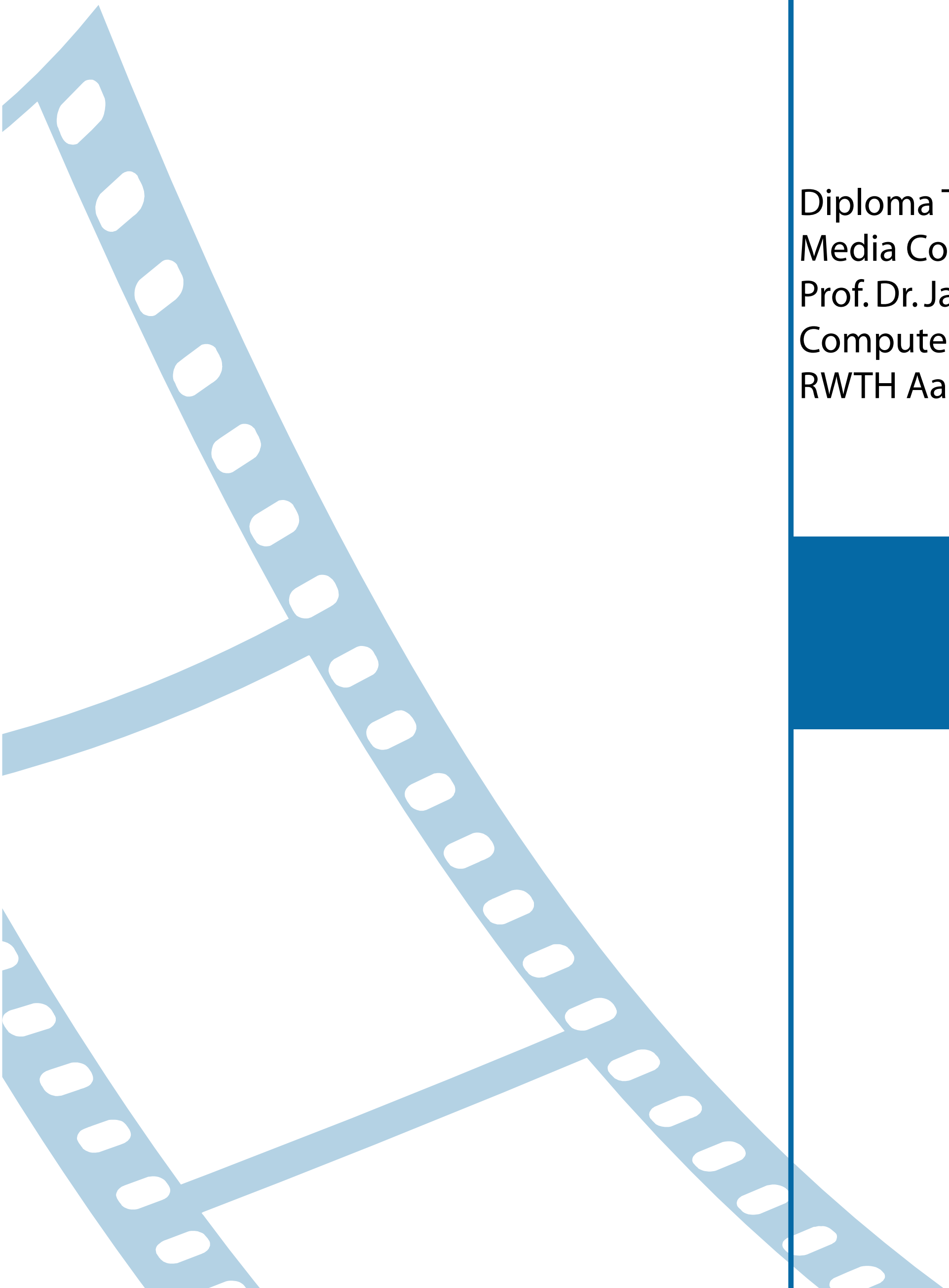
Exploring Mobile Augmented Reality Instructions to Assist Operating Physical Interfaces



Diploma Thesis at the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Can Liu

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Stéphane Huot

Registration date: 29.08.2011
Submission date: 16.01.2012

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

$Aachen, January 2012$
Can Liu

Contents

List of Figures

List of Tables

Abstract

People often need guidances for operating physical devices such as work equipment or home appliances. Traditional manuals provide all the technical data and instructions for operating such machines, but it is sometimes difficult for casual users to find the solution to a problem or to take note of a particular operation. In this work, we propose to provide users with customizable Augmented Reality (AR) instructions on hand-held devices, in order to facilitate their everyday use of physical appliances. This concept is explored with several prototypes and example applications, as an alternative to existing techniques.

Text instructions could be difficult to use for complex manipulations as they require to be interpreted before to operate the physical device. While pictures instructions simplify the spatial localization of the operations, their static nature makes it difficult to present dynamic operations. Sequences of pictures also miss to link between different steps. Conversely, AR techniques could display a graphical, dynamic and continuous instruction as an overlay on top of the real objects. Such in-place guidance can reduce division of users' attention between the instructions and the objects to manipulate. However, the major drawback of AR in this case is that the virtual content can occlude the physical objects, impairing users' perception of the result of their physical actions. Thus, we propose to add real-time feedback on AR overlay to reflect the status of physical objects, in order to compensate the weaken feedback in AR systems.

A controlled experiment is conducted to evaluate the benefits of adding feedback to AR. AR instructions with and without feedback are compared with text and picture instructions. The results show that AR with feedback significantly outperformed other techniques. Conventional AR performs as well as picture, and text reveals to be the less efficient instruction method. The observations and participants' qualitative assessments also provide some insights about the usability and acceptance issues of AR techniques.

This work is a first step for AR instructions to go out of professional domains and reach a larger audience. By showing the benefits of real-time feedback when manipulating physical objects, it highlights the link between AR and Ubicomp Computing technologies, and suggests opportunities to improve AR techniques.

Acknowledgements

First of all I thank one of my advisors Prof. Stéphane Huot for his guidance and tremendous help through this work. Thank him for giving me the opportunity to work in an excellent group, where I learned a lot about research.

Thank Jonathan Diehl for his enormous help and suggestions, which always cleared my mind for finding solutions when I got trapped in problems. From him I learned how to be organized and efficient at work. I also thank Prof. Jan Borchers for all his support in this work and for providing me the chance to work and study in Media Computing Group, which is a great place to learn.

A lot of thanks to Prof. Wendy Elizabeth Mackay for her important advices for the experiment design and all her delightful suggestions. Thank Prof. Michel Beaudouin-Lafon for polishing my paper, which became a success eventually.

Thank Clément Pillias, Pierre Rossinès, Theophanis Tsandilas and David Bonnet for their help and suggestions during the time in France. Thank other members at Situated Interaction Lab who gave feedback during the user study.

Thanks to my family that supported my study from thousands of miles away. In the end, a special thanks to Liude Hao for his encouragement for me to study abroad. Without him I would not have changed my life this way.

Conventions

Throughout this thesis we use the following conventions.

Important terms are written in *emphasized* typeset.

Source code and implementation symbols are written in typewriter-style text.

`myClass`

The whole thesis is written in American English.

Chapter 1

Introduction

Operating machines are hard with text and picture instructions

While operating physical interfaces such as those found on machines, people often need instructions to guide them through the tasks. Common instructions are made of text or pictures. Text instructions need to be interpreted in order to get all the information about an operation: where to find which object as well as how to operate it exactly. Pictures provide better spatial mappings that facilitate the retrieval of physical objects. However, dynamic operations are hard to be visualized with static pictures. Moreover, a task often requires multiple pictures to explain so that the instruction cannot be viewed continuously. Therefore, operational tasks are hard to perform following instructions in traditional mediums.

Cognitive load for switching attention

According to the cognitive theory from [Sohlberg and Mateer, 1989], manipulating objects while following instructions is a typical "Alternating Attention" task. It requires the switches between two complementary sub-tasks. Especially for long or complex instructions, users might have to memorize some information, search for the objects, and then return to the instruction to continue. Repeatedly switching between these sub-tasks can be highly demanding.

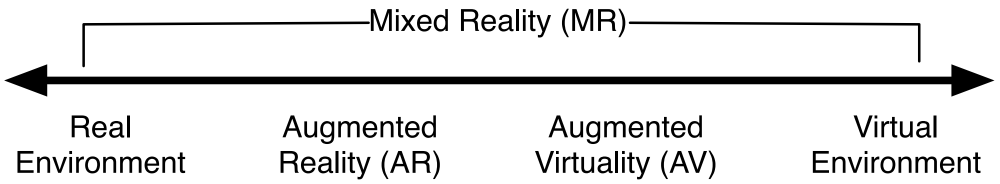


Figure 1.1: Milgram's Reality-Virtuality Continuum

1.1 Augmented Reality

Emergence of AR

Several decades ago, [Sutherland, 1968] created the first *head-mounted display (HMD)* which presented the user with a perspective image as he moved. Twenty years later, the phrase *Augmented Reality (AR)* was coined by [Caudell and Mizell, 1992] to refer to technologies that *augment* users' visual field with information required to perform the current task. In 1997, one of the most accepted definitions of AR was suggested by [Azuma, 1997] as following:

Definition:
Augmented Reality

AUGMENTED REALITY:
Systems that
- combine real and virtual,
- are interactive in real time,
- are registered in 3-D.

As [Milgram and Kishino, 1994] pointed out, AR is placed at a certain position on a continuum of Mixed Reality (MR) according to how much of the user environment is generated by computer, see Figure 1.1.

Early Mobile AR systems for outdoor navigation and 3D books

The emergence of *Mobile Augmented Reality* (Mobile AR) led to a fast development of this technology. The Touring Machine [Feiner et al., 1997], as the first Mobile AR system, provided virtual scene overlaid to real environment. The virtual content was updating according to the user's GPS location and face orientation. It was first used for exploring the urban environment, then further used to develop a restaurant guide that presented information sheets for nearby restaurants with practical informations [Bell et al., 2001]. The MagicBook by [Billinghurst et al., 2001] presented an "augmented" book that seamlessly transported users between reality and virtuality. Users viewed 3D virtual objects attached to certain content in the book through a pair of glasses. Many application domains of Mobile AR, such as maintenance, advertisement, education and entertainment, have been further explored.

3 types AR displays: See-through, Projection-based, Hand-held displays

As summarized by [Feng et al., 2008], there are three main ways to display AR content. One includes the see-through HMD displays, which have been carried on from the early work. Another type refers to projection-based displays, which project registered digital content directly into the real world objects. As an example, iLamps by [Raskar et al., 2005] presented the object augmentation provided by a hand-held projector-camera system. The projected display could be self-configured in response to the geometry of display surface. The third type consists of hand-held displays, such as mobile phones and tablet PCs. They often act as a *Magic Lens* [Bier et al., 1993], where people can see digital information aligned with physical objects through them.

Augment mainly with visual or sound media

Although all AR techniques present digital media that is registered in the physical environment, the media format can be varied. While the major "augmenting" approach is to add visual content to the physical world, some existing systems use sound to provide an augmented experience, such as the work from [Bederson, 1995] and CORONA [Heller and Borchers, 2011]. There is also further exploration related to other senses, such as augmenting the interaction with haptic feedback.

4 major AR interfaces: Tangible, Collaborative, Hybrid and Multimodal AR interfaces

AR has been recognized as a promising technology. But how to allow users to interact with the virtual content in an intuitive way is still a challenging question. Four types of existing AR interfaces are summarized in the following [Carmigniani et al., 2011]. *Tangible AR interfaces* allow users to use physical objects to interact with the linked digital media. TaPuMa by [Mistry et al., 2008] and SLAP by [Weiss et al., 2009] are examples on a projected tabletop surface. *Collaborative AR interfaces* allow multiple users to interact with multiple AR displays and the collaboration activities are supported. [Schmalstieg et al., 2000] proposed a concept to bridge multiple users, displays, applications and context with AR. *Hybrid AR interfaces* combine different interaction devices to achieve the goal in a complementary way. Flexible infrastructures of such systems allows the reconfiguration of input and output devices. An example was presented by [Sandor et al., 2005]. In the end, the emerging *Multimodal AR interfaces* use speech, gestures or other natural behavior as commands for the interaction. SixSense by [Mistry et al., 2009] envisioned a future world augmented with digital media, where digital content is controlled by gestural commands.

This work focuses on the visual AR with a Hand-held display

With the rapid improvement of capability, hand-held devices are increasingly becoming capable of running AR applications with adequate performance. Recent smartphones and tablet PCs have started playing important roles in assisting people's everyday activities. In this work I use a hand-held device to display the visual AR content, and I focus on investigating this approach. The existing Hand-held AR applications will be introduced in the next chapter (Section 2.2).

1.2 AR for Operations

AR assist operations for professionals, what about casual users?

Existing AR techniques have been used to assist workers in complex tasks such as assembly([Tang et al., 2003]) or maintenance([Henderson and Feiner, 2009]) by overlaying localized instructions onto physical objects with an HMD. In addition to professional domains, *In-place* guidances are needed for operations in everyday activities. Casual users often encounter problems while using appliances such as: "How do I set this washing machine for this cloth?", "What was the setting of my guitar amplifier when I played this song last week?" While the use of an HMD is impractical for daily activities, hand-held devices can be used for Mobile AR applications and provide in-place guidance to anyone anywhere.

AR helps to retrieve objects

AR techniques combine the advantages of text and pictures for operation tasks. By easing the localization of physical objects by displaying the instruction next to the object. This should help users by reducing both task switching and alternating attention.

Drawbacks of AR instructions

However, AR has its inherent problems. First, the AR layer and the user's hand often occlude the physical objects during the interaction. Second, the performance can be offset when the alignment of physical and digital objects is not perfect. Furthermore, the problems are even more severe with Hand-held AR, due to the small display size and relatively low resolution. In addition, a hand-held device often occupies one hand of the user, making it very inconvenient for operational tasks.

1.3 Add Real-time Feedback

Feedback is important for a system

"Offer Informative Feedback" is one of the "Golden Rules of Interface Design" from [Shneiderman, 1986]. It has been inherited and applied to handheld mobile devices as a design guideline by [Gong and Tarasewich, 2004]. As [Wensveen et al., 2004] pointed out for person-product interaction, the product should provide the information that guides the users' actions towards the intended function.

Feedback needs to be enhanced in AR systems

While AR instructions normally provide feedforward by indicating what to do for the upcoming operation, few systems add real-time feedback on AR overlay. Usually, users should get haptic and visual feedback from the physical device during the operation. Additionally, the physical device sometimes provides extra feedback such as light or sound, or display values on a display somewhere. However, the feedback on physical device is often hard to perceive in AR environment due to the occlusion. And even if it is visible, switching attention between the physical and digital might influence the performance.

Propose to add feedback on AR overlay

In order to take full advantage of mobile AR as instructions, I propose to provide real-time feedback for the status change of the physical device, e.g. the current value a manipulated knob, directly on the AR overlay. In this way, the AR instructions and the updating status of physical objects are both displayed on top of the camera input. The the occlusion problem might be less disturbing. And it may encourage users to interact while looking through the device screen, thus minimizing their attention alternation.

1.4 Research Contributions

This work aims to explore the possibilities and challenges of using AR to assist operation tasks in everyday life.

Explored a concept and evaluated the benefit of a proposed solution

First, I proposed and explored the concept about how Hand-held AR could assist people in daily operation tasks. I will show the scenarios and prototypes of example applications. Second, I prototyped the interaction with AR instruction adding real-time feedback on the AR overlay. In order to assess the advantages of adding feedback for AR instruction and obtain a deeper insight of the usability issues, I conducted a controlled experiment to evaluate the user performance with four types of instructions:*Text*, *Picture*, *AR* and*AR+Feedback*.

Experiment answers research questions

The experiment was designed to answer the following questions:

- Do Hand-held AR instructions assist operation tasks better than text and picture instructions?
- What are the benefits of adding real-time feedback to AR in this context?

1.5 Chapter Overview

Chapter 1 The first chapter introduces briefly about what is AR, how AR has been used for operations, and what the problems are. Given the problems, adding real-time feedback is proposed as a solution, which leads to the research contribution of this work.

Chapter 2 The related work first presents the existing AR systems that support operating physical interfaces. I then give an overview of the application domains of Hand-held AR. I also explain the close relationship between AR and tangible user interfaces. Finally, I discuss the opportunities of this work.

Chapter 3 This chapter illustrates how AR technique could be used in everyday activities to assist operations. It presents the problem scenarios, and illustrates the concept with examples. One application is developed through a DIA cycle with paper prototype and software prototype. And another application is demonstrated with paper prototype to show possible extensions of the interaction. I introduce the prototypes in detail and discuss issues that arose.

Chapter 4 A software prototype is built to simulate the interaction of AR+Feedback. This chapter explains how the prototype is implemented, and discusses the technical approaches for providing real-time feedback in AR systems.

Chapter 5 A controlled experiment is conducted to assess the benefits of adding feedback to AR and discover the usability issues. This chapter introduces the experiment design, hypotheses, procedure, measurement and implementation.

Chapter 6 This chapter explains the analysis of data collected from the experiment. It involves the methods and procedures of performed statistical analysis, and the validation of hypotheses from the results. In the end the discussion of observations provides more insights about the techniques.

Chapter 7 The last chapter summarizes the work and contributions, proposes future directions and possible extensions.

Chapter 2

Related work

AR has been introduced as a promising technique to assist operational tasks, and its benefits have been evaluated in different contexts. Generally speaking, most of these AR systems focus on specialized tasks with expert or specialized users. And they often require users to wear an HMD.

AR systems for operations are mainly for professional users wearing an HMD

As the goal is to use AR to assist operational tasks in everyday activities, and HMD is not socially acceptable in most circumstances, hand-held displays can be an alternative choice. There are fewer Hand-held AR systems dedicated to assisting operations in the existing work.

Hand-held AR is more socially acceptable, but few systems support operations

Before going from laboratories to the industry, AR still faces many challenges and issues regarding to its usability and acceptance. There is a large shortage of formal user studies to address these issues for this technology.

AR is lack of user study

In this chapter, I will first present the existing work that assists operational tasks with AR technology. Second, I will introduce the existing application domains of Hand-held AR, to give an overview about how it has been used. Third, I will draw a link between AR and Tangible User Interface (TUI) to indicate the vision of this research. In the end I will summarize the chapter and point out the opportunities of this work.

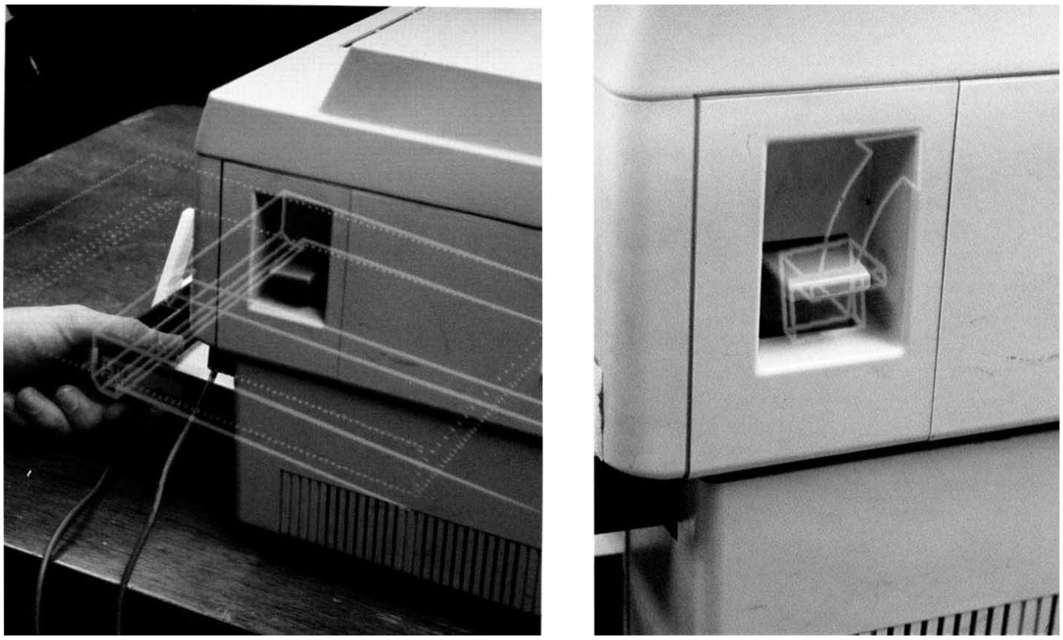


Figure 2.1: KARMA system. (Left) shows location of the paper tray. (Right) shows action of pulling up lid's lever.

2.1 AR for Operation Tasks

Existing AR systems supporting operations will be presented

In the last thirty years, researchers have used AR to assist operational tasks in many professional domains. In this section the existing work of such AR systems will be presented. While most of them are deployed with an HMD, only two of them use hand-held devices as the display. The last example is the only work I find that implements the approach of adding real-time feedback on AR overlay, which is applied in medical domain.

2.1.1 KARMA

[Feiner et al., 1993] demonstrate their early concept with the *Knowledge-based Augmented Reality for Maintenance Assistance* (KARMA) system. They distinguish AR from *Virtual Reality* (VR) as to present a virtual world that enriches, rather than replaces the real world.

They suggest that AR is a suitable technique to provide explanations of, and assistance with, complex 3D tasks. They built a prototype as testbed application. It illustrates simple instructions for using a laser printer via an HMD, such as refilling the paper tray or replacing the toner cartridge.

HMD-based AR instruction for a printer

It draws the abstract objects according to its visibility in the user's view. For instance it draws the cartridge in dash line when it is occluded by other objects, and it draws an object in solid lines to highlight it. As we can see in Figure 2.1, the object is highlighted and a "move" arrow is shown to indicate an action. While the user performs an action, it shows both the action and the designated position of the operated object in different illustration style. For example when the lid starts to move up, the user is guided to open it fully by seeing the expected position of the lid in dash lines.

Illustration in different styles to explain actions and help localize objects

The system is based on IBIS (Intent-Based Illustration System), a rule-based system that illustrates the scene according to an input communicative intent. They used Logitech 3D position and orientation-tracking system to determine the position of user's view point and the physical objects.

System architecture facilitates AR

The concept of this work inspired a lot of following work in this domain. The rule-based illustration architecture separated the graphical design and rendering to prevent the interference between design decisions and interactive rendering.

Inspiring concept and good system architecture

2.1.2 Maintenance

When it comes to supporting maintenance with AR, a significant work comes from [Henderson and Feiner, 2009], who presented a quantitative user study with professional users under field conditions. (See Figure 2.2) They showed how AR enhances localizations, so that it reduces the time and effort of sequential maintenance tasks in complex systems.

AR helps localization

The application uses a tracked HMD to display AR content combined with a mechanic's natural view. The AR content includes *arrows* for directing users' attention, *text instruc-*

AR instruction displayed with HMD

Figure 2.2: (Left) A mechanic wearing a tracked HMD performs a maintenance task. (Right) AR condition in the experiment shows localized labels and 3D models as instruction.

tions for describing the operation, *registered labels* for indicating locations and *3D models* of tools and turret components for visualization. For complex or ambiguous tasks, it shows animated 3D models in the user's view. Animations are controlled by a wireless controller worn on the user's wrist. Additional buttons and slider are provided for users to navigate animations.

Experiment compares AR to LCD and HUD

They compared their prototype application with two base line techniques. The LCD condition featured static 3D scenes displayed on an LCD monitor placed beside the workspace. The HUD (head-up display) condition was to control the general effects of wearing an HMD. It used screen-fixed graphics providing the same contents as AR condition but without registration to real objects. The within subject experiment repeatedly measured 18 maintenance tasks with different operation locations. They measured the overall task completion time, task localization time, error rate and users' head movement.

AR performs significantly better in task localization and reducing head movement

The result shows significant improvement of AR than both baselines regarding to task localization time. And AR requires significantly less head movement than LCD condition. Nevertheless the overall task completion time and error rate does not reveal significant difference of AR comparing to other conditions. In the post-experiment questionnaire, more users ranked AR as the most intuitive technique, but more users selected LCD as preferred technique.

This is the first formal controlled experiment I have found that evaluates the benefits of AR in operational tasks. Their conclusion about AR improving task localization and reducing head movement is the base of my hypotheses. However my work is very different from it because of the different AR display, the unprofessional users and tasks, and the added real-time feedback on AR overlay.

Experiment shows the benefits of AR in a professional domain

2.1.3 Object Assembly

[Tang et al., 2003] tested the effectiveness of an HMD-based AR system in object assembly tasks.

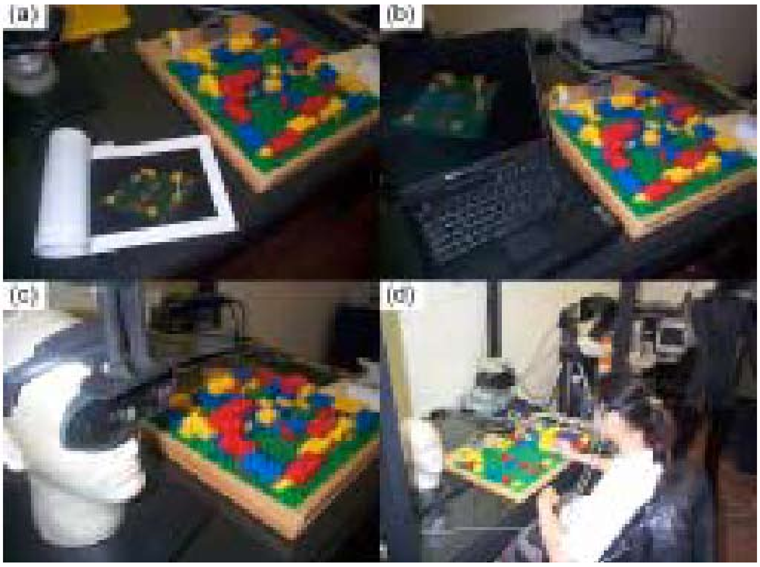


Figure 2.3: Experiment with Duplo assembly: (a) Printed manual, (b) CAI with LCD, (c) CAI on HMD and AR on HMD, (d) Experiment in action.

They conducted a between-subject experiment to compare four types of instructions: *a printed manual*, *Computer Assisted Instruction (CAI) on a Liquid Crystal Display (LCD) monitor*, *CAI on a see-through HMD*, and *spatially registered AR with HMD*. (See Figure 2.3) They measured the task completion time and error rate to evaluate the task performance, and measured the perceived mental workload with NASA Task Load Index (NASA TLX) after the experiment.

AR outperforms the 3 baseline conditions

Error rate is significantly reduced in AR, task completion time is not

They found that AR instructions reduce error rate and the user's cognitive effort. The assembly task has 56 procedural steps and some of them are correlated. The result of ANOVA analysis shows that AR has a significant improvement in both total error rate and dependent error rate comparing to the other treatments. In task completion time, only printed manual has significantly poor performance due to occupying one hand. This is avoided for the other three conditions since they use voice commands to control the instruction. No significant difference was found among the other three treatments regarding to time performance.

simple information overlay does not help, identifying the true benefits of AR

Some deeper insights are revealed from the unexpected poor result of *CAI on a see-through HMD*. It shows that simply overlaying information in the central area of the user's view does not facilitate the performance. This fact supports an explanation that the outperformance of AR is due to the assistance in mental transformation and the minimizing of attention switch. Although the tasks and setups are different from my work, this valuable finding also grounds my hypotheses.

2.1.4 Assembling with Hand-held AR

Hand-held AR for operations received less attention

With the dramatic development of the computing power, recent smartphones and tablet PCs become the most popular platforms for AR applications. However, very few of them assist operational tasks.

Instructions are animations on mobile phone adjusting to users' perspective

[Hakkarainen et al., 2008] describes a mobile AR system for assembly tasks. It displays simple animations on the mobile phone screen according to the user's changing perspective while viewing the work space. (See Figure 2.4)

Client takes images of the site, Server creates animations and sends to client

This system has a server-client structure. Markers are placed in the real assembly site for tracking the view angle. The mobile phone takes images of the site from the user's perspective and sends them to the server. The server creates a task guidance animation with a series of images according to the calculated perspective, and sends it back to the client to display. The user uses a keypad to navigate step by step in the task.

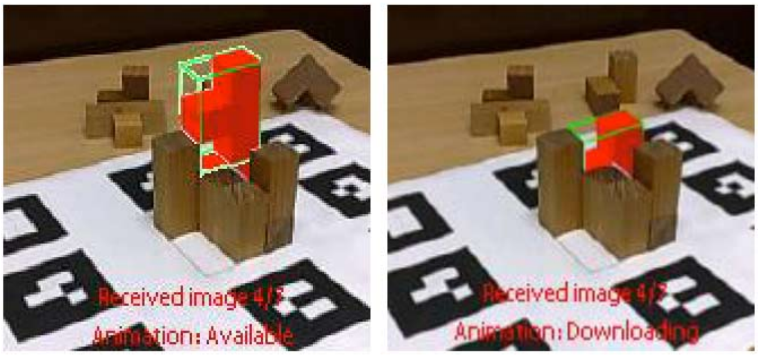


Figure 2.4: Augmented assembly of a 3D puzzle on a mobile phone.

In a subjective evaluation, participants were asked to construct a wooden 3D cube puzzle and answer a questionnaire with Likert scales. The results are positive, but the study is too informal to validate any effect.

Positive but subjective user assessment

This is the only work I found that provides AR instructions for operation tasks with a hand-held display. However the AR effect is limited to a perspective adjustment for animations, not with overlaying content on real-time camera images. And no formal study is presented.

Limited AR effect and lack of user study

2.1.5 Learning Complex Machines

AAM (Augmented Anesthesia Machine) presented by [Quarles et al., 2008] is a system that assists students to learn the mechanism of a complex machine. Knowledges are presented with a tablet PC. The tablet's position and orientation as well as the status of the machine components are tracked with vision-based technology.

AR system with Tablet PC for learning a complex machine

There are two ways to visualize the knowledge: concrete and abstract. *Concrete visualization* takes full advantages of a MR technique and displays spatially registered content. It displays animations of the component behind the tablet as if it is see-through. The user can see the effect of his interaction while turning the knob (Fig. 2.5 (Left)). Conversely, the *Abstract visualization* is showing 2D graphs il-

Concrete Visualization displays in-place animations of one component, Abstract Visualization displays dynamic 2D graphs of the whole machine

Figure 2.5: (Left) Concrete visualization is shown on a tracked tablet. A user turns a gas knob on real machine and AAM system plays animation to visualize the effect of user interaction. (Middle) A user interacts with the real machine while viewing the Abstract visualization with untracked tablet. (Right) Dynamic Abstract visualization with effect on the components.

lustrating abstract knowledges (Fig. 2.5 (Right)) about the machine. The tablet is not tracked in this case. The content is dynamic, it shows the corresponding effects on the graph while the user interacts with the controls (Fig. 2.5 (Middle and Right)).

User study compares Abstract-only visualization with Combined visualization

A between-subject user study was conducted to evaluate if MR's merging of real and virtual spaces can effectively help users understand the knowledge. They tested two conditions: *Abstract-only* and *Combined* (concrete + abstract) visualization. Two groups of users used different visualization techniques to perform exercises, and then completed a hands-on machine fault test and a written test about the machine's mechanism. The overall time for completing exercises was also recorded. The result shows that the combined visualization is more effective in teaching concrete concept and it helps to bridge abstract and concrete knowledge.

Shows the benefits of specially registered instructions

Although this work focuses on the benefits of MR as an educational tool, it actually illustrates how spatially registered instructions help users operating complex machines from a learning perspective.

2.1.6 Medical Examination With Feedback

MR system to help learning breast examination skills

Adding real-time feedback on the AR overlay could be ex-

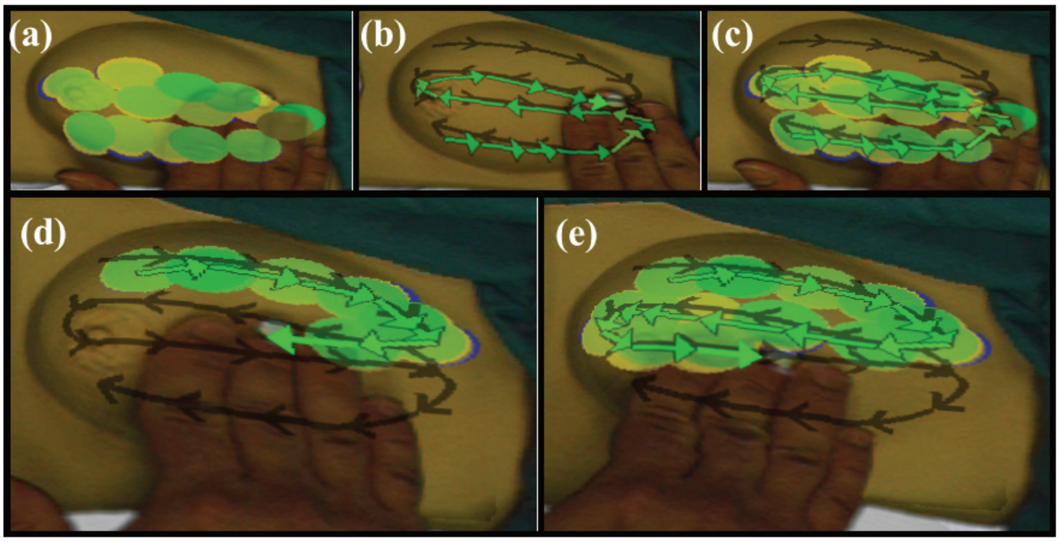


Figure 2.6: Breast Examination with MR system: (a) Touch map visualizes different pressure levels with colors. (b) Pattern-of-search map shows the path of hand operation. (c) Combined visualization of touch and search pattern. (d,e) Progression of the combined visualization.

tremely helpful when the physical feedback is hard to capture or measure by users. An MR medical learning system is proposed by [Kotranza et al., 2009] to help students learn breast examination skills.

Breast examination tasks require concurrent use of psychomotor and cognitive skills. The medical students have to perform hand operation with a certain pattern and a proper force on the model at the same time. With this MR system, the user is provided real-time generated and in-situ presented visual feedback of her performance in the tasks.

Realtime feedback of users' performance is provided on AR overlay

The user wears an HMD as the display. The instruction with visual feedback is provided as an AR overlay on top of a patient model and the user's hand. As we can see in Figure 2.6, the force levels of finger press are represented with different colors of the trace. It also shows the path of correct finger movement with filled color as feedback. A combined visualization could guide the fingers to be on the correct path with proper pressure. The AR content is drawn with partial transparency to ease the occlusion to the user's hand. Pressure sensors are placed beneath the

Provide real-time and in-situ feedback about pressure and operation pattern physical breast model to capture the user's hand motion.

Positive but subjective user evaluation

An informal qualitative study was conduced with expert educators and students. A lot of quotes from users are presented as the result. It indicates the system is very valuable in assisting learning.

This is the only found AR guidance system that provides real-time feedback on AR overlay. The effectiveness of the real-time AR feedback has not been evaluated in statistical approaches.

2.2 Hand-held AR Overview

My work focuses on investigating the interaction techniques with hand-held displays as Magic lenses. Therefore I introduce the application domains of Hand-held AR with examples, to give an overview about how this technology has been used.

2.2.1 Augment the Environment

Browse the world knowledge

With GPS, compass and Wifi network, location-based AR can help people in outdoor navigation. The Touring Machine [Feiner et al., 1997] is an early example for that. Wikitude[1] combines GPS and compass data with Wikipedia entries. It allows users to browse the knowledge about their surroundings in real-time camera images on their smartphone while they are physically "scanning" the world with it. Layar[2] is a commercial application that implements the same concept and integrates more services such as Twitter and local transportation services.

Indoor navigation

There are also AR applications for indoor navigation. [Wagner and Schmalstieg, 2003] introduced an AR navigation system running on a Personal Digital Assistant (PDA), which guides a user to a chosen location in an unknown

---

[1] `http://www.wikitude.com/`

[2] `http://www.layar.com/`

Figure 2.7: Hand-held AR museum guide.

Figure 2.8: SiteLens: visualize geocoded carbon monoxide data in AR scene for urban designer.

building by showing arrows beside the corridor and highlighting the doors with outlines. As another example, METAIO [Miyashita et al., 2008] is presented as a museum guide providing information of the art work as well as for navigation (see Figure 2.7).

Visualize data in the world for city designer

[White and Feiner, 2009] present SiteLens (Figure 2.8), which supports urban designers and planners by visualizing the relevant virtual data directly in the physical site. It displays 3D virtual color balls beside the road in the real-time camera image to visualize the geocoded carbon monoxide data sensed by the sensor.

AR provides localized information to save the effort of searching, thus helps people access the knowledge in the world in an effective way.

2.2.2 Augment the Paper

Many scientists are interested in adding digital powder to physical paper. After The MagicBook [Billinghurst et al., 2001], many projects have explored the applications of this concept in different areas. [Grasset et al., 2008] summarized them and explored the design space and user experience.

Augmented notebooks

[Mackay et al., 2002] present prototypes of an augmented laboratory notebook with a tablet PC and PDA. The tablet captures writing on the paper and the PDA carries the interaction with the electronic documents. ButterflyNet [Yeh et al., 2006] links multimedia with handwritten notes to create augmented paper notebook. It associates handwritten notes to the multimedia content, then allows an organized multimedia notebook browsing on a computer (see Figure 2.9).

Augmented maps

There is also some work that augments the paper map. [Rohs et al., 2007] evaluate three interaction techniques for map navigation with joystick, static peephole and magic lens approaches (Figure 2.10). The result shows that the magic lens and peephole navigation outperform joystick thanks to the reduced switching of attention between the surface and background. [Morrison et al., 2009] conducted

Figure 2.9: ButterflyNet: augment the notebook by associating the picture to specific position on the paper.

a comparative study between AR map and 2D mobile map in a location-based game, and discovered the potential of AR maps to be a collaborative tool.

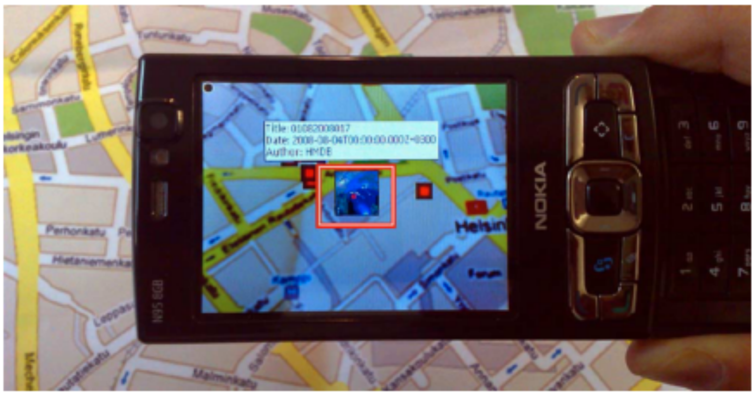


Figure 2.10: Augmented paper map of Magic Lens approach.

Paper is easy to grab and annotate, while digital devices can display multimedia content and get information from other resources. By linking the physical and digital world, AR provides a way to utilize the advantages of both mediums.

2.2.3 Entertainment

Various AR games have been developed

AR enhances gaming by providing entertaining virtual experience mixed with the reality. Here are some selected examples. Mosquito Hunt is the first commercial AR cam-

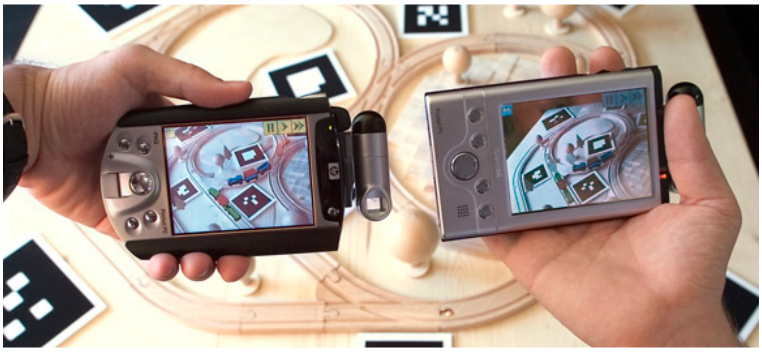


Figure 2.11: Invisible Train: an AR collaborative game.

era game running on mobile phone. The Invisible Train[3] implemented on PDA is the first multi-user AR application on handheld devices. Players control virtual trains on a real wooden miniature railroad track. AR Tennis [Henrysson et al., 2005] is a face to face collaborative AR game on mobile phones. Two players play tennis game together with a piece of paper in between, which is tracked as the play ground. AR Soccer [Geiger et al., 2004] shows a virtual goal on camera image of the mobile phone screen. The user views his own foot in the soccer environment and tries to shoot the virtual ball into the goal.

AR advertisement becomes popular

More and more AR advertisement is appearing on handheld platform. The first AR commercial was developed for the Wellington Zoo[4]. It displays a 3D animal with a mobile phone by "scanning" a marker printed in newspaper or a poster.

Gaming is the first application domain of AR that is entering people's everyday lives. Mobile phones are the channels to enable this.

---

[3] `http://studierstube.icg.tugraz.at/invisible_train/`

[4] `http://theinspirationroom.com/daily/2007/augmented-reality-at-wellington-zoo/`

2.2.4 Reflection

AR techniques have been used to preserve and share memories recorded as digital artifacts associated to physical objects.

Share memories during knitting by attaching the record on the scarf

With Spyn [Rosner and Ryokai, 2008], users can record the events or emotional moments happened during a knitting process as multimedia files (video, image), and link them to physical locations of a knitted scarf with time and location information attached (Figure 2.12). The scarf can be sent as a gift, and the recipient can using a mobile phone to access and display the record.

Figure 2.12: Spyn: associate digital artifacts with positions on a scarf to share memories of a knitting process.

Physical objects are shortcuts for digital record

As another example, The Memory Box [Frohlich and Murphy, 2000] allows a user to place a physical object in a box that is associated to an audio source. [Mugellini et al., 2007] present their work using personal objects for memory recollection and sharing. The physical objects are linked to photos or videos and act as a tangible interface for showing the digital contents.

Physical objects are linked to recorded memories with AR. Then the objects act as shortcuts for people to access the memories.

2.3 AR and TUI

Close relationship between AR and TUI

Tangible User Interface (TUI) is another interaction domain that also bridges the physical and digital worlds. It has a close relationship with AR, from both technical and interaction aspects. I explain this with two selected examples in the following.

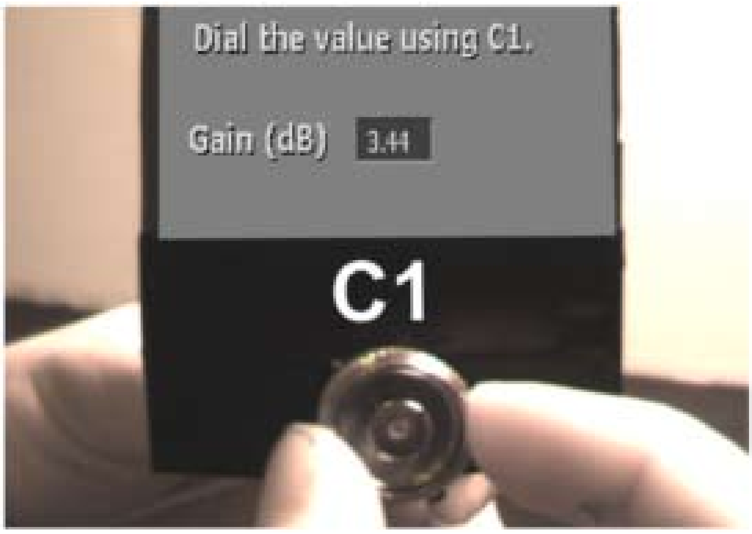


Figure 2.13: Opportunistic Control: the number in digital layer changes while users turn the physical control.

One control is a TUI for 3D content, and is augmented by the 3D content

[Henderson and Feiner, 2008] present Opportunistic Controls that combine 3D imaginary and physical controls (Figure 2.13). An opportunistic control is a tangible user interface of the 3D content, where it provides tactile feedback and physical affordance. At the same time, it is a physical control augmented by overlaid information to provide visual feedback of its updating value. Therefore it is described as a "compatible surface in the physical task domain of the AR application."

Physical widgets are input controls for virtual objects, and are augmented by virtual objects

SLAP [Weiss et al., 2009] provides several types of widgets that can be linked to any virtual objects (movies, images) and act as input controls for them (see Figure 2.14). The widgets include knobs, sliders, keypads and buttons. For example, linking a knob to an image will then allow the user to adjust the brightness of the image by turning the

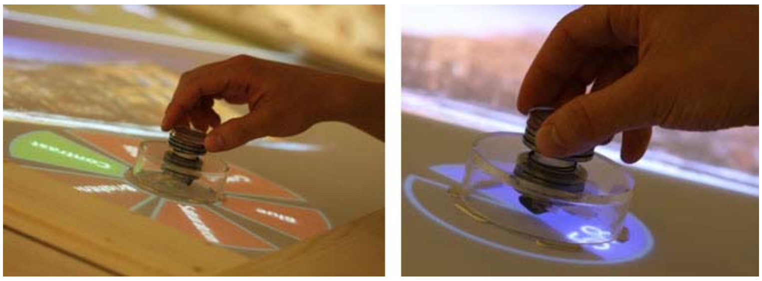


Figure 2.14: SLAP: link a knob to a virtual object as input control, (Left) to select a property, (Right) to set a specific value.

knob. The widgets are tangible interfaces for controlling the virtual objects. From another perspective, the physical widgets are augmented by being linked to the digital content.

Approaches of TUI can be utilized by AR

The techniques in TUI can be utilized in AR to enable novel interactions, such as adding real-time feedback to AR. Technically there are a lot of approaches in TUI to sense the status of physical objects. For instance, SLAP tracks the identifiable “footprints” created by reflective markers made of foam and paper to detect the status change of the physical widgets. This leads a possible way for AR to go.

2.4 Summary

In consideration of the research presented in this chapter, my work will extend this from the following aspects.

First, explore the possibilities of Hand-held AR in assisting normal users for everyday operations.

AR for operations are limited in professional use

Existing AR systems support operation tasks in very limited application domains. They are mainly built to help experts with professional tasks, such as maintenance of complex machines, medical operations or complex assembly. Usability and acceptance issues keep this technology from supporting people in everyday activities.

Hand-held goes into everyday life, but not applied for operations yet

Mobile devices are capable to bring AR to everyday life, and are more socially acceptable than current HMDs. However, the existing Hand-held AR systems spread in application domains such as navigation, gaming, learning and so on. Assisting operation tasks with Hand-held AR has received less attention, and has not been thoroughly explored.

Second, conduct formal experiments to evaluate the benefits of adding real-time feedback on AR overlay, and get a deeper insight about the benefits and problems of Mobile AR in assisting operations.

Shortage of user study for Mobile AR

Generally speaking, Mobile AR has a lack of user study for its effectiveness and usability. For supporting operations, most of the user studies focus on very specialized tasks with expert users. Adding real-time feedback on AR overlay appears to be promising in a medical application. But only informal and subjective evaluation is presented. Apart from this, I am not aware of any other work applying this approach or studying about its effect.

Third, envision a possible way to improve AR with the prototyping approach as an example.

AR techniques have been struggling for years with the unsatisfying performance due to the technical limitations. And it is even more critical with Hand-held AR due to the thinner computing platforms.

Blur the boundary of reality and virtuality

As explained in Section 2.3, physical objects act as tangible controls or interfaces of virtual objects, while they are actually augmented by the linked virtual objects at the same time. Referring to the continuum of Mixed Reality in Introduction chapter (Figure 1.1), the boundary of reality and virtuality is becoming more and more blurred.

Utilize approaches of Ubicomp Computing to improve AR

Ubiquitous Computing ([Weiser, 1991]) was mentioned as a complementary approach of AR in early work ([Feiner et al., 1993]). In my work, the approach to mock up the interaction of adding feed-back to AR is actually an example that embodies a vision: the approaches of Ubiquitous Computing can be utilized to complement the weakness of AR to engage novel interactions, and vice versa.

Chapter 3

Use AR to Support Everyday Operations

Use personal data to generate AR instructions

The objective is to use AR techniques to assist operational tasks in everyday activities or casual use of complex machines. Some everyday operations are not hard to perform, but cannot be accomplished due to missing information. For example, operating a keypad without a passcode, or setting a washing machine when one has forgotten what temperature to choose. To achieve the goal, I propose an AR system that utilizes users' personal data to generate AR instructions.

Mobile devices are easy to access and used to record data

Casual use of complex machines often happens in unexpected situations. Mobile devices are commonly carried with users everywhere. They are inherently personal for users. People record personal data with them by taking notes or pictures. These make good opportunities for Hand-held AR to fulfill the goal.

In this chapter I will first introduce the problems with scenarios, then I will demonstrate the concept of providing personalized AR instructions, and then introduce two example applications with prototyping processes.

3.1 Problem Scenarios

An *Operation* here means performing an action on an object to achieve a certain goal. In our daily lives, people often have problems with operations. I will introduce them with examples from several aspects.

3.1.1 Daily Operations

Daily operation with knobs, buttons etc.

People use tools or machines on a daily base. Common interfaces include controls such as buttons, sliders, switches, knobs, joysticks and so on. To utilize these tools, people have to learn how to operate the interfaces. Examples of daily operations could be setting an oven for a certain dish, configuring a certain program for a washing machine, finding a function for a photo camera, making cappuccino with a coffee machine and so on.

Operations need to be assisted, especially for special users or situations

Operations often need to be assisted when the interfaces of tools do not have clear affordances. Special user groups need help for using some devices. For example, senior citizens sometimes have difficulty using devices such as digital cameras because of lack of technical knowledge. Extra guidance is also needed for some particular situations. For instance, people may have difficulty using a washing machine if the instructions are not written in their spoken languages.

Example: Keypads

People enter a door by inputting passcode with a keypad

In some cities like Paris or Stockholm, people use keypads to unlock the doors. Figure 3.1 shows a set of keypads that one could meet everyday in different locations in Paris. People are commonly stuck in front of such a keypad, because they do not have the code. Most people use a piece of paper or personal devices to note the code. But notes often get lost or become hard to retrieve.

Different keys and layouts of keypads make the operation time-consuming

Since the keypads are made by different providers, they

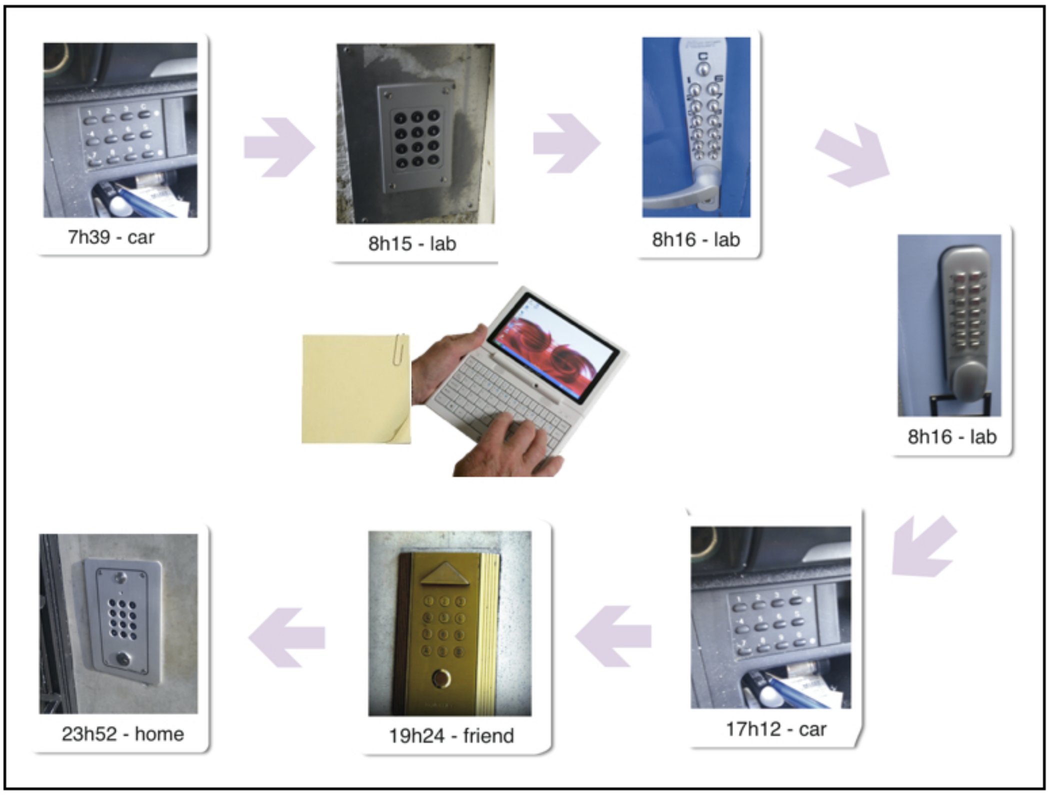


Figure 3.1: One day with keypads in Paris.

often have different layouts and even different letters and digits on the keys. This makes the code hard to remember. Moreover, the digits and letters need to be interpreted and mapped to the key locations. This increases the time required for entering the code. In addition, due to the limited space of humans' working memory, people may also have to memorize the digits in trunks and then input them. This method is error-prone, thus becomes time-consuming.

3.1.2 Casual Users for Complex Interfaces

Casual users need straightforward guidances for operations

Casual users are not as familiar with a tool because they only use it occasionally. People sometimes run into a situation that they have to use a machine that is alien to them. For example using a copy or fax machine with a lot of functions and confusing interfaces, or cutting a piece of paper with a laser cutter when one has no knowledge about it. They need very straightforward guidance for the operations.

Example: Audio Station

Figure 3.2: Audio station in a lecture room.

Presenter often have problems with the audio stations

Figure 3.2 is an audio station in a normal lecture room. It happens very often that a presenter could not use it properly because they are not familiar with it, or more commonly they do not have enough time to configure it. When problems occur, a common scene is that a technical staff shows up and solves the problem by simply pressing one more button or correcting a wrong wire connection. This should have been much easier if the presenter had direct instructions for his configuration.

Normal instructions cover all the situations, thus provide redundant information

A common solution is to put instructions beside the device. The instructions are often not easy to understand quickly since they cover answers for all the frequent questions. They provide redundant information that needs to be understood and matched to varied situations. It is sometimes frustrating when a user needs a quick answer directly for - “how to operate to get what I want”. There are cases that people need to complete one task as soon as possible, without having to understand the mechanism.

3.1.3 Multiple Configurations

Multiple settings for one device

Some facilities should be configured differently in various situations. Audio effect devices or control stations have specific pre-settings for every song or each environment. Washing machines could also be an example, because clothes require different settings depending on the materials and colors.

Example: Notes for Effects Pedals

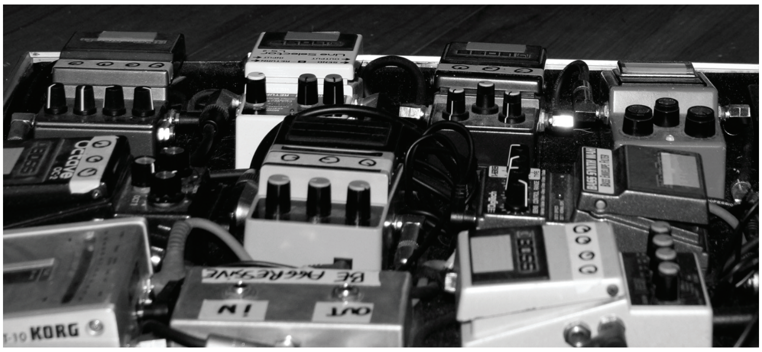


Figure 3.3: Effect Pedals with knob controls and notes beside the knobs.

Effect pedals with notes of settings beside

Figure 3.3 shows some effects pedals with multiple knob controls. They are used for adjusting audio effects of electric guitars. Users stick paper notes beside the knob controls to remind themselves of specific configurations.

3.1.4 Hand Work

Some operation tasks are with many physical objects or components, such as assembly, DIY (Do It Yourself) tasks, or cooking with many spices and mixing cocktails. Those operations all involve a searching phase for the right object, and a phase about what to do with it.

Example: Assembly of Furniture

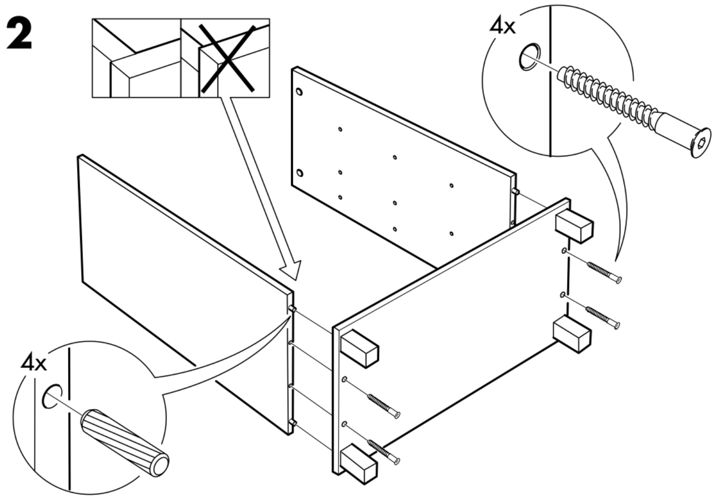


Figure 3.4: Instruction of furniture assembly.

Figure 3.4 is an example of common instruction about how to assemble a furniture. As we can see, it takes much time and effort to understand how to operate exactly.

Operations are difficult to describe

Operations are especially difficult to describe and understand with the traditional instructions such as text or pictures. Because there is much information needed, including the locations and descriptions of objects to manipulate and how to manipulate. In addition, static instructions are not suitable to describe dynamic operations as they require further mental interpretation.

3.2 Personalized AR instructions

AR reduces attention switch

As discussed in the Introduction chapter, common instructions use traditional mediums such as text and pictures.

The main disadvantage is that they force users to switch attention between the guidance and the physical interfaces, there are also other problems when it comes to everyday lives. AR instructions might reduce the attention alternating.

People take notes about operations

People often take notes about the operations when they are guided to do it the first time, for example the casual uses of a complex machine. Common personal instructions are written on papers, or stored in digital devices such as PDAs and mobile phones. And they mainly appear to be text, sketches or pictures taken with the camera.

Problems with existing personal instructions

Paper notes are easy to lose, and not always accessible when they are needed. Information stored in mobile devices is easier to access. However, when there are many notes stored without a organized structure, searching for particular ones also takes much time and effort. Furthermore, when the operation space is big, people have to take multiple pictures for one setting. It is very inconvenient to view them due to the discontinuity. Content is also easily to be missed during the recording. In addition, bad picture quality might also cause the missing of information or affect the accuracy.

Personalized AR instruction as one solution

To solve the problems, I propose to provide personalized AR instructions on hand-held devices as one solution. Users should be able to easily provide their personal data for creating AR instructions for the operation tasks.

Note Recording and Displaying as interaction metaphor

The interaction is designed with *Note Recording and Displaying* as a metaphor. One *AR Note* represents an instruction of one or a series of operations for a physical interface. It could be a note for one setting or one operational task. One AR Note could have multiple layers for a sequential operation task. It should display the in-place AR instruction generated according to the user provided data.

I investigated how such systems could be used in different situations, and how they could be designed. Due to the variety of operation tasks, it is very hard to find a generic interaction for all of them to visualize the instruction and input personal data. However, it is valuable to have different approaches for different applications.

3.2.1 Examples

Here are the proof-of-concept illustrations of two example applications.

Figure 3.5: Concept illustration of Keypad Assistant Application.

Operating a keypad while seeing through the AR overlay

Figure 3.5 shows how a keypad assistant application could appear. A user “scans” a keypad with a camera phone. Once the keypad is recognized, an AR instruction layer is overlaid on top of a physical keypad through the phone screen. Ideally, the mobile phone should be able to recognize individual keys, and the instructions could be automatically generated from the code. The green dots represent the recognized key positions, while the pink arrows are the animated tapping footprints between the keys, which indicate the operation sequence for unlocking the door. Users just need to provide the code in text to generate or change the instruction.

Viewing AR Notes for a guitar amplifier through a mobile phone

Figure 3.6 shows how AR notes can be used to augment a guitar amplifier. The picture on the top shows a guitar amplifier. Each setting is stored as one AR note. In (a, b, c) the green and red graphical components are the AR instructions overlaid on top of the knobs through the mobile

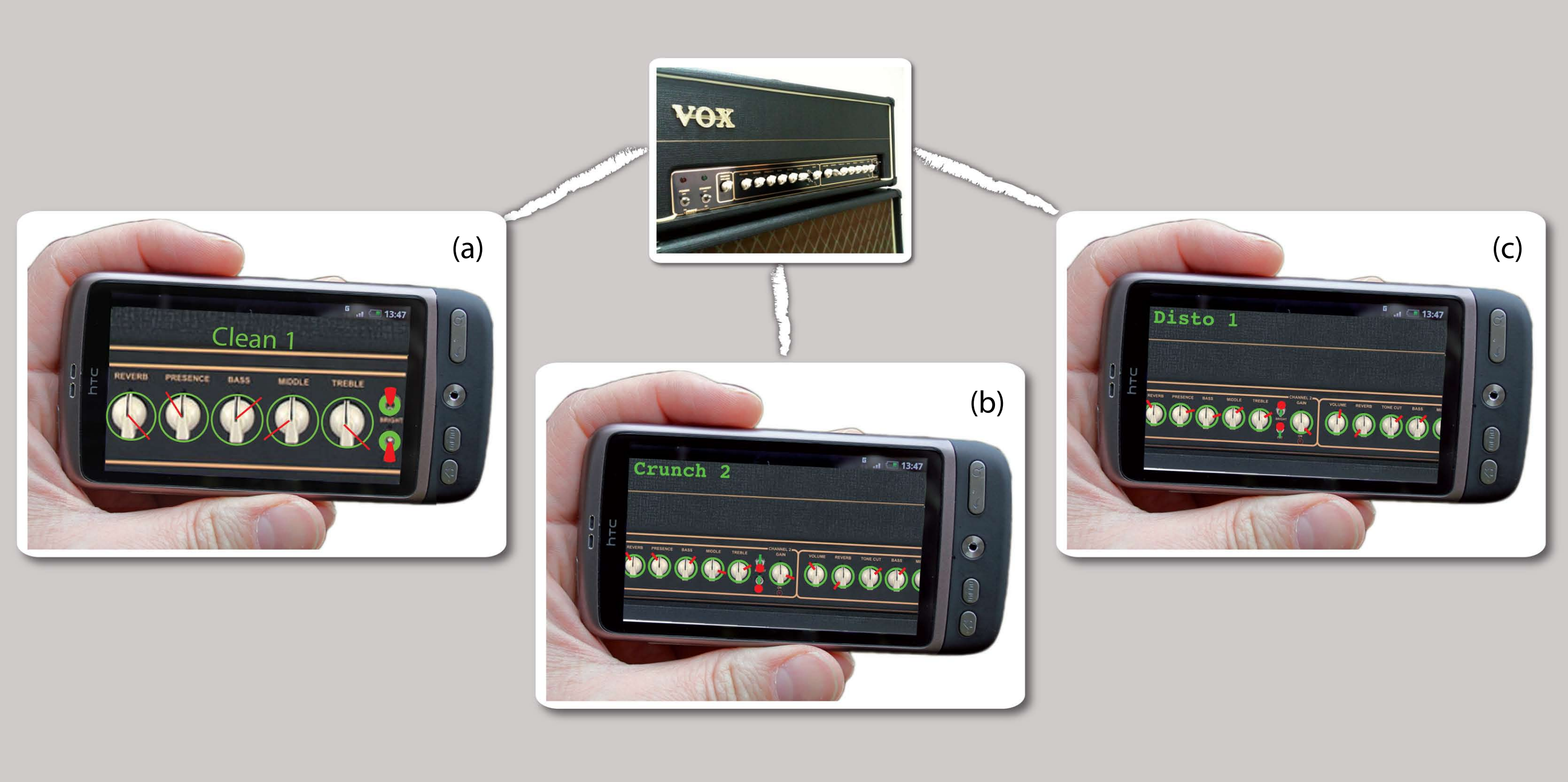


Figure 3.6: Display AR Notes as overlays on a guitar amplifier for multiple settings with different zoom levels.

phone screen. The green circles highlight recognized knobs and are drawn as outlines. The red bars indicate the values of the controls in the corresponding setting. While (b, c) show the overviews of two different instructions, (a) shows a closer view of a part of one instruction. A closer view presents more precise values or possibly more information. The zooming should be realized by physically moving the phone further or closer to the physical surface.

I also sketched a storyboard to explain an scenario for this application, see Appendix C.

3.2.2 Discussion

The illustrated two examples represent two situations for using AR Note.

Dynamic instruction shows the pattern of hand operation

In the Keypad Application, dynamic visualization provides natural and continuous guidance for such sequential operations. This application should be able to support the code entries by providing the patterns of hand operations, especially with varied keyboard layouts.

Static instruction provides continuous viewing experience

In the Guitar Amplifier Application, AR notes present multiple settings for the surface, regardless of the operation order. This AR instruction provides a continuous experience while zooming and moving the view among different controls. So it should be easier for the user to locate certain controls to manipulate.

Prototypes will be presented in the following

I created prototypes for two example applications - Keypad Application and Control Panel Application. For Keypad Application, I will present the paper prototype, software prototype and informal user evaluation. Control Panel Application is designed for any control surface. The paper prototype includes the design of interaction for users to author the instruction. They will be introduced in the following sections.

3.3 Keypad Application

DIA development process

Design - Implementation - Analysis (DIA) cycle is a typical process for developing an application with users' participation. It is used for developing the prototypes of Keypad Application.

I started with a low fidelity prototype and a small subjective user study. With the users' feedback, I developed the software prototype with some technical assumptions. In the end, an informal user evaluation resulted in positive assessment about the usefulness and rose some issues regarding to the usability and performance.

3.3.1 Low Fidelity Prototype

Fast prototype on an iPod

I created a low fidelity prototype on an iPod for the keypad application with a fast prototyping software - Realizer[1]. This software allows me to define arbitrary areas on a screenshot and link the areas to other screenshots. So in this prototype, the colored areas react to tap events and then it jumps to the linked screenshots.

Paper prototype explanation

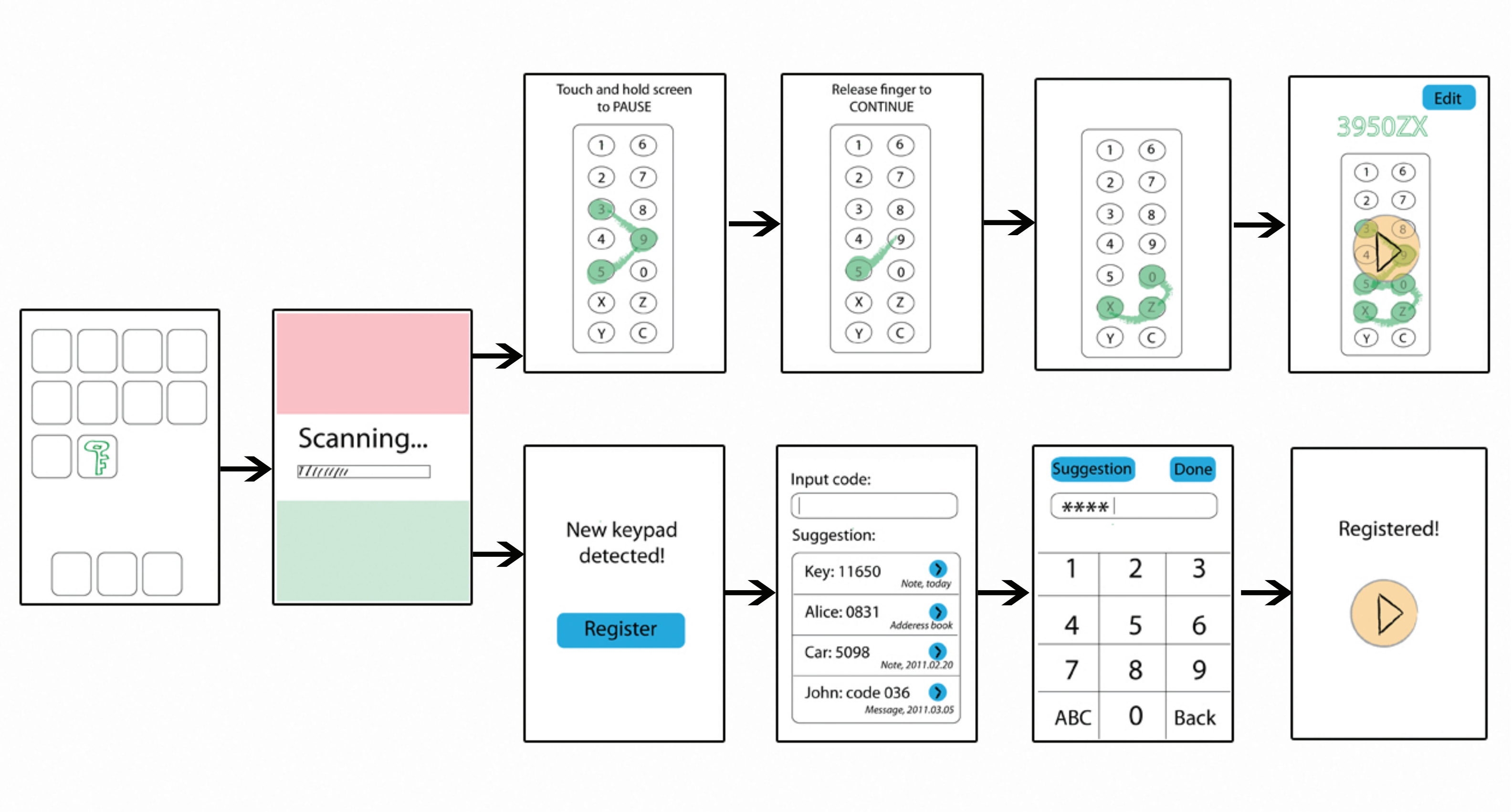
Figure 3.7 shows the designed interaction flow with the screenshots for this prototype. The arrows indicate the possible interaction sequence. The first screenshot is for opening the keypad application. Then after "scanning" the keypad with the iPod, two possible interactions would happen depending on if there is already a code registered to the keypad.

Interaction shown in upper row of screenshots

If there is a code, the interaction goes to the upper row of following screenshots. An animation is illustrated as a dot jumping between the keys and leave traces to indicate the path. Holding the screen with one finger should pause the animation, while releasing the finger should continue the animation play. In the end of the animation, it should show the digits and the complete trace. And tapping the screen should play the animation again.

---

[1] `http://realizerapp.com/`



Scanning...

Touch and hold screen to PAUSE

Release finger to CONTINUE

3950ZX

New keypad detected!

Input code:

Suggestion:

Key: 11650 Note, today
Alice: 0831 Adderess book
Car: 5098 Note, 2011.02.20
John: code 036 Message, 2011.03.05

****

Registered!

Figure 3.7: Screenshots of a low fidelity prototype for Keypad Application on an iPod.

If there is no code registered to the keypad, the interaction goes to the lower row of the following screenshots. If the user is willing to register a new code, it should ask for user to input the code. As shown in the second screenshot in the lower row, there are automatic suggestions based on context information, such as GPS location and contact list or personal note, to save input effort. When the user starts the manual input, the standard keyboard appears. This input interaction can also be invoked by tapping the "Edit" button shown after playing the animation, as drawn in the last screenshot of the upper row.

Interaction shown in lower row of screenshots

User Feedback

The prototype was demonstrated and explained to two colleagues who were not informed about this work. Their feedback is stated in the following.

- The application could be useful for solving their problems with keypads.
- The static traces of the operation should not disappear during the animation play. It could be better if they stay there after being drawn. Otherwise the user might miss the beginning of the animation.
- Although this prototype is already interactive by responding to tap events, it is still hard to imagine the real effect when this turns to be an real AR application. Because the screenshots cannot show dynamic instruction, and this does not have a see-through effect.

Could be useful

Prefer complete trace in the animation

AR effect is hard to imagine from this prototype

According to the feedback, the interface design is slightly changed. Furthermore, a higher fidelity prototype appears to be especially important for AR applications.

3.3.2 Software Prototype

In order to get user feedback with real AR effect and test the technical feasibility, I built a software prototype for keypad application. Some subjective user feedback was collected.

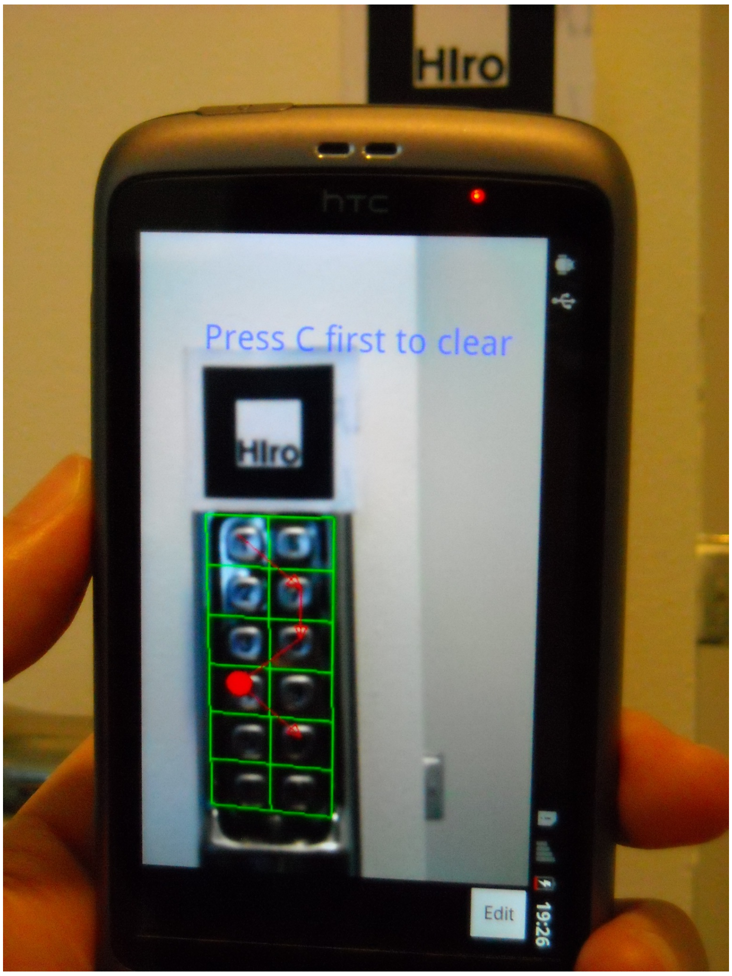


Figure 3.8: Demonstration of Software Prototype for Keypad Application.

AR Instruction with grid, arrows and text

Figure 3.8 is a picture of the software prototype for the keypad application. The mobile phone screen displays real-time camera image with the AR overlay, which includes the green grid, red arrows and blue text. The graphical instruction appears similar to the first prototype. The blue text provides extra explanation about the operation. In this case it tells that the user needs to first press “C” to clear the previous entry before entering the code.

The green grid outlines the recognized keys and the red arrows indicate the passcode. There are three types of visualization that users can choose.

3 types of visualization

- *Still* - static arrows show the full path;
- *Animated* - arrows appear one by one after a time interval and leave traces, the red dot jumps between keys at the same time to indicate the key presses;
- *Manual* - step by step controlled by swiping left or right on the screen, each step draws one arrow and move the red dot.

While a user moves the camera phone, the graphical instruction moves and the perspective transforms depending on the camera position and viewing direction. However, the blue text only changes its size and position but does not have perspective transformation. This is a design decision, because deformed texts could be hard to read.

Perspective transformation for graphical instruction, not text

If the user clicks on the "edit" button in the bottom-right corner, he leaves the AR display and navigates to an editing view. He could then enter the code and the text instruction via standard keyboard widget. There are also radio buttons for selecting one of the three visualization types of the instruction.

Use edits the code and text instruction

Due to the technical limitations and fast prototyping, a technical assumption remains in this prototype. It is assumed that the keys are recognized by the smartphone and the grid layout is generated automatically. It is mocked up by using a fiducial marker (the "HIRO" pattern in the figure) placed at a fixed position relative to the keypad. The phone tracks the fiducial marker and applies the perspective transformation on the AR graphical layer.

Mock up with fiducial markers

The application is implemented on an HTC smartphone running Android[2] system. Graphical components are drawn with AR effect, while the text is drawn separately in a 2D `View`[3] with `Canvas`[4] according to a position relative to the fiducial marker. As the implementation for AR

Implementation aspects

---

[2] `http://www.android.com`

[3] `http://developer.android.com/reference/android/view/View.html`

[4] `http://developer.android.com/reference/android/graphics/Canvas.html`

layer is basically the same for all applications in this work, more technical details could be found in next chapter (Section 4.3.3).

3.3.3 User Evaluation

Subjective evaluation by 2 users

Two users were recruited to test the application. They are male researchers, use keypads in a daily base, and they were not involved in the previous phases of this project.

Positive assessment

Their feedback shows the application is helpful since it displays the operation patterns instead of text. The performance is not perfect but adequate to assist the task.

Usability Concerns

Two major concerns were stated by the users.

Concern about the incomplete view for larger interface

"What if I hold the phone too close and only a part of the keypad is shown?" This is probably not a big issue for keypad application because users would adjust this. However for more complex machines or control panels, this can be solved by indicating a moving direction of the phone to force users to move to the designated position. However, this could present more technical challenges.

Animation has to be followed

It is sometimes not easy to follow the animation while operating. The user needs to adjust the speed of operation to synchronize with the animation speed.

Inspired the idea of adding feedback to AR overlay

"Wouldn't it be cool if it shows the next key entry after I enter the previous key?" It would allow users to control the animation by physically tapping the keys on the keypad. In the meantime, showing the next key entry is actually a feedback of a correct key entry for previous step. This inspired the idea of providing feedback of users' operations on AR overlay. It will be introduced in the next chapter.

Performance Issues

Three performance issues were pointed out by the users.

Image deformation

One user mentioned there is deformation of the camera overview after rotating the phone between vertical and horizontal directions. This was due to the inconsistency of camera resolution and screen size. The problem was fixed later by changing the size of the AR display.

Degraded image resolution

One user noticed the decrease of camera image resolution while running the AR application. This is due to some technical constraints. It does not influence the usability for this application, because the precision needed to distinguish the keys is not high. However this might be a problem for some other applications.

Lag in updating and unstable graphs

The positions of AR components are not updated in perfect frequency, so there is slight lag while the user moves the phone. In addition, the drawn lines are not very stable due to the updating perspective transformation. However, these are minor issues that do not have much influence for this application.

3.4 Control Panel Application

Paper prototype for configuring a control surface

To extend and generalize the concept of AR Note, I designed and prototyped another example application. It supports the browsing and authoring of AR Notes for any control surface with common controls such as buttons, knobs or sliders. The purpose is to explore this concept and give an example of how the interaction could be.

No user test for this prototype

I only present an untested prototype here, because this work focus on the more concerned issues regarding to the usability and acceptance of Mobile AR in general. They will be elaborated in later chapters. The evaluation of this prototype will be considered in future work.

3.4.1 Paper Prototype

The paper prototype is an application for viewing and recording settings for a control surface. It includes two parts: *AR Note Browsing* (Fig. 3.9) and *AR Note Authoring* (Fig. 3.10).

One AR Note for one setting, consists of the colorful widgets

In this application, one AR Note presents one setting for the whole control surface. The prototype consists of graphs of a mobile phone screen that shows real-time camera image. Colorful widgets belong to the AR overlay and the grey shapes are the physical controls appear in the camera image. Since a paper prototype cannot be dynamic, the user needs to imagine the AR effect that the colorful widgets stick to the physical controls with perspective effect while the phone is moving.

Swiping to browse information for the same controls in different AR notes

Figure 3.9 is about AR Note Browsing. When the physical device is recognized, the existing AR Notes are listed on the screen of the smartphone. If the user select one, it then shows the green AR components aligned to the physical controls. For sliders, the long bars highlight the controls to be configured and the small circles on them indicate the designated value positions. If a user swipes on the screen, it switches to the next or previous note (Fig. 3.9 Graph 2 to 3). The view is not changing as the AR widgets are constantly mapped and aligned to the real objects, so the user could easily view the different values of one control from one setting to another.

Zoom in to see more information by moving the phone closer

Continuous zooming could be realized by moving the AR display closer or further. For a more complex interface, the application could be designed in a way that it shows more information when it gets closer. As shown in the last graph in figure 3.9, it displays the value range, setting value and function of the control in text in a very close view.

Associating AR Notes with physical objects should be able to ease the search, visualization, browse and organization of the instructions.

To provide personalized AR instruction, we need to allow users to author the AR content. Figure 3.10 illustrates the

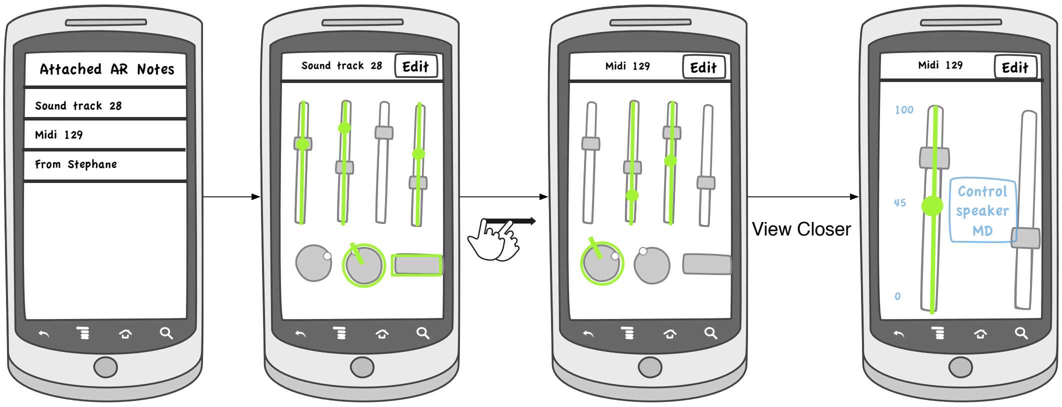


Figure 3.9: Paper prototype for browsing AR Notes on a control station.

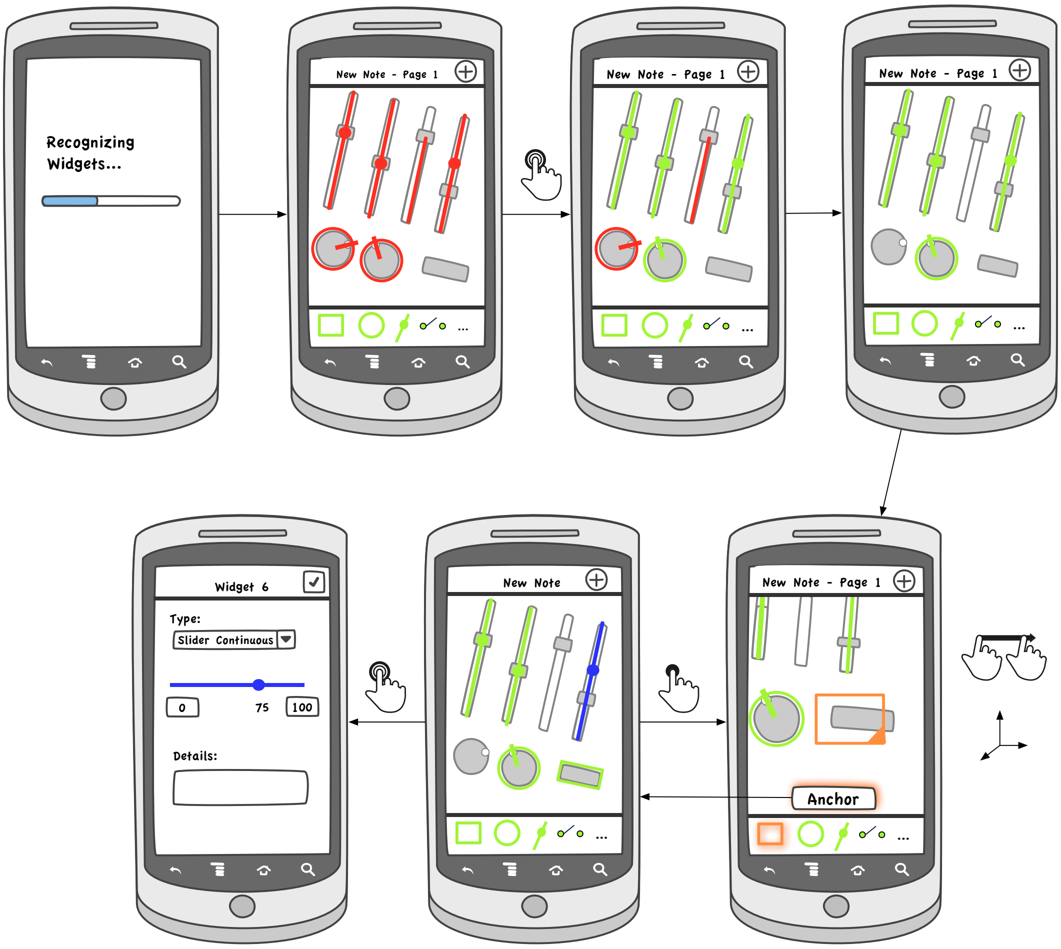


Figure 3.10: Paper prototype for authoring AR Notes for a control station.

designed interaction for authoring an AR note. It should be able to allow users to annotate controls and input information about them.

Users can select the recognized widgets to include in a note

The physical controls and their status should be recognized automatically. The red outlines in the second graph show the recognized controls and their values. As you can see, the recognition is not perfect as there is a missing button and a wrong recognition of one slider. A double tap on one widget could turn it from red to green. Green widgets are included in a note. The fourth graph in Figure 3.10 shows the view after discarding two recognized controls for the current note.

To put a widget into the scene, the user resizes the scene by moving the phone instead of resizing the new widget

Users should also be able to add widgets to the scene. As demonstrated in the fifth graph of Figure 3.10, the user could drag one widget from the lower tool bar, which contains widgets for common controls. The orange square is the new widget to add into the 3D scene. At this moment, it stays as a static 2D square fixed on the screen, while other green widgets are interactive in the AR scene with 3D effect while the phone is moving. The ratio of the square can be changed by dragging the little rectangle in the corner of the square. To make the widget fit and align to the physical control, the user rotates the phone physically and moves it further and closer to zoom the scene, instead of pinching the screen to zoom the square. This approach could be an alternative to using common hand gestures to reshape a widget, and might be more natural and convenient for user input on such a small display.

Press "Anchor" to release the widgets into 3D scene

After getting a fit of the widget and physical control, the user could tap the "Anchor" button to release the square from the 2D screen into the 3D AR scene. The widget is then registered to a certain location on this control surface to represent the aligned control. As an inverse way, if the user want to redefine the location of an existing widget, a long press on this widget would "pick" it up from the 3D scene and get it back to a position on the 2D screen, just like before tapping the "Anchor" button.

Users can provide more information about one control in edit view

Double tapping a green widget in the 3D scene will navigate to an editing view, where the user can change the widget type, the setting value and the value range, as well as details in text. The detail information will be shown when the user browse the AR Note in a close view, as shown in the last graph in figure 3.9.

3.4.2 Discussion

Support customization of widgets

Ideally for this application, all controls and their states would be automatically recognized perfectly. This can be realized by advanced Computer Vision technology. However, it is not realistic yet to assume it always works perfectly. Even if the recognition can be perfect, users might not want to annotate all the controls. Therefore the customization of the widgets should be supported. The widget selection and refinement in the first phase of note authoring is provided as such a complementary approach.

AR Authoring is another topic, not the focus of this work

There is a lot of existing work about general AR authoring. They aim to provide a general way for users to sketch and annotate the 3D environment from a 2D screen. For instance, [Guven et al., 2006] present a technique that allows users to annotate the environment through a tracked tablet PC by freezing the scene and later editing with multimedia and hypermedia data. [Zauner et al., 2003] provide an easy way for authors to create MR based assembly instructions for furniture. [Broll et al., 2005] present an infrastructure to ease the customization of AR interfaces. [Tenmoku et al., 2005] propose a view management method to facilitate the AR annotation using 3D models of the real scene. Nevertheless, the focus of this work is not to provide AR authoring solutions in a technical aspect, but to investigate and evaluate the usability and acceptance of Mobile AR in a certain context. Therefore, there is no further exploration on this topic in this work.

Different authoring approaches for different physical interfaces could be another solution

In addition, instead of a generic authoring interface, there could also be different ways to enable AR authoring for different physical interfaces. For instance, the keypad application, users could simply input the code through keyboard on the mobile phone. For more complex devices or machines, the manufacturer of the assisted machine could provide a graphical template for all the controls. This would also enable a finer interface to provide information for specific controls and operations. If the instruction is sequential with multiple steps, multiple layers could be provided for one AR Note.

Chapter 4

Adding Feedback to AR

Many possible applications for Hand-held AR assisting operations were introduced in the last chapter. Nevertheless, when it comes to operational tasks, using a see-through mobile device for guidance has possible disadvantages due to the hand occupation and other issues.

Hand-held AR for operations has drawbacks

Conventional AR provides only feedforward of an action, which demonstrates what the user should do. As mentioned in the Introduction chapter, it is essential for a system to provide informative feedback. As an example, [Bau and Mackay, 2008] presents a dynamic guide for performing gesture commands. It shows the benefits of providing concurrent feedforward and feedback.

Feedforward + Feedback is essential

While AR augments the real world with overlaid information, it also interrupts the communication between the users and the real world. The visual feedback of the physical world cannot be effectively perceived by users.

AR layer occludes some feedback from real world

Therefore, I propose that an AR system could get the status of the real world in responding to the user's operations, and illustrate the meaningful information directly on the AR overlay.

Adding feedback as a solution

To investigate and evaluate the proposed approach, I built a high fidelity prototype to simulate this interaction technique. This prototype is used as one condition in a con-

Build prototype to simulate the interaction trolled experiment for the evaluation, which will be introduced in next chapter.

Chapter summary

In this chapter, I start with a discussion about the challenges of Hand-held AR for operations and will cover how these problems may be eased by adding real-time feedback. Then I will demonstrate the prototype application and will explain some of the implementation details. I will conclude by discussing the technical approaches for a system to detect the status of the real world in order to provide the feedback.

4.1 Problems of Hand-held AR for Operations

Some problems of Hand-held AR were raised during the the prototyping phase.

Small display

First, the small screen size of hand-held devices limits the user's view. Users physically move the phone further and closer to zoom out and in. When the physical surface is large, it is hard to see the overview without sacrificing the accuracy or detail information.

Occlusion

Second, occlusion has been a central problem in AR, one that can be even more severe during operation tasks. Two areas of concern are the user's hand obscuring real objects in the workspace and the AR content obscuring operation feedback. This means that the information flow between the user and the physical world is interrupted, which may cause precision to be degraded.

Uncomfortableness and unnaturalness

Third, depending on the camera position and how the user holds the phone, the view point of the camera often has a large offset to the view point of the user. In this situation if a user looks at his own fingers through the camera view, it feels a bit lost while moving the finger towards the object. It may cause people's feelings of uncomfortableness and unnaturalness towards AR to increase. This effect was noticed from the user feedback in previous prototyping processes.

Fourth, although recent smartphones have adequate capabilities to run AR applications, there are still some limitations. The technical bottleneck is to track the physical objects stably and align the digital and physical objects in the scene while the user changes the viewing distance and perspective. The failure of tracking, lag of the movement and shaking artifacts can degrade the usability.

Technical performance

Finally, one hand of the user is usually occupied by the Hand-held display. So this technique does not apply to hands-free operations, which causes inconveniences when users have other concurrent tasks to do.

Occupy one hand

4.2 Benefits of Adding Feedback

The direct manipulation of physical objects is impaired in AR systems, because the real world is seen through a mobile lens that is partially covered by virtual content. Some machines have feedback in the form of sound, light or other media, while others have none. When visual feedback is important for an operation, the AR content is in the way. If the user has to move the mobile lens away every time he needs to check the status of a real object, the task performance may be degraded. Additionally, other displaying and viewing problems can make the situation worse. Therefore, feedback needs to be enhanced to reflect the status of real world.

AR display is weak at reflecting the real world status

Adding real-time feedback to AR means means having users get the feedback corresponding to their operations directly on the AR overlay. The feedback should visualize the meaningful information that users need to get for performing the operation. For example, providing a notification for when the user turns a control to the correct position.

Provide meaningful feedback on AR layer

There are several possible benefits for this approach.

- The enhanced feedback complements the weakness of AR displays in reflecting the real world's status.

  Reflect status of real objects

- Even if the virtual content is perfectly placed, switching attention between the AR layer and physical

  Avoid attention switch

world still requires some time and effort. If the feedback on the AR layer is informative enough for the operation, switching attention may be not necessary anymore.

Ease error handling

- It eases the error handling. While the user himself can discover errors in time with real-time feedback, the system will also compare the user operation with the instruction and indicate correctness directly. This saves the user from having to compute and interpret the data, thus lowering the task complexity.

Simplify the operation

- Direct feedforward and feedback about the operation will make an operation very simple. Users will not even have to understand the physical interface. This could be useful for special user groups (e.g. elderly people) and in situations where users only want to learn how to use the device to complete a simple task. For instance the casual users for complex machines (Sec. 3.1.2).

4.3 Mock-up Implementation

Bypassed the technical challenges

Providing real-time feedback in Mobile AR applications raises technical challenges. It requires state-of-the-art technologies in Ubiquitous Computing and Computer Vision, to establish the communication between the mobile device and real world objects. Since the primary objective is to evaluate and understand the benefits of this approach, the technical challenges are bypassed in this prototype.

Ad-hod prototype was built with faked tracking and detection of physical controls

An ad-hoc prototype is built based on a generic controller that communicates with a mobile device. It fakes the detection of the status of controls to provide feedback. The same as the Keypad application prototype, it fakes the recognition and tracking of the physical controls.

4.3.1 Interaction

AR view draws outlines and values overlaying the real controls

The user can operate the control surface while looking through a mobile phone. The AR instruction is displayed

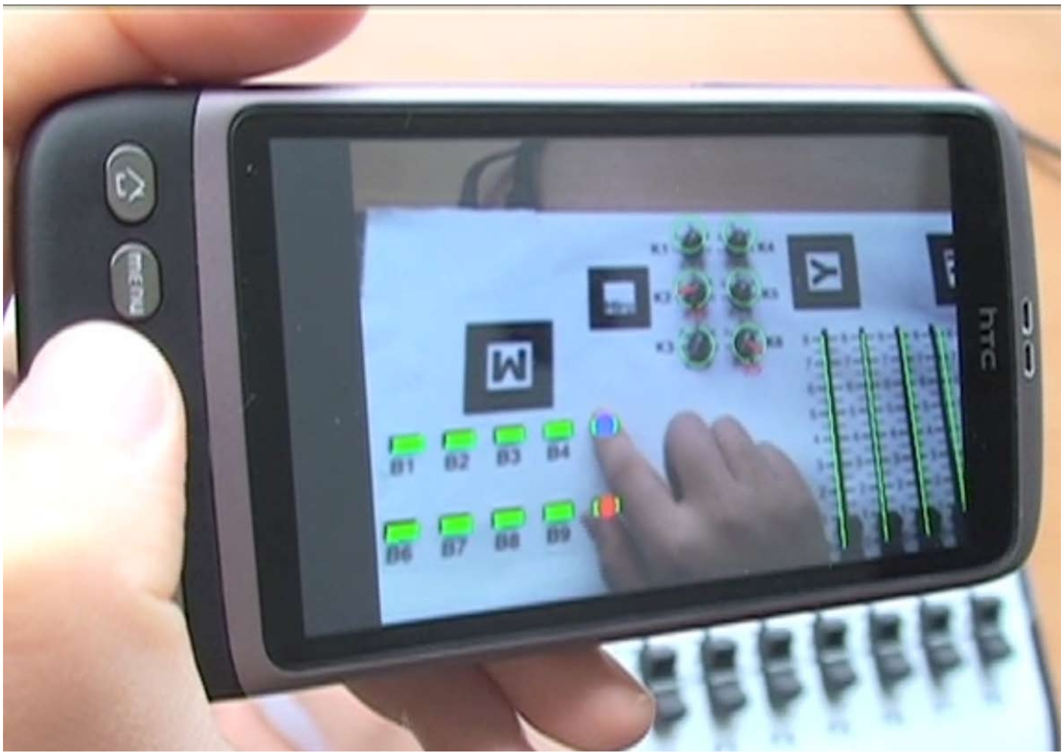


Figure 4.1: The instruction on a button turns blue after the user presses it. The color feedback on AR overlay indicates the effect of operations.

at a fixed position relative to the control surface, thus the widgets are aligned to the controls. As shown in figure 4.1, the AR instruction consists of the bright green, red and blue graphical components. It highlights the "recognized" physical controls with green outlines and draws setting values in red. Red components can be dynamically changed to blue according to the user's operation, which provides real-time visual feedback.

Dynamic color change provides feedback of operation correctness

The green squares are the outlines of the buttons. The ones with red circles are to be pressed according to the current setting. The figure shows that the red circle turns blue immediately after the user presses the button. Similarly for the knobs, the green circles outline the knobs and the red bars indicates the values for the controls to be manipulated. The precise values of the designated knobs are shown as numbers in red text below the knob outlines. If the user turns a designated knob to the correct value, the value bar and number both turn blue. If a wrong slider is manipulated, its outline turns purple.

The feedback saves some effort for moving the phone to zoom

Users can move the phone further away to see an overview of the settings, or move it closer to zoom in. To check the whole surface with a close up view, the users should physically move the phone. With the provided feedback, users can fulfill the operation task easily by observing the color change in the overview. They neither have to read the values in the digital layer, nor check the value of the real widgets closely.

Possible extension of visual effect

There are a lot of possibilities to improve the interaction for this application. For instance, a richer animated effect could be added when the correctness changes. It is also possible to provide various visualizations for the values, such as drawing 3D bars with different heights to display values instead of the plain 2D bars. In addition, instead of solely changing colors, sound feedback could be provided for the status change.

Simple design of this prototype

Nevertheless, the simple visualization is enough to demonstrate the concept, especially since this prototype is dedicated specifically to evaluating the added feedback in AR displays. To eliminate the influence of other factors and prevent the degrading of technical performance, the interface is kept simple. The experiment will be introduced in next chapter.

4.3.2 Setup

Values of physical controls are received by computer

Figure 4.2 illustrates the setup of this prototype. Operation tasks are performed on a JLCooper CS-10[2] MIDI control station with several types of controls, which represent the common controls that may be found on physical appliances (i.e, buttons, knobs, sliders). It is connected to a computer via USB connection, and the computer runs a software called WILD Input Server Pietriga et al. [2011]. It can be configured to communicate with the control station and receive updating values while the controls are physically moved. The software also provides an interface for processing and visualizing the control values.

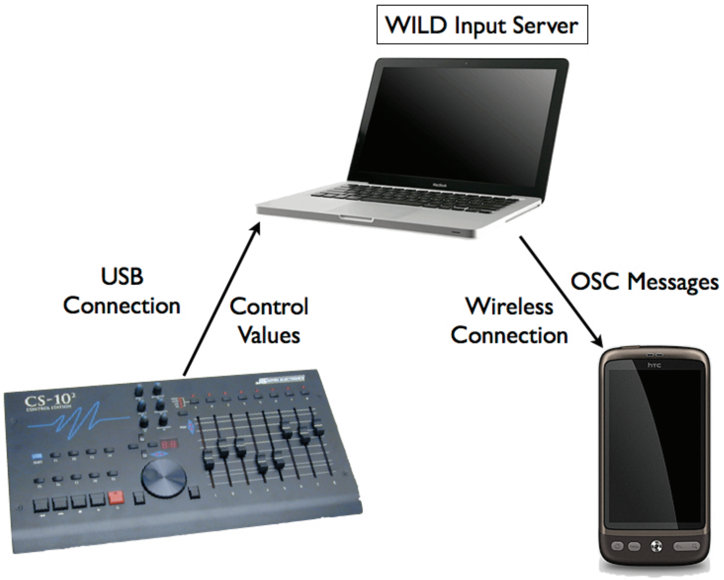


Figure 4.2: Setup for the mock-up implementation.

Server-client communication

The computer is a server that sends OSC messages[1] to the client mobile phone via a Wireless network. The messages carry the updating values of the controls while users manipulate them. The messages are defined with this protocol: `/mobileAR/<control_type>/<control_name> <control_value>`

Here are three example OSC messages following this protocol.

```
/mobileAR/fader/f4 0.30708662
/mobileAR/button/b9 true
/mobileAR/knob/k6 0.27559054
```

Reaching correct value leads to color change

The mobile phone gets the updating values of manipulated controls, and compares it to the correct control values that are predefined in the settings. When a control reaches a correct value, the graphical layer is informed to change the color for the corresponding widget.

Limitation

This setup relies on the instrumented control panel and a computer to communicate to the mobile phone. So it is not

---

[1] `http://opensoundcontrol.org/`

suitable for many other appliances or scenarios. However it is adequate to simulate a general configuration task environment, where users can operate an AR system without any professional knowledge or stills.

4.3.3 AR Display

Control surface is covered by white paper with markers for tracking

The control surface is covered by a piece of white paper to cover the unused controls for the experiment. As we can see in figure 4.1, the black squares with patterns in the augmented scene are fiducial markers. They are printed on the white paper, at fixed positions relative to the controls. Instead of tracking real physical controls, it is faked by tracking the markers for fast prototyping. The color white makes the best contrast to the black markers and helps facilitate the tracking.

Android phone tracks markers to display AR

The mobile application runs on an HTC Desire mobile phone with Android[2] 2.3.3 operating system. The mobile phone tracks the position of one or more markers, and displays an AR layer on top of the real-time camera image.

Tracking

The application utilizes a framework called NyARToolkit[3] for tracking fiducial markers. It is a port of ARToolkit[4] for Android. When a marker is recognized and tracked, the assigned graphical content is displayed at a fixed position relative to the marker. The perspective transformation will be explained in the following paragraphs.

Get transformation matrixes by processing camera image with markers

The application receives the camera image and processes the image for each frame. After the marker area is extracted from the image data as a transformed square, the framework calculates a transformation matrix from it. Then the matrix can be used to transform other graphical compo-

---

[2] `http://www.android.com/`

[3] `http://nyatla.jp/nyartoolkit/wiki/index.php`

[4] `http://www.hitl.washington.edu/artoolkit/`

nents into the same plane of the marker with a zooming effect.

A group of graphical widgets are drawn in a marker's coordinate system. This means that the center of the marker is the origin, and the graphical components are drawn with the coordinates relative to this origin. The transformation matrix explained above is applied to these graphical components to create the effect of perspective. In the end the dynamic AR effect is realized by frequently updating the transformation matrix while the camera position is changing.

Widgets are drawn in marker space and transformed to augmented scene

Train Markers

One marker is not enough for a large control surface. In this prototype, four markers are used for tracking and displaying instructions for the whole panel. One marker is used to track the ten buttons, another marker is used for the six knobs, and the remaining two markers are used to track the eight sliders.

4 markers for the whole surface

With the Marker's Online Generator[5], new markers can be designed and generated for ARToolkit in three steps.

The first step is to design new markers. It is important to make the markers as different as possible, because afterwards each marker image is only a pattern with 16 x 16 resolution. To make the tracking stable even when the phone is a bit far away from the panel, I chose capital letters "M", "Y" and "N" as the patterns for these three new markers. I also kept the original "Hiro" marker because it yielded good tracking results in practice.

Choose different patterns to train markers

The second step is to get raw data to train the markers. Begin by taking a photo with the mobile phone camera for a printed marker. The lighting condition when taking this photo should be as close as possible to the experiment lighting condition. It is also better to use the same camera that the AR application runs with.

Pay attention to the lighting setup and use the same camera to collect raw data

---

[5] `http://flash.tarotaro.org/blog/2009/07/12/mgo2/`

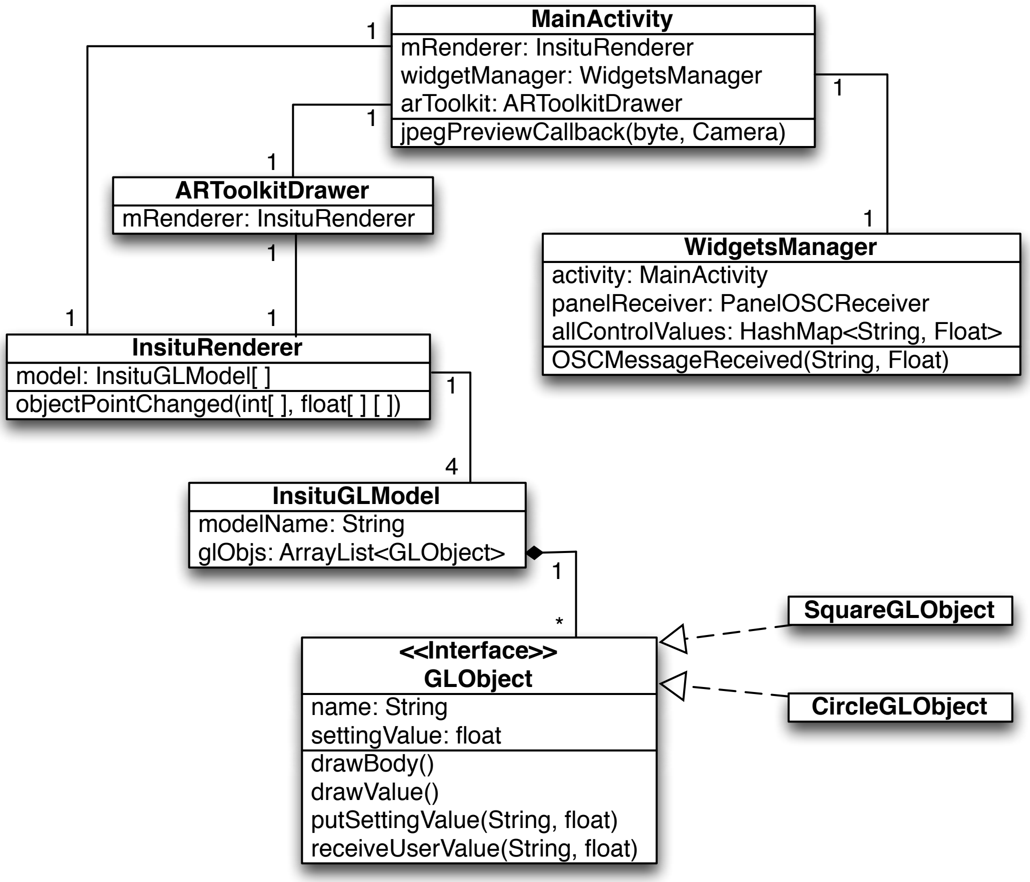


Figure 4.3: Class Diagram for the AR techniques.

The third step is to run the online generator. This generates one `*.pat` file for each marker. These files are placed in one of the source code folders of the application.

Graphical Widgets

The graphical widgets are drawn with OpenGL ES 1.0[6]. As explained in the tracking section, the coordinates of the graphical content are defined in the marker's coordinator system, where the marker is the origin.

Figure 4.3 is a class diagram that illustrates the relation-

---

[6] `http://www.khronos.org/opengles`

ships between the major classes that are related to the functions of AR drawing and providing feedback. Part of the code comes from NyARToolkit framework.

Calculate transformation matrix and apply to graphical model

`MainActivity` receives camera images through a callback function and sends them to `ARToolkitDrawer` for image processing. `ARToolkitDrawer` calculates the transformation matrix from the extracted marker information in the image. Whenever it detects a view point change, `ARToolkitDrawer` sends the new transformation matrix to `InsituRenderer` by calling its `objectPointChanged` method. `InsituRenderer` applies the transformation matrix to the graphical model, which consists of four objects of `InsituGLModel`.

Composition of the model

One `InsituGLModel` is the graphical content registered to one marker. It is mapped to one marker name, which is associated with a `*.pat` data file that was created in the marker training process. Each object of `InsituGLModel` is composed of several objects of `GLObject`. `GLObject` is an Interface[7], which is inherited to be either `SquareGLObject` or `CircleGLObject`. An object of `SquareGLObject` is a button or a slider, while an object of `CircleGLObject` is a knob.

WidgetsManager receives values of physical controls and notifies GLObjects

The `WidgetsManager` receives OSC messages of the updating values of the physical controls. It notifies the `MainActivity` about the changes, and then the `MainActivity` passes the information to the model down to the specific `GLObject`. The `receiveUserValue` method in `GLObject` is called for this. Correspondingly, its `putSettingValue` method is called to get the correct value for this control after a setting is loaded.

GLObject draws its outline and value separately, value color is updating

For each `GLObject`, its outline and value part is drawn separately in order to change colors easily. The green outline is drawn with `drawBody` method. Then the `drawValue` method draws the value part, which is represented as a bar and numeric value for a knob and slider, or as a circle for a button to indicate a press. The notified correct value from a setting and the current value of the

---

[7] `http://docs.oracle.com/javase/tutorial/java/IandI/createinterface.html`

physical control is compared to decide the correctness. If the value of the physical control is within the defined precision range, the application changes the color for drawing values from red to blue.

4.4 Technical Discussion

The technical aspect is not the focus, but worthy of discussion

To provide real-time feedback in AR, the AR device needs to be notified about the status changes of physical objects, which is a technically challenging task in most situations. This challenge is bypassed in the above prototype with a Server-Client setup. Although it is not the focus of this work to solve this challenge, it is interesting to discuss the possible approaches to establish communication between devices and physical objects.

RFID bridges physical objects and digital with electronic tags

RFID (Radio-frequency identification) is a technology that uses radiowave signals to transfer data from an electronic tag that is attached to an object. The objects with tags can be identified and tracked by a RFID reader. This technology has been proposed to bridge physical and virtual worlds since the early years. Novel applications was presented by [Want et al., 1999].

Marker tracking with square and non-square makers

Vision-based tracking is the most popular tracking approach in existing AR systems. Marker tracking appeared as the leading approach in earlier work. [Zhang et al., 2002] conducted a study to compare the performance of different kinds of square markers. Other researchers explored the tracking techniques with non-square visual markers. For instance the 2D bar-coded fiducial system proposed by [Naimark and Foxlin, 2002]. Marker tracking is easily interrupted since misrecognition happens when the marker is partially covered, a common scenario since the markers occupy some space of the scene.

Natural feature tracking is more advanced and runs on mobile devices

Instead of tracking fiducial markers, the more advanced approach is to track the naturally occurring features, such as the edges or textures. Natural feature tracking techniques are more advantageous than the marker tracking methods because of the marker-free tracking ability and stable performance. The detected features are often used to construct a model of the scene, which can improve tracking robustness and performance. These technologies are becoming lightweight so that they can be deployed on mobile platforms. For instance, [Reitmayr and Drummond, 2006] developed a robust model-based tracking system for outdoor AR with hand-held devices.

Hybrid tracking is promising

The vision-based tracking is not perfect due to the outliners and occlusions. Many other approaches can be combined to compensate for the weakness, thus improving overall performance. Outdoor AR systems often combine GPS with other tracking methods. This helps to eliminate some extreme outliners. [Foxlin et al., 2004] utilizes tilt sensors and a compass to correct the drift differential of inertial tracking. According to the properties of specific tasks and physical objects, there are many ways to combine the approaches.

Ubicomp computing provides many complementary approaches

As noted by [Feiner et al., 1993], Ubicomp computing is a complementary approach of AR in blending the border of digital and physical worlds, and enabling connections between physical objects. [Estrin et al., 2002] summarized the emerging approaches of instrumenting the physical world with sensor-rich and embedded computation. The properties of physical objects can be tracked with various types of sensors, such as pressure, acoustic or optical sensors. The status information of these objects can then be transferred to a receiver device. For instance, [Newman et al., 2001] used ultrasonic sensors for indoor tracking in a wide area.

Discussion of the approaches for Keypad Application as example

Going back to the example of Keypad Application (Sec. 3.3), how can the user operation on a keypad be detected by the camera phone? One possible approach is to install pressure sensors on the keypad to detect the key press. However, it is not feasible to expect all keypads to be equipped with pressure sensors. A more generalizable approach is to use vision-based tracking. Pattern Recognition technology can recognize arabic numbers and letters, so the keys can be automatically localized. The user's finger can be tracked assuming there is strong color contrast between the background and the finger. A topological approach can then be combined to help determine the pressed key. For example on a standard numeric keypad with nine digits, when "5" is pressed it is very likely that both “2” and “3” are occluded by a finger.

As the world is becoming “smarter” and everything is becoming more connected, we can foresee a future where the communication between digital devices and physical objects, as well as between objects, is easy and efficient.

Chapter 5

Experiment

AR techniques lack of user studies

Generally speaking, there is a shortage of user studies for AR techniques. Before an AR system could go from the laboratory to the industry, there are many questions to be answered concerning its usability and acceptance.

Questions to be answered

Would this technique make users' operations more efficient? Do users feel comfortable using it? Would they put forth the effort to get used to this technique, or prefer to use other more familiar tools? As explained previously, adding real-time feedback on AR overlay is proposed to overcome the challenges of Hand-held AR in assisting operational tasks. Would the added feedback effectively improve the task efficiency? Do users like it?

Existing work does not answer our questions

Although AR has been proven to be helpful for operations in the related work, the use of a mobile lens as the AR display instead of an HMD makes a huge difference regarding to usability. Professional users differ from normal or casual users with their trained skills. They make the effort to get used to the AR technique in order to improve the task efficiency. So it is a very different situation when it comes to casual or unprofessional users for normal operations.

Experiment goals

A controlled experiment was designed and conducted to answer our questions. The goals were to assess the benefits of adding real-time feedback on AR overlay in a certain context, and get more insight about the usability issues of

Hand-held AR instructions.

In this chapter, I will introduce the experiment design and procedure. This part of the work was conducted in collaboration with one of my advisors, Prof. Stéphane Huot.

5.1 Design

This is a [4×3] within-subject controlled experiment with two factors - *Technique* and *Difficulty*. I will explain each factor in the subsections.

Setting with instructions in 4 formats

Participants were asked to perform *Setting* tasks on a control surface. They were given instructions on a mobile phone to guide the operations. The instructions were in four formats: *AR+Feedback*, *AR*, *Text* and *Picture*. The first two conditions were to test how much benefits the added feedback could bring to AR. The latter two were chosen as the baseline as they are the common medium that people use to record knowledge or experiences.

All instructions were displayed with the mobile device

All instructions were displayed with the same mobile device. Although common instructions can be printed in paper, displayed on the computer, or even recorded as audio resources, the mobile device was chosen as the platform for all the conditions in this experiment. Because the objective was not to compare AR instructions with all possible instructions. Choosing a unique platform was also to eliminate unnecessary noises in the experiment.

5.1.1 Apparatus

Experiment Setup

Figure 5.1 shows the experiment environment. Setting operations were performed on a control surface, with which the prototype in Section 3.3.2 was implemented. The instructions were displayed on a HTC Desire mobile phone (display: 3.7 inches, resolution: 480×800 px, weight: 135 g, dimensions: 119×60×11.9 mm), running Android 2.3.3 operating system.

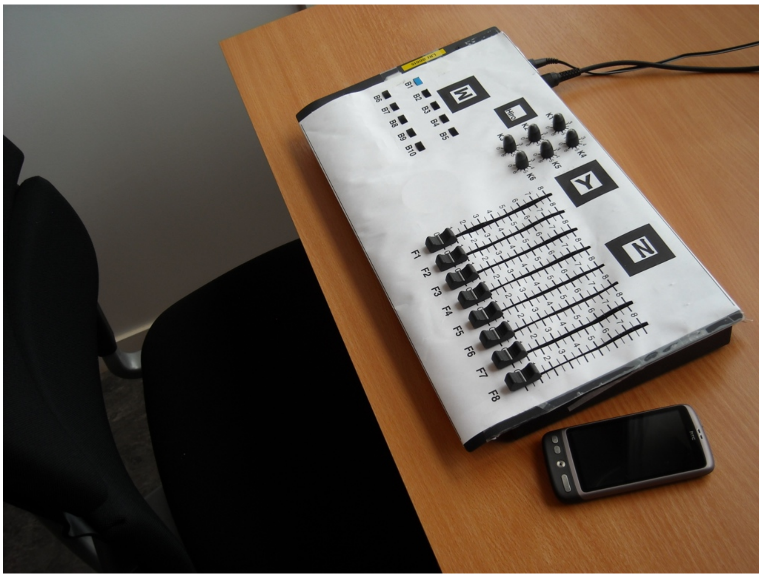


Figure 5.1: Experiment environment.

Appearance of the physical interface

The control surface was covered by white paper to hide the unused controls. The purpose was to avoid distracting participants and allow them to focus on the controls used for the tasks, thus to avoid the uncontrolled complexities such as extra visual search of controls. Ten buttons, six knobs and eight faders were in use for the experiment. Scales were printed on the white paper with black lines and arabic numbers.

5.1.2 Tasks

One Setting included pairs of controls and values

One *Setting* task required a participant to set a number of controls on the panel as instructed with the corresponding TECHNIQUE on the mobile phone, in any order. Every setting task included all three types of controls – buttons, knobs and sliders. For each type of controls, the ranges of possible values were (i) 0 or 1 for buttons, (ii) 1 to 5 for knobs and (iii) 1 to 7 for sliders. Both knobs and sliders had a 0.5 resolution (interval between two consecutive values).

Participate pressed a button to finish a trial, passes in case of Success

When a participant thought he has finished one task, he should press a hardware button on the mobile device, which was the optical button on the used phone. The task was finished in cases of success or timeout. A task was successfully performed when all the physical controls in the setting were in the correct value, and all other controls stayed in the initial states. For continuous controls, it was counted to be correct when the value of physical control had a derivation of less than 0.3 from the correct setting.

Each validation of failure was counted as an error

A failure was validated when the participate pressed the finish button but the physical settings were in fact wrong. In the case of a failure, the participant was forced to stay and continue the trial, until he corrected the setting or a timeout occurs. I will explain the design for timeout later. Every validation of a failure was counted as an error.

Settings were randomly generated and counterbalanced

In order to avoid the learning effects, the settings were randomly generated for each difficulty level, so that none of the participants performed the same setting twice during the experiment. To eliminate the effect of the settings themselves, they were counterbalanced across participants.

5.1.3 Techniques

- Push button B3
- Set Knob K2 to 3.0
- Set Fader F6 to 1.5

Figure 5.2: Text instruction

Control-value pairs in lines, no grouping, randomized order

Text instructions were displayed with a control-value pair on each line, as shown in (Figure 5.2). To avoid a

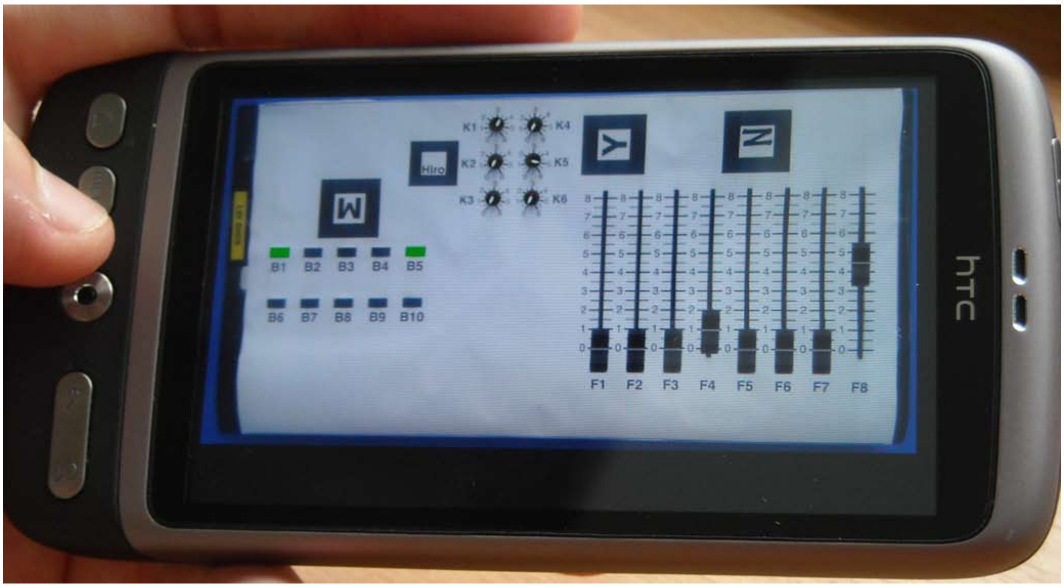


Figure 5.3: Picture instruction

potential order effect inherent for this line-based presentation, the order of the lines for a setting was randomized across participants. The instruction lines were not grouped by control types, because the preferred grouping can be dependent on user habits or preferences. Grouping of controls might introduce a bias depending on the participants.

Picture instructions were presented as 1024 × 537 pixels images for settings (Figure 5.3). This was to mimic a common situation that people take a picture of a setting to keep record of it. The picture was initially fully visible and participants were able to pan and zoom by dragging and pinching to get a better view of the controls.

Pictures were programmatically generated

Since all the setting tasks were generated before the experiment, the picture instructions had to be prepared beforehand. The pictures were programmatically composed with a plain panel image and control images placed in transformed positions (for sliders) or orientations (for knobs). The buttons to push were highlighted with green marks. There was no perspective or quality difference among the pictures.

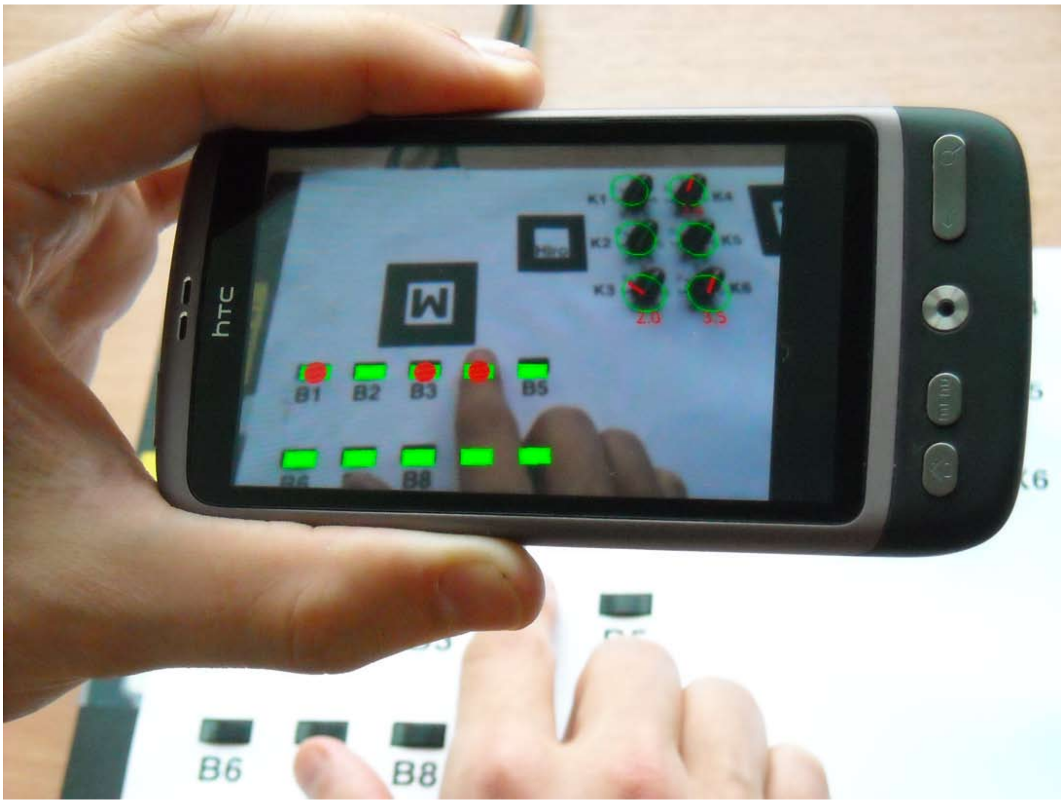


Figure 5.4: AR instruction without dynamic feedback

Graphical overlay with highlighted outline and values

AR instructions were presented in a vector graphics layer on top of the device camera input (Figure 5.4). Every control of the panel was highlighted into the AR layer with green outlines. For sliders and knobs, each control in a setting was annotated with a value bar, which was a red mark crossing the outline at the designated position to indicate the value. The numeric value to reach was written beside the red mark (for sliders) or the outline (for knobs). The buttons were highlighted by green squares and the ones to be pushed had a red circle on the outline.

Same graphics as AR, but with color change as feedback

AR+Feedback was similar to *AR* for the graphical layout (Figure 5.5). But the color of widgets was updated in real-time while physical controls were manipulated. When a control reached the correct value, the value indication (bars and numbers for sliders and knobs, circles for buttons) changed from red to blue immediately. If a wrong

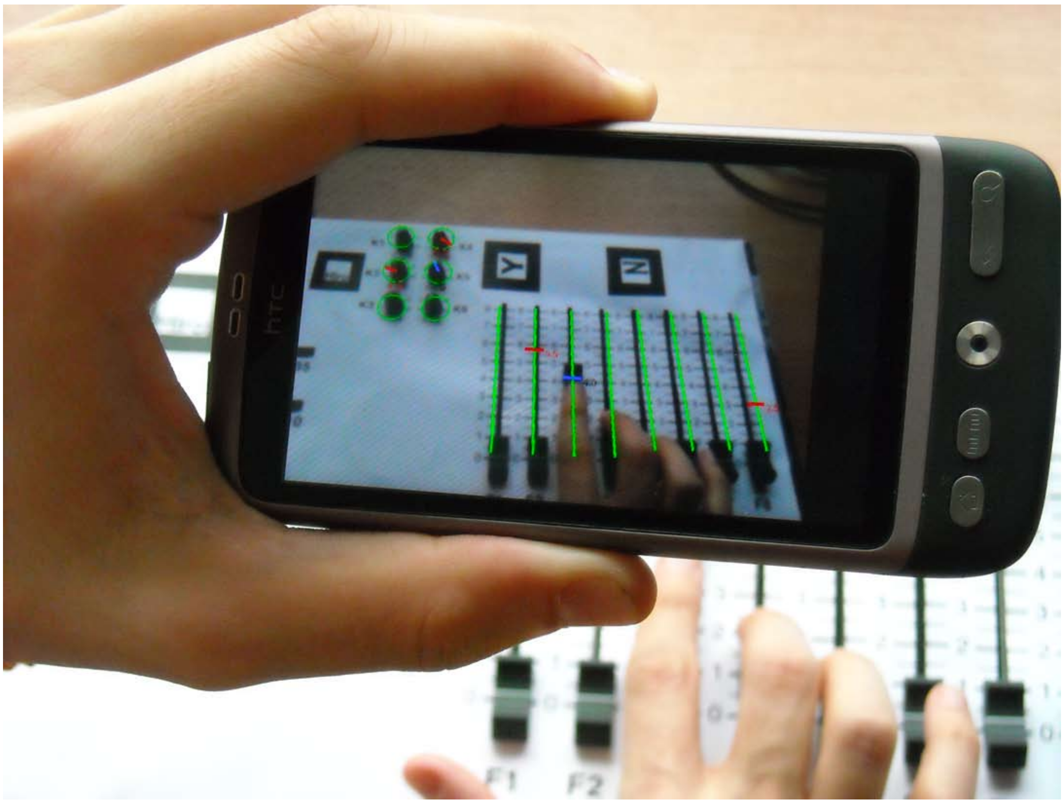


Figure 5.5: AR+Feedback instruction with dynamic color change as feedback

control was moved, the outline of this control turned purple.

5.1.4 Define Task Difficulty

Setting difficulty as an independent variable

In common sense, the task difficulty might have a large influence to the performance of each technique. For example when there are 2 or 3 controls to manipulate, all techniques might not perform very differently. But when there are a lot of controls in one setting, a long list of text might be difficult to read and follow, while graphical instructions become much more convenient with the provided spacial information. So DIFFICULTY was introduced as an independent variable to manipulate in the experiment.

Select the Number of Controls to define Difficulty

A setting is more difficult when it requires longer time to finish or it causes errors more easily. DIFFICULTY of one setting can be influenced by many possible parameters. For example, the number of controls, the number of control types included, the concurrent manipulation of multiple controls, the precision required for continuous controls, the offset for each control to move, and so on. To define the levels of DIFFICULTY and ensure that different setting tasks have the same DIFFICULTY, the parameters of DIFFICULTY need to be selected. It was decided to define the DIFFICULTY only by the number of controls. Since the settings were generated programmatically, number of controls was easy to control and the levels of it were easy to define. All the other confounding variables were eliminated by the rules applied in generating the setting tasks. The rules are about the control types and control values.

Control type

Include all control types

- Every setting includes all three types of controls. Since the controls are physically arranged in group of types, the switching between different types of controls can influent the performance. So including all types of controls in every setting is to eliminate this effect.

No duplicate controls in one setting

- In one setting, the same control should not be manipulated more than once.

Balanced control number for each type

- The number of controls of each type is balanced. This means every setting has equal number of controls for each type of control.

Switch control type after 1 or 2 lines in text instruction

- Since text instruction is inherently sequential, the generated sequence of a setting might lead a participant's operation. Thus there are no more than two adjacent lines in text instructions about the same type of control. This means if a participate operates with the sequence of text instructions, the control type is switched after one or two operations.

Control value The values of controls correspond to the physical positions. So it takes different time to reach different values for one control. In order to eliminate this effect, there is equal number of controls in low, medium and high values for the continuous controls (sliders and knobs) in one setting.

5.1.5 Pilot Study - Define Levels

An informal pilot study was conducted with 2 participants and 6 difficulty levels (3, 6, 9, 12, 15, 18 controls) to validate the rules and determine the difficulty levels to use in the study. Not all 6 difficulty levels was used due to the constraint of experiment time. Fatigue effect is likely to be introduced if an experiment is too long. The levels of the parameters was defined based on the pilot study.

Selected difficulty levels and timeout according to pilot result

The result suggested that settings with the same number of controls had equivalent performances. Three levels of difficulty out of six were selected to control the experiment time to be completed within one hour. It was defined to be 3 controls for *Easy*, 9 for *Medium*, 18 for *Hard* difficulty. Timeout for a trial was defined to be a time that a normal task would be not likely to reach. This was to prevent unexpected situations that caused the experiment to last for too long. For example, a participant was distracted and did not perform a task properly. Timeout was defined as 2, 4, and 6 minutes for the tasks in difficulties of *Easy*, *Medium* and *Hard* respectively.

5.2 Hypotheses

Based on the related work and the experience from the prototypes and pilot study, we made the following hypotheses:

- *H1:* With respect to performance speed, (a) *AR+Feedback* outperforms *AR*, (b) both *AR+Feedback* and *AR* outperform *Text* and *Picture*. Both performance improvements increase with task difficulty.
- *H2: AR* techniques are less error-prone and should facilitate the correction of errors.
- *H3: AR+Feedback* instructions are preferred to text, pictures and AR instructions by users.

AR reduces alternating attention, feedback helps participants focus on tasks

The "In-place" techniques (*AR* and *AR+Feedback*) should outperform *Text* and *Picture* since they reduce alternating attention during the tasks. The advantage may increase with task difficulty as *Text* and *Picture* require more memorization when the task includes many operations. *AR+Feedback* should also perform better than *AR*, because it helps focusing on the task instead of switching focus to check the status of physical controls.

AR provides in-place overview for checking values, and feedback informs mistakes immediately

AR is possibly less error-prone because people could hover the phone above the panel and check the correctness through the overlaid overview. *AR+Feedback* should be less error-prone because the added feedback informs the correctness directly so that participants do not even have to check the value or position.

AR was predicted to be preferred

Finally, the AR techniques should save people some effort of searching for controls. *AR+Feedback* technique even checks the correctness automatically. With these advantages it was predicted that they would be preferred by participants.

However, these are all guessed reasons and they require further experiments to be validated.

5.3 Procedure

Participants information

16 participants were recruited, twelve men and four women, all right-handed. Their ages were between 24 and 44. None of them had any experience with AR applications, but 4 were frequent users of surface controllers, e.g., mixing consoles or guitar amplifiers and 9 owned a smartphone.

Participants were given an introduction sheet with text to read before starting the experiment. The introduction sheet can be found in Appendix A.

Practice for all techniques in the beginning, one trial to remind before measurement

In the beginning of experiment, participants went through a practice session for all the techniques with the guidance of the operator. There were at least three practice trials for each technique. They were asked to practice for each technique in easy and medium difficulties till they felt familiar with it. The initial design was to let participants practice for each technique right before the measured session for this technique. This decision was changed after the pilot study. Because it was observed that the participants needed to get used to manipulating this control station. The speed of their operations had a rapid increase after some training due to the familiarity with the operations. A complementary one-trial *recall* in easy or medium difficulty was set before they started the measured trials for each technique.

Participants thought about their strategies in practice session

In the practice session, participants were asked to think about their own strategies for every technique and try to perform the tasks as fast as possible. Participants were free to use portrait or landscape mode of the mobile phone. And they were also told to minimize errors. From the observation, most of them learned the cost of errors after some training and became careful while operating.

Reset before a trial starts

One *Trial* was one Setting task introduced in earlier section. In the end of the trial, the participant was instructed to reset the sliders and knobs to initial value zero. Since the state of each control was tracked, the program allowed participants to go to next trial only after everything was reset. Participants were asked to put their operating hand at a default position in front of the panel before starting the next trial, so every trial started from the same condition.

Counterbalanced both factors across participants

Trials were grouped into *Blocks* by TECHNIQUE. The presentation order of TECHNIQUE and DIFFICULTY was counterbalanced across participants using a Latin Square[1]. To make sure both factors were counterbalanced, there should be at least (4 × TECHNIQUE) × (3 × DIFFICULTY) = 12 participants.

Two Replications

Each TECHNIQUE × DIFFICULTY condition was replicated twice. Given the limited experiment time and according to the time costed in pilot study, it was decided to have two replications for the measured trials. This meant two data points were collected for each condition. In the end, (4 × TECHNIQUE) × (3 × DIFFICULTY) × (2 replications) × (16 × PARTICIPANT) = 384 measured trials were collected.

---

[1] `http://www.maths.qmul.ac.uk/~rab/DOEbook/`

Participants ranked each technique and gave comments

The experiment lasted about 40–50 minutes. After the experiment, participants were asked to fill in a questionnaire (Appendix B). They ranked from 1 to 4 for the worst to the best for each TECHNIQUE in general, and then ranked again for each TECHNIQUE in each DIFFICULTY level respectively. Afterwards they were interviewed and talked about their opinions for each technique, and explained reasons about their strategies for performing the tasks.

5.4 Measurements

The following three measurements were collected in the experiment:

- *TrialTime*, the trial completion time. It was counted from the instruction's appearance to the time a participant successfully accomplish the task by pressing the physical button, or a timeout occurred in the case of failure.
- *ReactionTime*, the time interval from the appearance of the instruction to the occurrence of the participant's first action on the control surface.
- *Errors*, the number of errors by trial. An error was counted whenever a participant clicked the physical button to finish a task and the system detected the setting was not correct yet.

5.5 Implementation

Using Touchstone for experiment control and log

The TouchStone [Mackay et al., 2007] platform was utilized to facilitate the experiment control. This software runs on a computer and acts as a server to control the states of the experiment. The mobile phone as a client communicates with the server via OSC messages.

TouchStone controls the designed experiment as state machine

Touchstone controls the start of a trial and a block, and what technique and difficulty the current trial is presenting. It is actually a state machine, which sends commands for starting and ending trials or blocks with experiment factors as parameters, and waits for the responses from the client with the measurement. The experiment design is encoded in an XML file by defining all the factors of each block and trial. It logs the measurements to text files by trials and blocks with participant ID. So if the experiment is interrupted due to technical reasons, it can start from the current block.

Structure of client software

The mobile phone receives the states control commands from TouchStone and responds to it with measured time and error values. The experiment program consists of several parts. The mobile phone communicates with two servers, both via OSC messages. One is the WILDInputServer for getting updated about user operation on the physical control surface (Section 4.3.2). The other one is the Touchstone platform for exchanging the experiment data.

Physical control values were tracked through the experiment

The states of all the physical controls are actively received, as explained in the implementation of software prototype (Section 4.3). They are stored in a HashMap[2] object `allControlValues`, which has 24 pairs of control name and value. It is a variable in Class `WidgetsManager` (Figure 4.3). This class is introduced in the previous chapter as it is related to the functions for providing the graphical feedback.

Track the status of physical controls for 3 purposes

In the software, the tracking of physical controls runs for the entire experiment. This function is used for three purposes. First is to mock up the detection of the status of physical controls in order to provide feedback in *AR+Feedback* technique, as explained in the last chapter. Second is to check the correctness of a performed setting for experiment control, such as deciding if the trial should be passed or an error should be counted. The third is to ensure all the controls were reset before each trial. In case some controls are not reset, it also warns the participant to reset the missed controls.

Classes for Settings

The predefined settings for all tasks are stored in `Setting.txt` file. Every `Setting` includes a `SettingName`, `Difficulty`, `NumberOfControls`,

---

[2] `http://docs.oracle.com/javase/6/docs/api/java/util/HashMap.html`

and the `Name`, `Value` and `ValueRange` for each `Control`. The `Setting` class loads the `Setting.txt` file and holds functions to provide the corresponding instruction according to a setting and the specified TECHNIQUE type. The `Control` and `ControlType` classes manage the properties and methods for controls in settings, such as changing a control's value range.

Chapter 6

Analysis

After collecting the experimental data, a statistical analysis was performed. The effects of TECHNIQUE and DIFFICULTY regarding *Performance Time* and *Error Rate* are analyzed from several aspects. Part of the hypotheses are validated based on the concluded effects. The results of subjective evaluation are consistent with the quantitative conclusions, and some insights for each technique are revealed.

Statistic analysis for the effect of TECHNIQUE and DIFFICULTY regarding *Performance Time* and *Error Rate*

The statistical analyses use the REML technique for two-way repeated measure ANOVA and Tukey HSD Post-hoc tests for pairwise comparisons. All Tukey HSD Post-hoc tests use 0.05 as alpha value. The software tool for analysis is JMP 9.0.

This chapter discusses the procedure and results of the statistical analysis. I will explain the analysis methods and the concluded effects, and discuss the reasons and issues based on observations and participants' comments.

6.1 Prepared Data

As mentioned in last chapter, 384 measured trials were collected from the experiment. 3.03% of them appear to be outliers from a Grubbs' ESD test [Grubbs, 1969]. They are defined as trials with a total time greater than two standard

Removed 3.03% outliners

deviations from the mean.

The following analysis was performed after removing the outliners.

6.2 Performance Time

No effect of learning, fatigue or setting task

To begin with, I ran an ANOVA test with the factors TECHNIQUEORDER ×DIFFICULTYORDER to test if there is any learning or fatigue effect. The result shows there is none. In the same way, I tested the effect of settings and there is also no effect shown. This validates the approaches of generating settings with defined difficulty levels.

6.2.1 Trial Time

Significant effects on both TECHNIQUE and DIFFICULTY as well as their interaction

A full-factorial analysis was performed with the model TECHNIQUE×DIFFICULTY×Random (PARTICIPANT). Significant effects of TECHNIQUE ($F_{3,45} = 45.62$, $p < 0.0001$) and DIFFICULTY ($F_{2,30} = 299,84$, $p < 0.0001$) and a significant TECHNIQUE×DIFFICULTY interaction effect ($F_{6,90} = 27.59$, $p < 0.0001$) on *TrialTime* were found (see Table 6.1).

Source	Nparm	DF	DFDen	F Ratio	Prob > F
Technique	3	3	45	45,6157	<,0001*
Difficulty	2	2	30	299,8361	<,0001*
Technique×Difficulty	6	6	90	27,5933	<,0001*

Table 6.1: ANOVA result of TrialTime with Technique and Difficulty

Effect of Difficulty

As indicated in Figure 6.1, *TrialTime* increases with DIFFICULTY and all difficulty levels are significantly different (*Easy* $= 11.8s$, *Medium* $= 28.4s$, *Hard* $= 56s$) (see Table 6.2). In this table, levels not connected by the same letters are

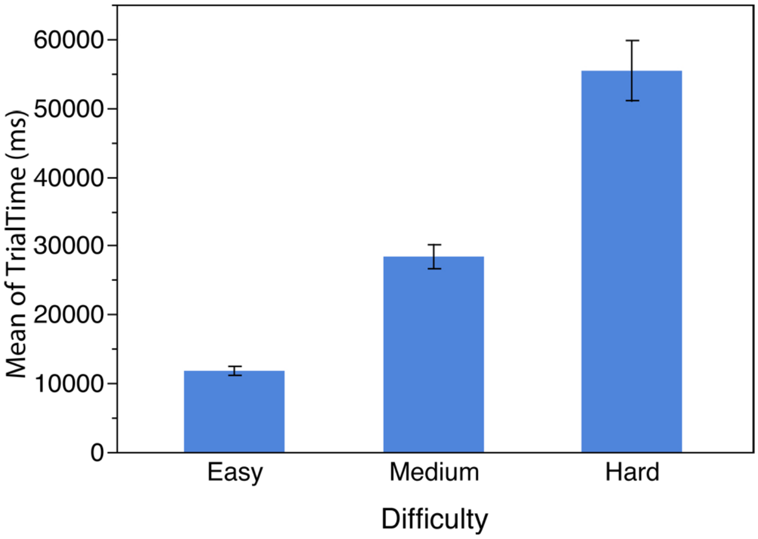


Figure 6.1: Means of TrialTime by Difficulty. Each error bar is constructed using a 95% confidence interval of the mean.

Level				Least Sq Mean
Hard	A			56045,750
Medium		B		28398,766
Easy			C	11783,227

Table 6.2: Tukey HSD Post-hoc test result for TrialTime with Difficulty.

significantly different. The same applies to all the tables of Tukey HSD Post-hoc test results in this chapter.

Effect of Technique

AR+Feedback (23.9s) is significantly faster than all other techniques (Figure 6.2), and *Text* (43.9s) is significantly slower (*AR*= 29.4s and *Picture*= 31.1s) (Table 6.3).

Interaction of Technique and Difficulty

The TECHNIQUE×DIFFICULTY interaction effect shows a

Interaction effect between two factors is interesting

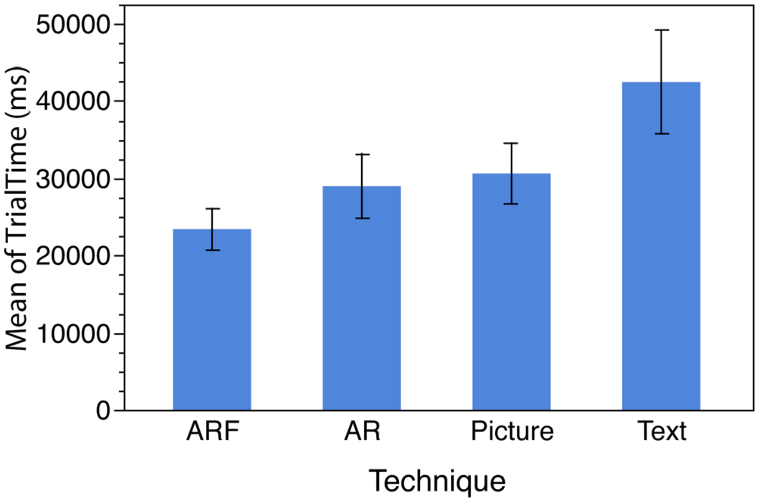


Figure 6.2: Means of TrialTime by Technique. Each error bar is constructed using a 95% confidence interval of the mean.

Level				Least Sq Mean
Text	A			43874,250
Picture		B		31144,583
AR		B		29404,688
ARF			C	23880,135

Table 6.3: Tukey HSD Post-hoc test result for TrialTime with Technique

more interesting insight into the performance of each technique according to the task difficulty. As shown in Figure 6.3 , *TrialTime* exhibits a rapid increase with *Text*, but a lower slope for *AR+Feedback*, with *AR* and *Picture* in between.

Analyze the effect of TECHNIQUE in each DIFFICULTY level

This trend is confirmed by a TECHNIQUE×Random (PARTICIPANT) by DIFFICULTY ANOVA test, which reveals significant effects of TECHNIQUE for each DIFFICULTY level. See Table 6.4 for the result of ANOVA and Table 6.5 for the result of Post-hoc test.

No need to switch focus for all techniques, not much difference is shown

Easy As only 3 controls need to be set, the values can be

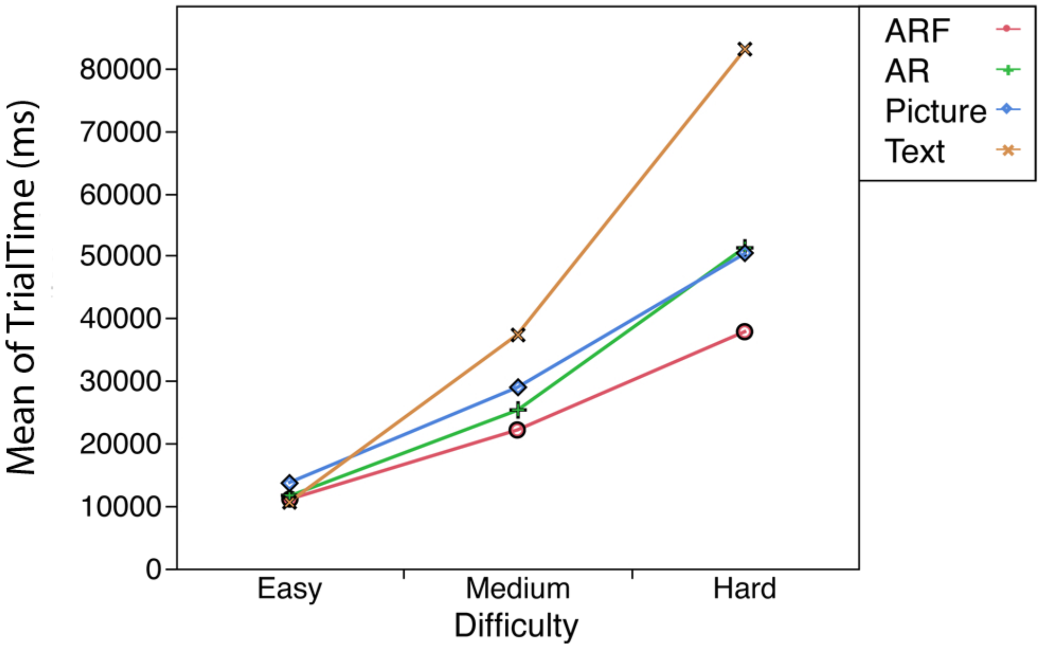


Figure 6.3: Means of TrialTime by Technique and Difficulty.

Source	Difficulty	Nparm	DF	DFDen	F Ratio	Prob > F
Technique	Easy	3	3	45	5,8188	0,0019*
	Medium	3	3	45	29,0032	<,0001*
	Hard	3	3	45	36,2818	<,0001*

Table 6.4: ANOVA result for TrialTime with Technique by Difficulty

Difficulty	Level				Least Sq Mean
Easy	Picture	A			13766,875
	AR		B		11597,250
	ARF		B		11085,531
	Text		B		10683,250
Medium	Text	A			37276,000
	Picture		B		28933,906
	AR		B	C	25249,188
	ARF			C	22135,969
Hard	Text	A			83663,500
	AR		B		51367,625
	Picture		B		50732,969
	ARF			C	38418,906

Table 6.5: Tukey HSD Post-hoc test result for TrialTime with Technique by Difficulty

memorized. The switch between the work space and instructions is not necessary. Only *Picture* is performed more slowly than other instruction methods (Table 6.5). This is likely due to the time required to find the controls to set in the pictures. It might be more time-consuming than reading and memorizing three textual instructions or skimming through the overview with the AR techniques.

Text performance drops faster than others, *AR+Feedback* performs the best

Medium At the *Medium* level, the techniques starts to show some differences. The performance of *Text* drops significantly and *AR+Feedback* performs faster than *Picture* (Figure 6.3). *Text* probably makes it more difficult to keep track of the instructions in the list while switching between the work space and instructions. Graphical instructions are more efficient: *Picture* and *AR* are close, possibly because they have similar presentations. *AR+Feedback* starts to perform better because it allows the participant to set the control without switching from the device to the panel with the color feedback. In medium level with 9 controls, attention switches are needed more frequently than in the easy level because these instructions are more difficult to memorize.

Large differences among techniques are shown, but *Picture* and *AR* performs similar

Hard At the *Hard* level, most differences in *Medium* are exacerbated. *Text* is by far the slowest technique, *Picture* and *AR* have similar performance, and *AR+Feedback* is the fastest (Fig. 6.3). The small difference in performance between *AR* and *Picture* could be explained based on the observation. Many participants used *AR* in a similar way as using *Picture*, which will be explained later in Section 6.5.1. For *AR*, the in-place benefits are narrowed by issues such as occlusions or low resolution. *AR+Feedback* allows participants to set the controls while keeping attention on the on-screen instructions. This might be the reason for the significant improvement of performance. In *Hard* level, it reduces attention switches between the instructions and the physical controls to the largest extent.

More details about how the participants operated with each technique will be provided in the observation section (Section 6.5).

To validate the significant effects of DIFFICULTY for each TECHNIQUE, I also tested DIFFICULTY×Random (PARTICIPANT) by TECHNIQUE in the same way. The result were unsurprising. For each technique, there is a significant difference between each difficulty level. Table 6.6 and Table 6.7 present the details of the result.

Analyze the effect of DIFFICULTY for each TECHNIQUE

Source	Technique	Nparm	DF	DFDen	F Ratio	Prob > F
Difficulty	ARF	2	2	30	114,4377	<,0001*
	AR	2	2	30	93,1465	<,0001*
	Picture	2	2	30	69,1335	<,0001*
	Text	2	2	30	201,4867	<,0001*

Table 6.6: ANOVA result for TrialTime with Difficulty by Technique

Technique	Difficulty				Least Sq Mean
ARF	Hard	A			38418,906
	Medium		B		22135,969
	Easy			C	11085,531
AR	Hard	A			51367,625
	Medium		B		25249,188
	Easy			C	11597,250
Picture	Hard	A			50732,969
	Medium		B		28933,906
	Easy			C	13766,875
Text	Hard	A			83663,500
	Medium		B		37276,000
	Easy			C	10683,250

Table 6.7: Tukey HSD Post-hoc test result for TrialTime with Difficulty by Technique

H1 Validated

To recall from last chapter, H1 is: "With respect to performance speed, (a) *AR+Feedback* outperforms *AR*, (b) both *AR+Feedback* and *AR* outperform *Text* and *Picture*."

(a) is validated by the significant effect between *AR+Feedback* and *AR* in both *Medium* and *Hard*. (b) is only partially supported by the significant effect between *AR+Feedback* and all other techniques.

(a) is validated, (b) is partially validated

AR does not perform better than *Picture*

There is no significant difference between *Picture* and *AR* in any difficulty levels. As shown in Figure 6.3, *Picture* even performs slightly better than *AR* for the *Hard* level.

6.2.2 Reaction time

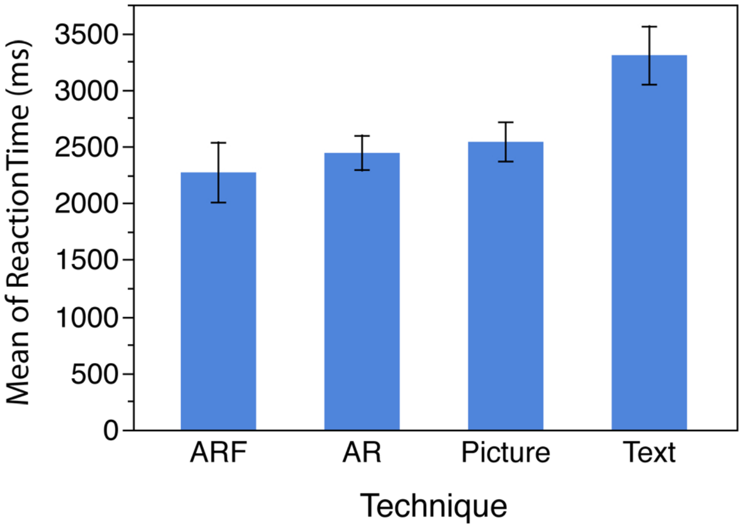


Figure 6.4: Means of ReactionTime by Technique. Each error bar is constructed using a 95% confidence interval of the mean.

The same analysis process is performed for the *ReactionTime*. It is again a full-factorial analysis with the model TECHNIQUE×DIFFICULTY×Random (PARTICIPANT).

Source	Nparm	DF	DFDen	F Ratio	Prob > F
Technique	3	3	45	13,1920	<,0001*
Difficulty	2	2	30	7,7803	0,0019*
Technique×Difficulty	6	6	90	2,0575	0,0661

Table 6.8: ANOVA result of ReactionTime with Technique and Difficulty

Significant effects for both TECHNIQUE and DIFFICULTY

Figure 6.4 and 6.5 show the means of *ReactionTime* for TECHNIQUE and DIFFICULTY, respectively. There are significant effects for both TECHNIQUE ($F_{3,45} = 13.19$, $p < 0.0001$) and DIFFICULTY ($F_{2,30} = 7.78$, $p = 0.0019$), see Table 6.8.

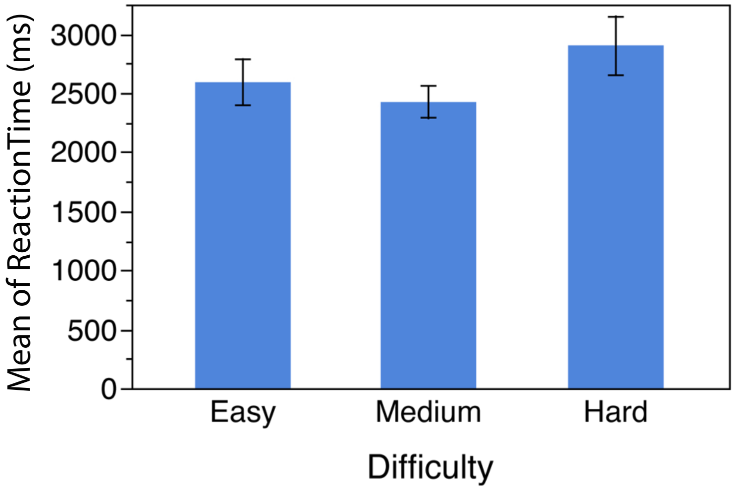


Figure 6.5: Means of ReactionTime by Difficulty. Each error bar is constructed using a 95% confidence interval of the mean.

Level			Least Sq Mean
Text	A		3325,3333
Picture		B	2530,1354
AR		B	2446,3333
ARF		B	2281,2708

Table 6.9: Tukey HSD Post-hoc test result for ReactionTime with Technique

Difficulty Tukey HSD Post-hoc test result shows that only *Hard* level is significantly slower than others (Table 6.10). The more difficult the setting is, the more time participants needed to have a glance of it before starting any action. But the difference between *Easy* and *Medium* is not statistically significant at this point.

Hard is slower than *Easy* and *Medium*

Technique The Post-hoc test result reveals that for TECHNIQUE only *Text* is significantly slower than other techniques (Table 6.9) *Text* requires more time to get an overview because there is no spacial information presented. The graphical instructions take approximately the same time for reaction.

Text takes more time to react than others

Level			Least Sq Mean
Hard	A		2903,4297
Easy		B	2592,0156
Medium		B	2441,8594

Table 6.10: Tukey HSD Post-hoc test result for ReactionTime with Difficulty

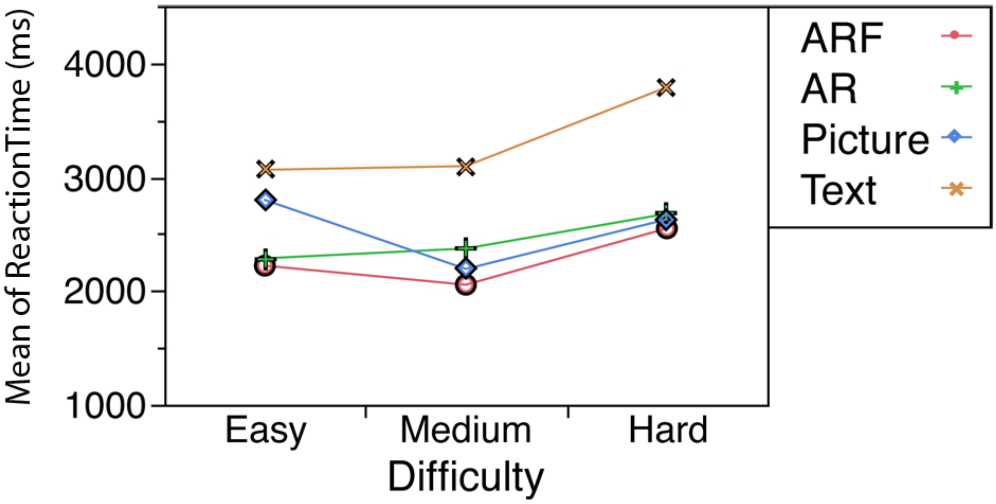


Figure 6.6: The interaction between Technique and Difficulty for ReactionTime

No significant interaction

Interaction The interaction between these two factors regarding *ReactionTime* is not significant. This is because reaction time should be more or less consistent when there is no learning or fatigue effect. However, Figure 6.6 shows *Picture* does not follow the same trend as other techniques when DIFFICULTY increases.

ReactionTime has significant effect on *Picture* and *Text*

I ran another ANOVA test on *ReactionTime* with DIFFICULTY by TECHNIQUE. The result shows DIFFICULTY does not have significant effect for *AR+Feedback* ($F_{2,30} = 1.9140, p = 0.1651$) and *AR* ($F_{2,30} = 2.81, p = 0.08$). However, DIFFICULTY has significant effects on *Picture* ($F_{2,30} = 7.34, p = 0.003$) and *Text* ($F_{2,30} = 4.33, p = 0.02$).

Picture does not follow the same trend as others when DIFFICULTY increases

Post-hoc test results show that for *Text*, the significant difference is between *Hard* and other difficulty levels. But for *Picture*, *ReactionTime* is approximately the same for *Easy* and *Hard* levels, while it is significantly less for *Medium* level

(Table 6.11).

Possible reasons for this effect

Easy level requires more time to react than *Medium*. This could possibly because of the sparse density of the controls. Participants need some time to find the controls from one to another. When there are more controls, the relative distances between controls are smaller so that they are easier to find. In *Hard* level, much time is required probably due to the amount of information to be parsed.

Technique	Difficulty			Least Sq Mean
Picture	Easy	A		2798,1562
	Hard	A		2599,4375
	Medium		B	2192,8125
Text	Hard	A		3805,6563
	Medium		B	3107,2188
	Easy		B	3063,1250

Table 6.11: Tukey HSD Post-hoc test result for ReactionTime with Difficulty by Technique

Setting Time

Setting time is similar with *TrialTime*

The *SettingTime* (trial completion time subtracted by *ReactionTime*) was also tested with the same analysis process. Unsurprisingly, it has the same significancy as the result of *TrialTime*. Thus the explanation of the analysis for *SettingTime* is omitted.

6.3 Error Rate

Errors often happen in consequences

As introduced previously, an error is counted when a wrong setting detected by the system after the participant clicks the finish button. Before timeout occurs, the participant must try to correct it till it is correct. With this design, it happens often that once an error occurs in a trial, more errors occur in the same trial consequently. This is because it is not easy to find the incorrect control.

Analysis errors with two variables

Due to above reasons, the error numbers are influenced by two effects: the chance to cause an error and how easy the error recovery is. Therefore, the statistical analysis is performed separately on two values: the number of trials with at least one error, which reflects the error occurrence, and the error number by trial for the trials with errors, which reflects the error recovery.

6.3.1 Number of Trials with Errors

13.4% of the trials contained at least one error. Here we use a boolean variable *ErrorTrial* to indicate if the trial has at least one error.

Significant effects on both TECHNIQUE and DIFFICULTY, no interaction

A nominal logistic ANOVA model for TECHNIQUE ×DIFFICULTY ~ *ErrorTrial* shows significant effects for TECHNIQUE ($\chi^2 = 11.40, p = 0.0098$) and DIFFICULTY ($\chi^2 = 30.63, p < 0.0001$), see Table 6.12. There is no significant interaction effect between these two factors.

Source	Nparm	DF	L-R ChiSquare	Prob > ChiSq
Technique	3	3	11,3957882	0,0098*
Difficulty	2	2	30,6258718	<,0001*
Technique×Difficulty	6	6	6,09670404	0,4124

Table 6.12: Nominal Logistic model for ErrorTrial with Technique and Difficulty

Only *Hard* is significantly different

From Figure 6.8 we can see an increase in *Errors* with DIFFICULTY, though the result of post-hoc test only show significance for *Hard* (26.56%) against *Medium* (11.72%) and *Easy* (3.13%), Table 6.13.

Level			Least Sq Mean
Hard	A		0,26562500
Medium		B	0,11718750
Easy		B	0.03125000

Table 6.13: Tukey HSD Post-hoc test result for ErrorTrial with Difficulty

Picture is significantly more error prone than AR+Feedback

For TECHNIQUE, *Picture* has the highest *Errors* and *Text*,

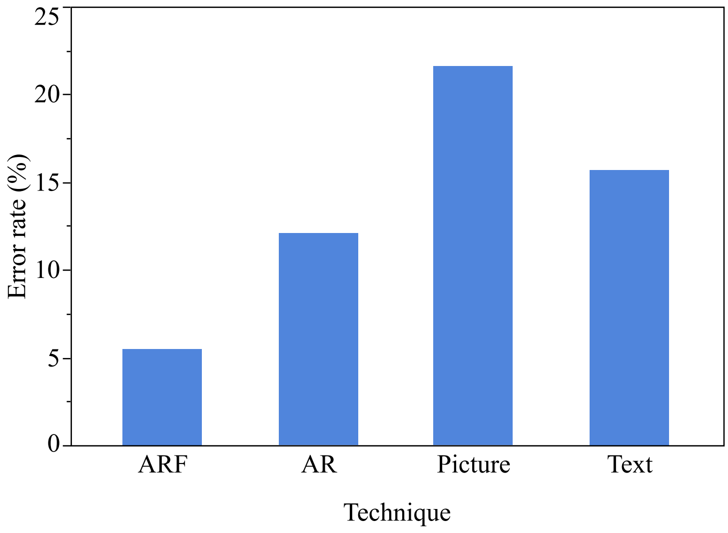


Figure 6.7: Error Rate by Technique

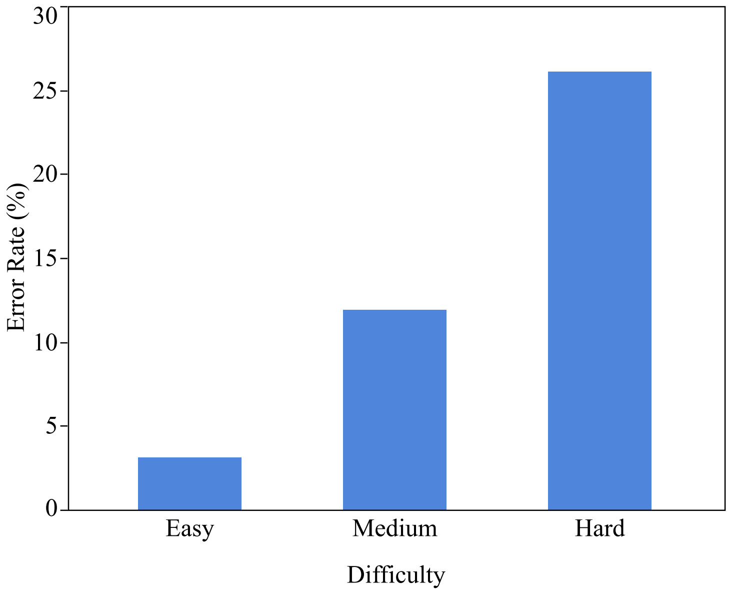


Figure 6.8: Error Rate by Difficulty

AR and *AR+Feedback* goes after it in order, see Figure 6.7. However the only significant difference is between *Picture* (21.88%) and *AR+Feedback* (5.2%). More details are in Table 6.14.

Level			Least Sq Mean
Picture	A		0,21875000
Text	A	B	0,15625000
AR	A	B	0,12500000
ARF		B	0,05208333

Table 6.14: Tukey HSD Post-hoc test result for Error with Technique

Explanation for each technique regarding *Errors*

Picture is always the most error-prone category in all DIFFICULTY levels. This is possibly due to the inaccurate estimation of values from the pictures. There are no numeric values written beside the controls for *Picture* instructions. *Text* is the second error-prone technique. It is not easy to check values with *Text* instructions because the participants have to read line by line and check the controls one by one. So in most cases they performed the setting once and tried to finish without checking. *AR* provides a good in-place overview for checking values. However one cannot rely on the spacial mapping between AR widgets and physical controls to check values, because even slight shaking of the graphical layer can offset the alignment. So participants still had to read the numeric values and check the controls one by one. With *AR+Feedback* the error check is very easy to do by simply checking the color.

Different impact of DIFFICULTY on each TECHNIQUE

Although there is no interaction effect between the two factors, Figure 6.9 indicates different impacts of DIFFICULTY on *Errors* for each technique. In *Easy* difficulty, *AR* has no error while *AR+Feedback* has almost equal *Errors* as *Text*. This is different in the other two difficulty levels, where *AR+Feedback* has the least error. The occurrence of errors could be caused by many attributes. For example, a participant might be too confident with using *AR+Feedback* and pushed a slider too much as he tried to operate as quickly as possible. However, no common user behavior was observed to explain this effect.

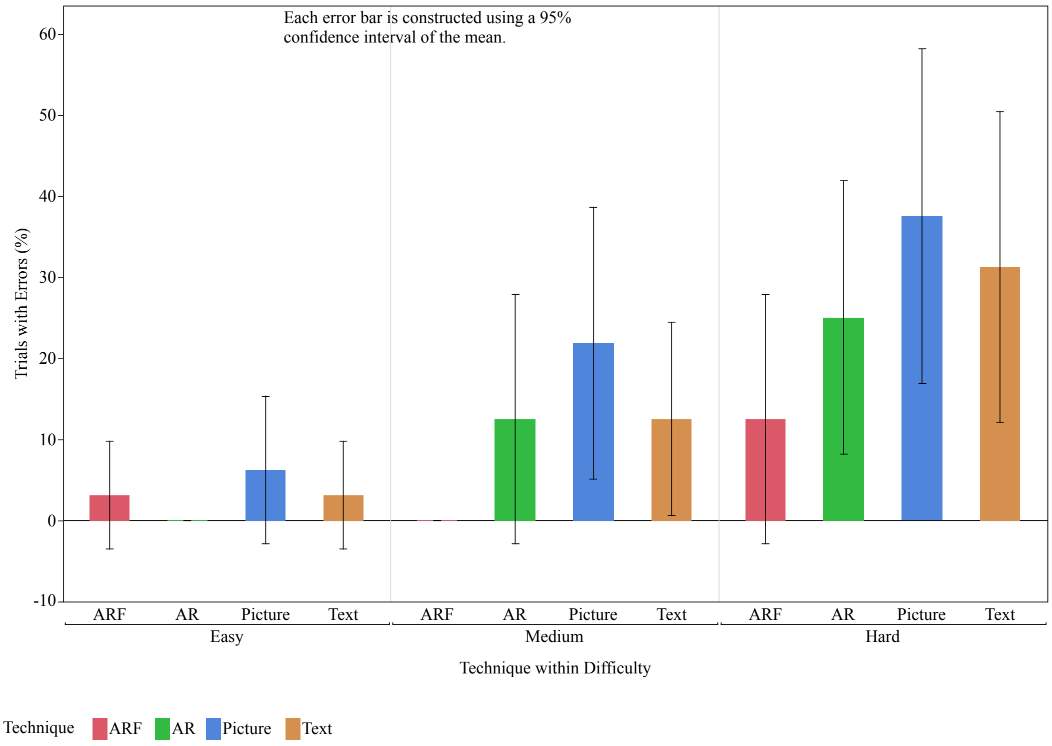


Figure 6.9: Error Rate by Technique and Difficulty

6.3.2 Number of Errors by Trial

For trials with errors, there are no significant effect detected on the number of errors with the factors TECHNIQUE and DIFFICULTY. The result is plotted in Figure 6.10.

There are not enough errors to conclude any more significant results according to the collected data. Error is not the most influential performance factor in our case, so the experiment is not designed for this. This would require a specially designed experiment to investigate error recovery.

Not enough errors to conclude significant effect

6.3.3 H2 Partially Validated

As a reminder, H2 is: "*AR* techniques are less error-prone and should facilitate the correction of errors."

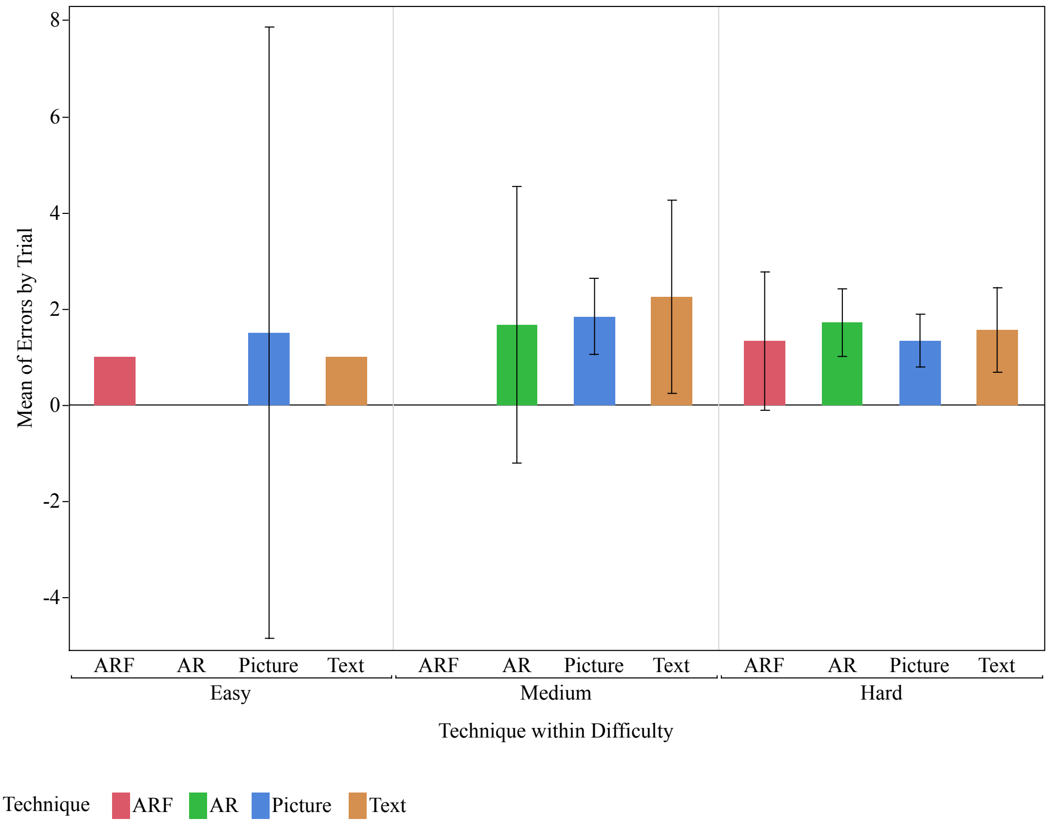


Figure 6.10: Number of Errors Per Trial by Technique and Difficulty

H2 partially supported

The results partially support H2: *AR+Feedback* and *AR* have fewer errors, but not all differences are significant. Only *Picture* is significantly more error-prone than other techniques.

Can not conclude error recovery

Also, there is no effect of TECHNIQUE or DIFFICULTY detected on the number of errors per trial for trials with errors. Therefore the error correction can not be concluded and this part of the hypotheses is not validated by this experiment.

It is observed that *AR+Feedback* helps to check values, thus possibly preventing error occurrence

However, according to the observation, *AR+Feedback* made it very easy to correct errors with the feedback. However, the errors were corrected immediately after the mistake was made, so there was no error counted for those. To evaluate this, it requires specific error correction tasks.

Finally, Error Rate alone cannot explain the difference in performance among techniques, as it does not exhibit the same effects as *TrialTime*. Alternating between subtasks has more influence on performance time than errors.

Error Rate can not explain the differences of techniques

6.4 User Preference

A nominal logistic ANOVA model for TECHNIQUE×DIFFICULTY ∼ *Rating* shows a significant effect of TECHNIQUE ($\chi^2 = 233.61309, p < 0.0001$) and a significant TECHNIQUE×DIFFICULTY interaction effect ($\chi^2 = 46.816, p = 0.0104$), see Table 6.15.

Significant effect on TECHNIQUE and the interaction

Source	Nparm	DF	L-R ChiSquare	Prob > ChiSq
Difficulty	9	9	2,50946255	0,9806
Technique	9	9	271,513482	<,0001*
Technique×Difficulty	27	27	76,2135076	<,0001*

Table 6.15: Nominal Logistic model for Rating with Technique and Difficulty

AR+Feedback is ranked as the preferred technique by all participants in general and by most of them for each difficulty level. It is followed by *AR*, *Picture* and *Text*.

AR+Feedback is preferred to others

At the *Easy* level, the ranking of *Text* is not as bad as in other difficulty levels (see Figure 6.11). This is likely because for 3 controls, *Text* is simple to grasp and use directly.

Text is not that bad in *Medium* difficulty

Level				Least Sq Mean
ARF	A			3,7343750
AR		B		2,4843750
Picture		B		2,3437500
Text			C	1,3750000

Table 6.16: Tukey HSD Post-hoc test result for Rating with Technique

In the end, post-hoc test results show significant effects of the rating between techniques, see Table 6.16. *AR+Feedback* is significantly more preferred than *AR* and *Picture*, and *Text*

Ratings are consistent with other measured results

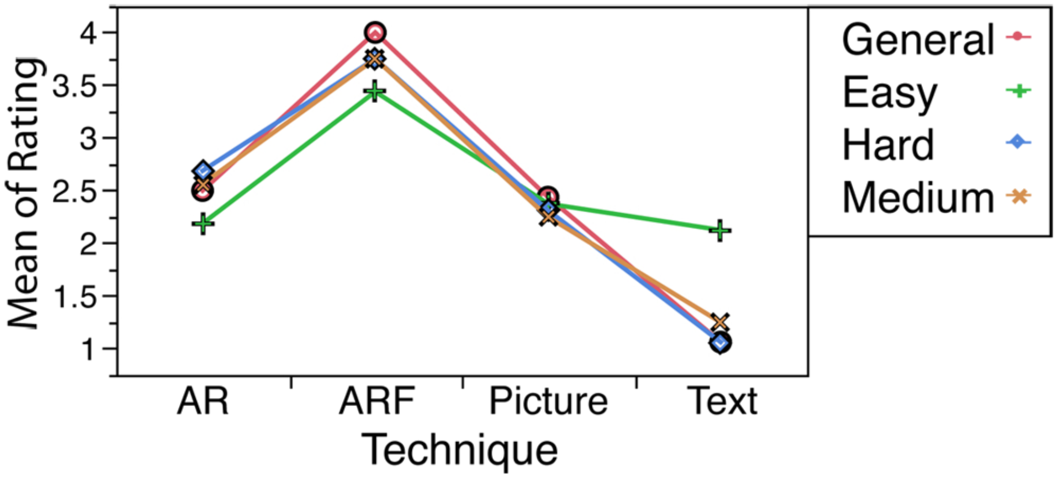


Figure 6.11: Participant Rating for Technique by Difficulty

is the least preferred. This is consistent with the results of quantitative analysis.

Reasons for preferences will be introduced with observations

While the quantitative study is only based on the performance time and errors, the user preference includes more subjective factors, such as the comfort of use. The comments from participants explain the reasons of their preferences. They are introduced together with the observations in the next section to give insights about the usability issues for each technique.

6.5 Observation

AR+Feedback was preferred to *AR* as it eased the problems and compensated for uncomfortableness

Generally , participants preferred *AR+Feedback* because it did not require them to switch between the device and the panel. The dynamic feedback eased the occlusion problems of *AR*. For *AR*, most participants raised the issue of graphical occlusion. Also, when they operated while looking through the screen, the small screen size and the resolution of the camera image impaired precise manipulation when not looking at the panel. Some participants also felt that it seemed unnatural to set the controls through the screen with both AR techniques. This was compensated by the benefits of real-time feedback of *AR+Feedback*.

Participants' comments matched their performance and the ratings, which supported the predicted reasons that led to the hypotheses.

6.5.1 AR

11 participants held the phone horizontally in order to have an overview, while 1 participant held it vertically in order to read numbers.

Most participants hold the phone horizontally

All 16 participants moved the phone away and operated while looking at the physical panel for knobs, 12 did this for sliders, 9 did this for buttons. One participant said he did not switch focus for buttons because it does not require precision.

Most participants did not operate while looking through the phone

I noticed that some participants used the AR technique in a similar way with *Picture*. They held the mobile phone over the panel for an overview, and then moved closer for a close-up of several controls in order to memorize the values. Instead of looking through the AR display, they moved it aside to interact with the controls directly. They then moved the AR display back to learn with the next set of controls.

AR is used similar with *Picture*

Participants were asked to explained the reasons why they use *AR* in a similar way with using *Picture*.

Reasons for using *AR* similar with *Picture*

- It was not precise enough to see real widget values through camera (7 participants).
- Zooming in by moving the phone took time (4 participants).
- Problems of occlusion (3 participants).
- The image was blurry while moving the phone (2 participants).
- It felt unnatural seeing my finger through the camera (2 participants).
- The image was smaller on the phone (2 participants).

Three types of occlusions were observed:

- The finger occludes the value of physical controls.

3 types of occlusion problems

- AR overlay occludes the physical controls and their values.
- AR overlay occludes the finger.

User comments explained some measured effect

The comments from participants provided some explanation of the measured results. One participant said "I look through the phone to find the slider to move." This indicated that information in-place is good for localizing widgets. Another mentioned, "(AR techniques are) better than picture because I don't have to zoom in on the screen." The comment "Help by just looking through the device, But too complicated for simple settings" could be an explanation for the non-significant effect of AR techniques in *Easy* difficulty. And the comment "Time to move the phone out of the way delayed me" also indicates that switching between the instruction and the physical controls is a major factor for the unsatisfying performance of *AR*. They also mentioned that the AR techniques are good for checking values.

6.5.2 AR+Feedback

Most participants did not switch focus in *AR+Feedback*

10 participants did not switch focus between AR overlay and physical interface at all. 4 participants switched focus occasionally. One explained the reason: "There is occlusion for my finger by the digital layer, especially for buttons". One participant occasionally switched for easy tasks.

Reasons for switching attention in AR+Feedback

2 participants switched focus every time, but used the color feedback to make a quick check in the end. One of them said:"It is easier to adjust the widget by directly watching the real objects as soon as I know the value. With the color feedback I am more confident." The other participant explained the reason as "fear of occlusion".

Comments are highly positive

The comments showed very positive assessment for *AR+Feedback*. "Good, especially for harder tasks, could do everything just through phone." "I don't need to look in real world, only display is enough when I see feedback." "Great with knobs, because fingers occlude the actual value, and with this you can stop turning it when it turns blue!"

One participant mentioned that *AR+Feedback* might interfere with learning because it provide too much assistance. This could be negative when it is important to learn the mechanism of the physical device. However as discussed in chapter 3, people need direct guidance for operations in many circumstances, where understanding the machine is not important.

Possibly prevent learning

6.5.3 Baselines

For *Text* instructions, 7 participants tried to group widgets by type to operate them together. 3 participants used thumb as indicator in order to find the step after switching the focus back from the panel, because it is easy to get lost after switching back from the physical surface to the text instruction.

Easy to get lost with text instruction

The comments showed it was not pleasant to operate with *Text* instructions: "It takes a lot of time.""Easy to lose oneself in the rows." "When list is long, more difficult to see error."

Text is not pleasant

For *Picture* instructions, most participants did not zoom in to view control values. 3 participants zoomed in to see values every time. One participant zoomed in occasionally, mainly for correcting error.

Most participants did not zoom in to view values

Some comments revealed the drawbacks of *Picture*: "Ok but have to switch visual attention." "Hard to read without zooming."

Picture requires attention switch, and it is hard to read values

Other comments reflected the reasons why *Picture* is sometimes preferred to AR techniques: "Efficient for easy tasks." "Faster to get what to be done, don't have to place the phone." "Prefer picture to AR because picture is still, easier to grab information."

Why *Picture* is sometimes preferred to AR techniques

Chapter 7

Summary and future work

7.1 Summary and Contributions

Identified the opportunities of the research

I started by identifying the opportunities of this research. AR systems have been proved to be beneficial in assisting professional operation tasks with an HMD. Hand-held AR is spreading with the rapid development of smartphones. But few AR techniques aim at assisting daily or normal operations. AR techniques have been facing acceptance problems for years and the issues need to be addressed by user studies.

Proposed the concept of Personalized AR Instructions with prototypes

I proposed the concept of using Personalized AR Instructions on mobile devices to assist people's operations in everyday lives. I illustrated the concept with use cases, scenarios and possible applications. Following a DIA process, the Keypad Application was developed as a paper and software prototype. The initial feedback about the usability of the proposed AR system was obtained and the technical performance for the prototype was tested.

Explored the interaction for authoring AR instructions

I designed another example application for a control panel to show the metaphor of note browsing and recording to personalize AR instructions. An unvalidated paper prototype was built to demonstrate how the interaction could be. As AR authoring was not the focus of this work, the discussion of this topic was closed with a list of possible approaches.

Prototyped an interaction technique with realtime feedback on AR overlay

Being inspired by the prototyping experience of the Keypad Application, the concept of providing real-time feedback on AR overlays was proposed as a solution for easing essential problems of AR techniques. I used an ad-hoc setup with a physical control surface, in which the status of the controls is visualized on the AR overly

Controlled experiment

Using this prototype, a controlled experiment was conducted to evaluate the benefits of real-time feedback on an AR overlay. The simulated interaction technique was compared with a conventional AR technique and two baselines - *Text* and *Picture* as the medium for providing operation instructions.

Significant results showed the benefits and issues

The statistical results revealed significant improvement of user performance for *AR+Feedback* technique compared to others. *AR* instructions had similar performance to *Picture*, while *Text* performed the worst. The subjective evaluation provided some reasons why conventional AR did not work well for the tasks. This gave us some insights about the usability and acceptance issues in AR techniques.

Summarize the contributions

As the major contribution, adding realtime feedback was proposed as a solution for essential problems of AR and its benefit was proved by solid experiment results. Although the result can not be applied to hand-free operations or AR systems with an HMD, they can be generalized to a wide range of applications. The reported issues provide valuable tips for designers and developers for developing AR systems. Furthermore, the concept of Personalized AR Instructions suggests a way for AR to go out of professional domains and play a role in helping people in daily tasks. In addition, a close relationship between AR and TUI was drawn, which led to the suggestion of using Ubicomp Computing techniques as complementary approaches to improve AR.

7.2 Future work

This work can be extended and refined in several aspects. I will present some possible extensions in the following.

7.2.1 Further Experiments

AR+Feedback is better than *Text* and *Picture* due to two factors

The experiment evaluated the benefits of feedback in AR systems by compared *AR+Feedback* and *AR*. However the improvement of *AR+Feedback* comparing to *Text* and *Picture* is caused by two factors: the feedback and the AR. The effect can not be attributed to any of them alone.

More experiment conditions could be valuable

More insights could be gained by adding more experiment conditions, for example adding feedback to *Text* and *Picture* or providing feedback on the physical surfaces with light or sound and so on. However, the simple design of the experiment is to control the time and prevent a fatigue effect. And the conducted experiment is sufficient to investigate the research questions of this work.

Experiment for Errors

As discussed in Section 6.3.3, Error Rate did not yield significant results, because not enough errors occurred in the experiment. Another experiment designed for error occurrence and recovery would be able to evaluate the performance of AR techniques for preventing and correcting users' mistakes.

Experiment with sequential tasks

The setting tasks in the conducted experiment were not order-dependent, so participants could operate in any order they wanted. In real life, many tasks require a fixed order to be performed, like entering a key code. I suspect that fixed-order tasks could be well-supported using *AR+Feedback*, because the AR layer could lead the user through the steps. This could be verified in a similar experiment to the one I conducted.

7.2.2 Improving Technical Performance

Technical improvement is valuable

There were minor performance issues in the prototypes that affected the usability. They were pointed out in the user feedback of Keypad Application (Section 3.3.3). The performance of the prototype was adequate for the experiment since the operations did not require high accuracy. Nevertheless an improvement is valuable for follow up studies.

Lag of graphical layer and resolution of camera image

The lag and slight shaking of the graphical layer while the phone is in motion should be reduced. Otherwise the alignment of virtual widgets and physical objects is not good enough for operations with higher accuracy. The resolution of camera image should be improved.

Use natural feature tracking instead of markers

Natural feature tracking can be used for the tracking of physical objects, instead of using fiducial markers. The fiducial markers take some space in the AR scene, thus could also cause inconvenience to users.

7.2.3 AR Notes Recording and Browsing

Further validation is needed for Control Panel Application

I presented the design of Control Panel Application with an unvalidated paper prototype in Section 3.4. An interaction of authoring AR Notes was proposed. It needs to be validated and refined. Certain technical challenges need to be overcome to implement a working prototype. They were not included since this work focused on other topics.

Possible conceptual extensions

There are many possible conceptual extensions to the note browsing and recording metaphor for AR instructions. For instance, the distance to the object and orientation of the phone could be used to visualize various levels of overview of the virtual notes. This research could possibly yield novel interactions that blend the physical and digital spaces.

Appendix A

Experiment Introduction for Participants

INTRODUCTION TO THE STUDY

Thank you for participating in our study. You will be asked to perform some setting tasks on the panel. The tasks are to reproduce settings according to the given instructions in different formats: TEXT, PICTURE, AR or INTERACTIVE-AR. Please do it as quickly as possible, without sacrificing the accuracy.

- Hold the phone as you like.
- You will be asked to reset the panel before each trial, please put your hand back to the designated place at this time.
- The order you manipulate the widgets does NOT matter, do it as you want.
- Try to think and find your own strategy for each technique.
- The optical button is the only button to press on the phone.
- When your setting is correct, you will be allowed to go to the next trial, otherwise you need to check and correct them until it is correct.
- There will be a practice session for all technique before the experiment, and short practice session before measurement for each technique.

Appendix B

Post Experiment Questionnaire

Post-experiment questionnaire

Subject n°

Gender: **Age:** **Profession:** **Descipline:**

Do you use mobile phone?

Never Sometimes Often Daily

Do you use smart phone?

Never Sometimes Often Daily

Do you use Augmented Reality software?

Never Sometimes Often Daily

Do you use appliance with physical widgets?

Never Sometimes Often Daily

What is the appliance:

Please rank the techniques in general for your preference.

Text	Picture	AR	AR+Feedback

why?

Please rank the techniques for each difficulty level for your preference.

	Text	Picture	AR	AR+Feedback
Difficulty: Easy				
Difficulty: Medium				
Difficulty: High				

Any opinion for each technique?

Text:

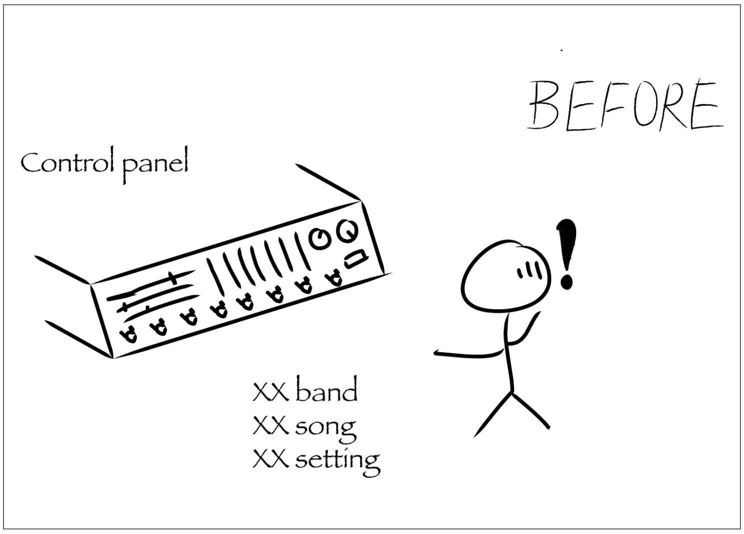
Picture:

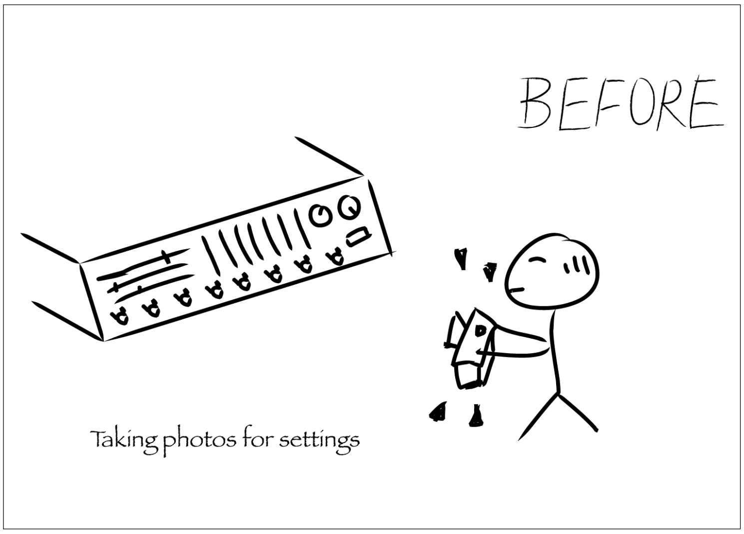
AR without feedback:

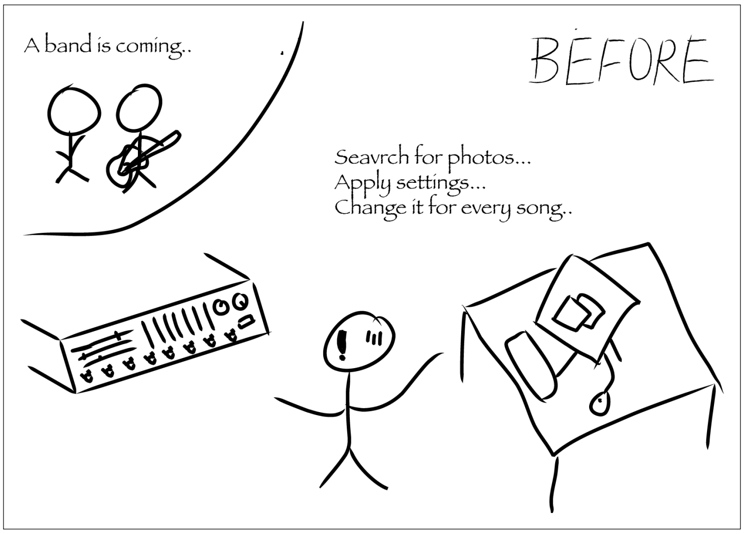
AR with feedback:

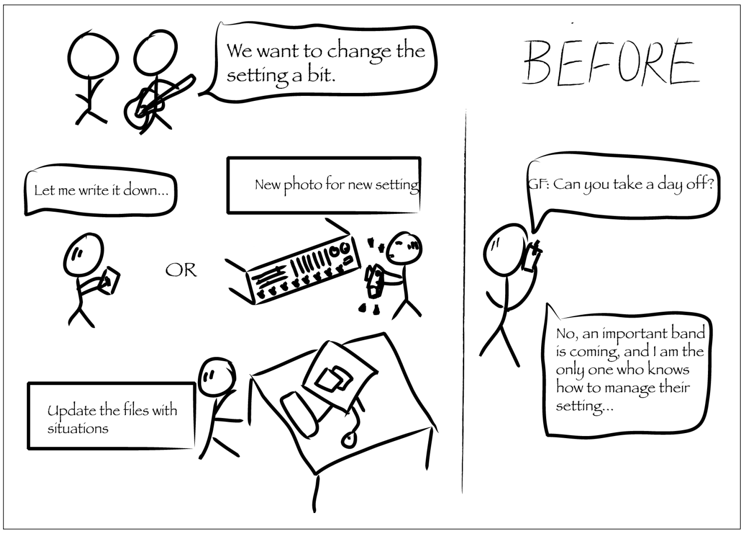
Appendix C

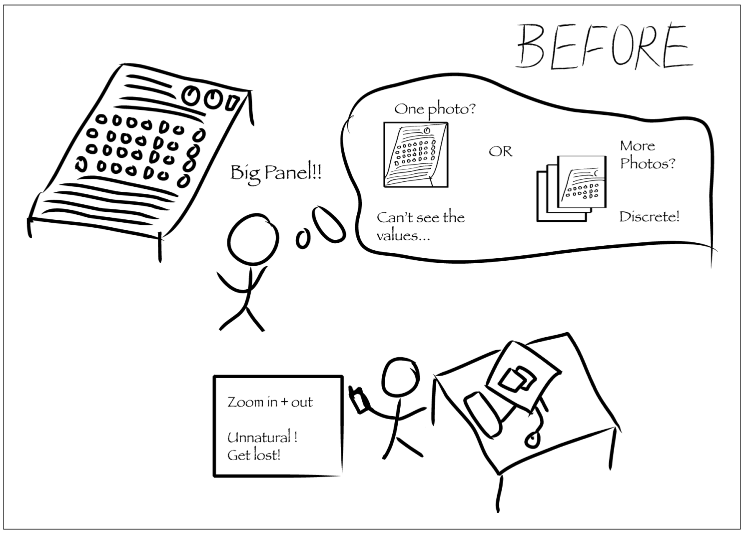
Storyboard of Guitar Amplifier Application

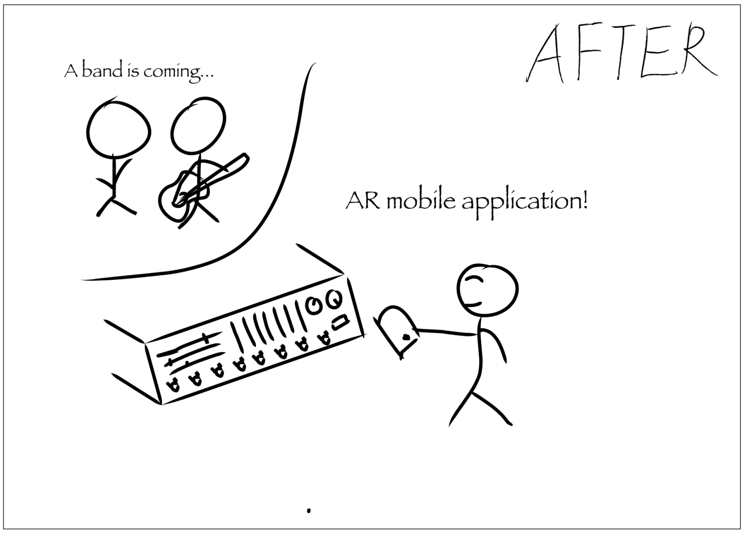

BEFORE
Control panel
XX band
XX song
XX setting

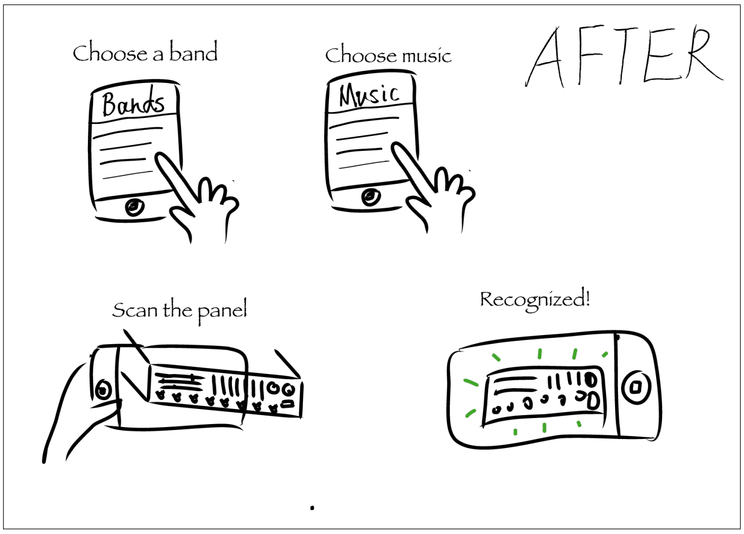

BEFORE
Taking photos for settings

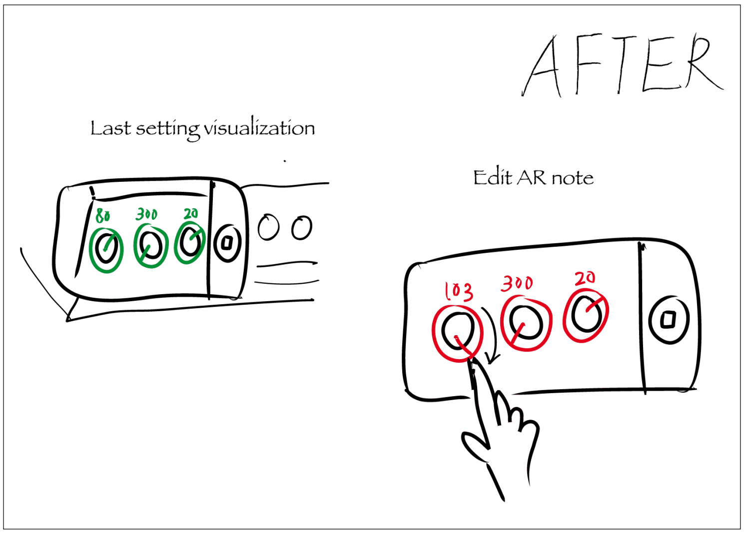

A band is coming..
BEFORE
Seavrch for photos...
Apply settings...
Change it for every song..

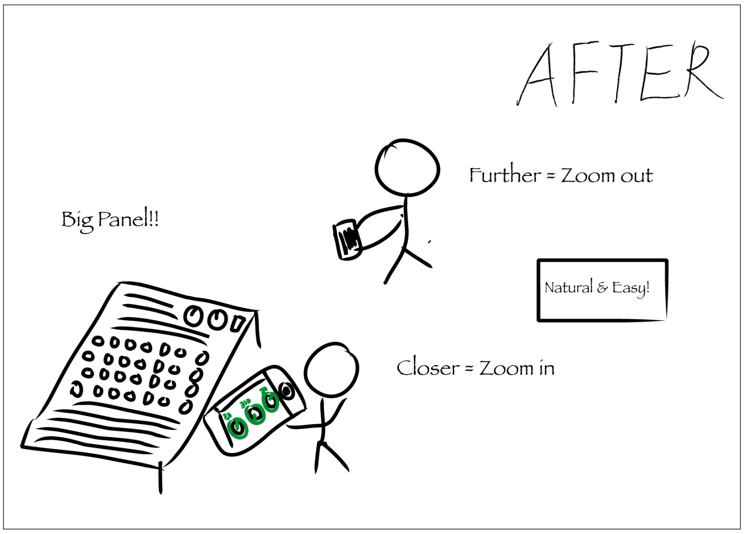

We want to change the setting a bit.
BEFORE
Let me write it down...
New photo for new setting
OR
Update the files with situations
GF: Can you take a day off?
No, an important band is coming, and I am the only one who knows how to manage their setting...

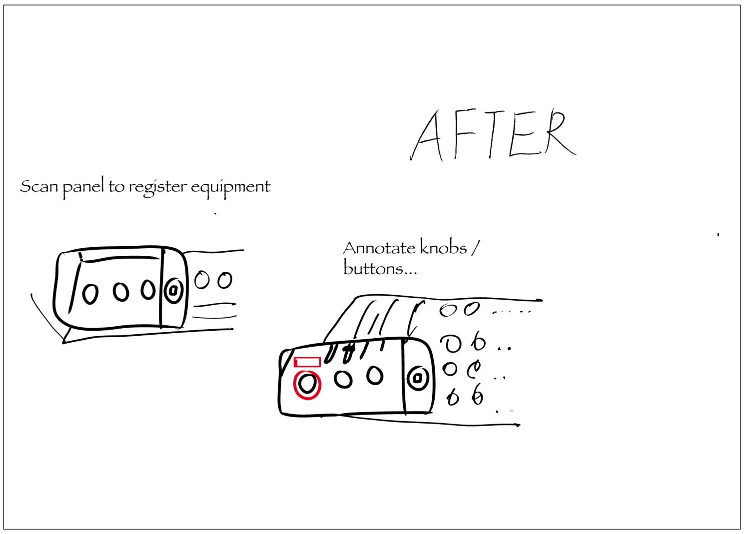

BEFORE
Big Panel!!
One photo?
Can't see the values...
OR
More Photos?
Discrete!
Zoom in + out
Unnatural ! Get lost!

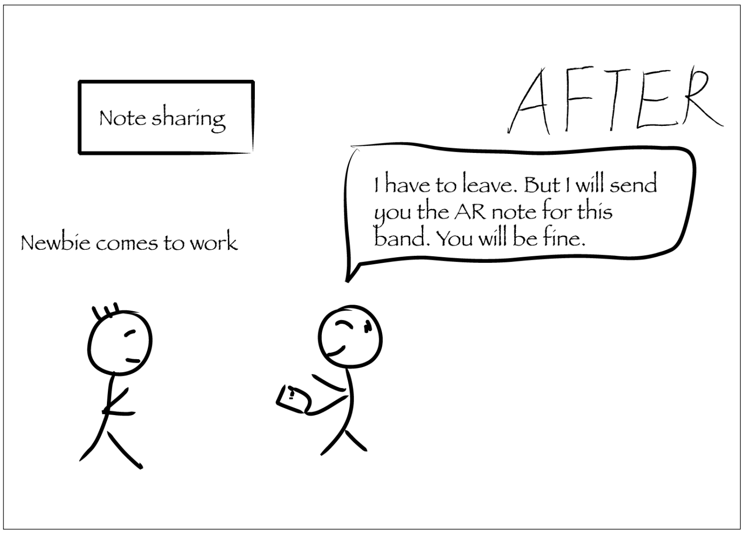

AFTER
A band is coming...
AR mobile application!


AFTER
Choose a band
Bands
Choose music
Music
Scan the panel
Recognized!


AFTER
Last setting visualization
80
300
20
Edit AR note
103
300
20


AFTER
Big Panel!!
Further = Zoom out
Natural & Easy!
Closer = Zoom in


AFTER
Scan panel to register equipment
Annotate knobs / buttons...


Note sharing
AFTER
Newbie comes to work
I have to leave. But I will send you the AR note for this band. You will be fine.

Bibliography

Ronald T Azuma. A survey of augmented reality. *Media*, 6 (4):355–385, 1997.

Olivier Bau and Wendy E. Mackay. Octopocus: a dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST '08, pages 37–46, New York, NY, USA, 2008.

Benjamin B. Bederson. Audio augmented reality: a prototype automated tour guide. In *Conference companion on Human factors in computing systems*, CHI '95, pages 210–211, New York, NY, USA, 1995.

Blaine Bell, Steven Feiner, and Tobias Höllerer. View management for virtual and augmented reality. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, UIST '01, pages 101–110, New York, NY, USA, 2001.

Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 73–80, New York, NY, USA, 1993.

M. Billinghurst, H. Kato, and I. Poupyrev. The magicbook - moving seamlessly between reality and virtuality. *Computer Graphics and Applications, IEEE*, 21:6–8, may/jun 2001.

Wolfgang Broll, Irma Lindt, Jan Ohlenburg, Iris Herbst, Michael Wittkamper, and Thomas Novotny. An infrastructure for realizing custom-tailored augmented reality user interfaces. *IEEE Transactions on Visualization and Computer Graphics*, 11:722–733, November 2005.

Julie Carmigniani, Borko Furht, Marco Anisetti, Paolo Ceravolo, Ernesto Damiani, and Misa Ivkovic. Augmented reality technologies, systems and applications. *Multimedia Tools Appl.*, 51:341–377, January 2011.

Thomas P Caudell and David W Mizell. *Augmented reality: an application of heads-up display technology to manual manufacturing processes*, volume ii, pages 659–669. 1992.

Deborah Estrin, David Culler, Kris Pister, and Gaurav Sukhatme. Connecting the physical world with pervasive networks. *IEEE Pervasive Computing*, 1:59–69, 2002.

S. Feiner, B. MacIntyre, T. Hollerer, and A. Webster. A touring machine: prototyping 3d mobile augmented reality systems for exploring the urban environment. In *Wearable Computers, 1997. Digest of Papers., First International Symposium on*, pages 74–81, oct 1997.

Steven Feiner, Blair Macintyre, and Dorée Seligmann. Knowledge-based augmented reality. *CACM*, 36:53–62, July 1993.

Zhou Feng, Henry Been-Lirn Duh, and Mark Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '08, pages 193–202, Washington, DC, USA, 2008.

E. Foxlin, Y. Altshuler, L. Naimark, and M. Harrington. Flighttracker: a novel optical/inertial tracker for cockpit enhanced vision. In *Mixed and Augmented Reality, 2004. ISMAR 2004. Third IEEE and ACM International Symposium on*, pages 212–221, nov. 2004.

David Frohlich and Rachel Murphy. The memory box. *Personal Technologies*, 4:238–240, 2000.

Christian Geiger, Volker Paelke, and Christian Reimann. Mobile entertainment computing. In Stefan Göbel, Ulrike Spierling, Anja Hoffmann, Ido Iurgel, Oliver Schneider, Johanna Dechau, and Axel Feix, editors, *Technologies for Interactive Digital Storytelling and Entertainment*, volume

3105 of *Lecture Notes in Computer Science*, pages 142–147. Springer Berlin / Heidelberg, 2004.

Jun Gong and Peter Tarasewich. Guidelines for handheld mobile device interface design. In *Proceedings of the 2004 DSI Annual Meeting*, 2004.

R. Grasset, A. Dunser, and M. Billinghurst. The design of a mixed-reality book: Is it still a real book? In *Mixed and Augmented Reality, 2008. ISMAR 2008. 7th IEEE/ACM International Symposium on*, pages 99–102, sept. 2008.

F. E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11:1–21, 1969.

S. Guven, S. Feiner, and O. Oda. Mobile augmented reality interaction techniques for authoring situated media on-site. In *Mixed and Augmented Reality, 2006. ISMAR 2006. IEEE/ACM International Symposium on*, pages 235–236, Oct. 2006.

Mika Hakkarainen, Mark Billinghurst, and Charles Woodward. Augmented assembly using a mobile phone. In *MUM '08*, pages 84–87, New York, NY, USA, 2008.

Florian Heller and Jan Borchers. Corona: Audio augmented reality in historic sites. In Marco de Sa, Elizabeth F. Churchill, and Katherine Isbister, editors, *MobileHCI 2011 Workshop on Mobile Augmented Reality: Design Issues and Opportunities*, pages 51–54, Stockholm, Sweden, August 2011.

Steven J. Henderson and Steven Feiner. Opportunistic controls: leveraging natural affordances as tangible user interfaces for augmented reality. In *VRST '08*, pages 211–218, New York, NY, USA, 2008.

Steven J. Henderson and Steven Feiner. Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret. In *ISMAR '09*, pages 135–144, Washington, DC, USA, 2009.

Anders Henrysson, Mark Billinghurst, and Mark Ollila. Face to face collaborative ar on mobile phones. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '05, pages 80–89, Washington, DC, USA, 2005.

A. Kotranza, D. Scott Lind, C.M. Pugh, and B. Lok. Real-time in-situ visual feedback of task performance in mixed environments for learning joint psychomotor-cognitive tasks. In *ISMAR '09*, pages 125 –134, 2009.

Wendy E. Mackay, Guillaume Pothier, Catherine Letondal, Kaare Bøegh, and Hans Erik Sørensen. The missing link: augmenting biology laboratory notebooks. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*, UIST '02, pages 41–50, New York, NY, USA, 2002.

Wendy E. Mackay, Caroline Appert, Michel Beaudouin-Lafon, Olivier Chapuis, Yangzhou Du, Jean-Daniel Fekete, and Yves Guiard. Touchstone: exploratory design of experiments. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07, pages 1425–1434, 2007.

Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. In *IEICE Trans. Information Systems*, volume E77-D, pages 1321–1329, 1994.

Pranav Mistry, Tsuyoshi Kuroki, and Chaochi Chang. Tapuma: tangible public map for information acquirement through the things we carry. In *Proceedings of the 1st international conference on Ambient media and systems*, Ambi-Sys '08, pages 12:1–12:5, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Pranav Mistry, Pattie Maes, and Liyan Chang. Wuw - wear ur world: a wearable gestural interface. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, CHI EA '09, pages 4111–4116, New York, NY, USA, 2009.

T. Miyashita, P. Meier, T. Tachikawa, S. Orlic, T. Eble, V. Scholz, A. Gapel, O. Gerl, S. Arnaudov, and S. Lieberknecht. An augmented reality museum guide. In *Mixed and Augmented Reality, 2008. ISMAR 2008. 7th IEEE/ACM International Symposium on*, pages 103–106, Sept. 2008.

Ann Morrison, Antti Oulasvirta, Peter Peltonen, Saija Lemmela, Giulio Jacucci, Gerhard Reitmayr, Jaana Näsänen, and Antti Juustila. Like bees around the hive: a comparative study of a mobile augmented reality map. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pages 1889–1898, New York, NY, USA, 2009.

Elena Mugellini, Elisa Rubegni, Sandro Gerardi, and Omar Abou Khaled. Using personal objects as tangible interfaces for memory recollection and sharing. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, TEI '07, pages 231–238, New York, NY, USA, 2007.

Leonid Naimark and Eric Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, ISMAR '02, Washington, DC, USA, 2002.

J. Newman, D. Ingram, and A. Hopper. Augmented reality in a wide area sentient environment. In *Augmented Reality, 2001. Proceedings. IEEE and ACM International Symposium on*, pages 77 –86, 2001.

Emmanuel Pietriga, Stéphane Huot, Mathieu Nancel, and Romain Primet. Rapid Development of User Interfaces on Cluster-Driven Wall Displays with jBricks. In *EICS '11: 2nd ACM SIGCHI symposium on Engineering interactive computing systems*, Pisa, Italy, June 2011.

J. Quarles, S. Lampotang, I. Fischler, P. Fishwick, and B. Lok. A mixed reality approach for merging abstract and concrete knowledge. In *VR '08*, pages 27 –34, 2008.

Ramesh Raskar, Jeroen van Baar, Paul Beardsley, Thomas Willwacher, Srinivas Rao, and Clifton Forlines. ilamps: geometrically aware and self-configuring projectors. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, New York, NY, USA, 2005.

Gerhard Reitmayr and Tom Drummond. Going out: robust model-based tracking for outdoor augmented reality. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '06, pages 109–118, Washington, DC, USA, 2006.

Michael Rohs, Johannes Schöning, Martin Raubal, Georg Essl, and Antonio Krüger. Map navigation with mobile devices: virtual versus physical movement with and without visual context. In *Proceedings of the 9th international conference on Multimodal interfaces*, ICMI '07, pages 146–153, New York, NY, USA, 2007.

Daniela Rosner and Kimiko Ryokai. Weaving memories into handcrafted artifacts with spyn. In *CHI '08 extended abstracts on Human factors in computing systems*, CHI EA '08, pages 2331–2336, New York, NY, USA, 2008.

Christian Sandor, Alex Olwal, Blaine Bell, and Steven Feiner. Immersive mixed-reality configuration of hybrid user interfaces. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '05, pages 110–113, Washington, DC, USA, 2005.

D. Schmalstieg, A. Fuhrmann, and G. Hesina. Bridging multiple user interface dimensions with augmented reality. In *Augmented Reality, 2000. (ISAR 2000). Proceedings. IEEE and ACM International Symposium on*, pages 20 –29, 2000.

Ben Shneiderman. *Designing the user interface: strategies for effective human-computer interaction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1986.

McKay Moore Sohlberg and Catherine A. Mateer. *Introduction to Cognitive Rehabilitation: Theory and Practice.* Guilford Press, 1989.

Ivan E. Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, AFIPS '68 (Fall, part I), pages 757–764, New York, NY, USA, 1968.

Arthur Tang, Charles Owen, Frank Biocca, and Weimin Mou. Comparative effectiveness of augmented reality in object assembly. In *CHI '03*, pages 73–80, New York, NY, USA, 2003.

Ryuhei Tenmoku, Masayuki Kanbara, and Naokazu Yokoya. Annotating user-viewed objects for wearable ar systems. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '05, pages 192–193, Washington, DC, USA, 2005.

D. Wagner and D. Schmalstieg. First steps towards handheld augmented reality. In *Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on*, pages 127–135, Oct. 2003.

Roy Want, Kenneth P. Fishkin, Anuj Gujar, and Beverly L. Harrison. Bridging physical and virtual worlds with electronic tags. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, CHI '99, pages 370–377, New York, NY, USA, 1999.

Mark Weiser. The computer for the 21st century. *Scientific American*, 265(3):66–75, 1991.

Malte Weiss, Julie Wagner, Yvonne Jansen, Roger Jennings, Ramsin Khoshabeh, James D. Hollan, and Jan Borchers. Slap widgets: bridging the gap between virtual and physical controls on tabletops. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pages 481–490, New York, NY, USA, 2009.

S. A. G. Wensveen, J. P. Djajadiningrat, and C. J. Overbeeke. Interaction frogger: a design framework to couple action and function through feedback and feedforward. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques*, DIS '04, pages 177–184, New York, NY, USA, 2004.

Sean White and Steven Feiner. Sitelens: situated visualization techniques for urban site visits. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pages 1117–1120, New York, NY, USA, 2009.

Ron Yeh, Chunyuan Liao, Scott Klemmer, François Guimbretière, Brian Lee, Boyko Kakaradov, Jeannie Stamberger, and Andreas Paepcke. Butterflynet: a mobile capture and access system for field biology research. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 571–580, New York, NY, USA, 2006.

J. Zauner, M. Haller, A. Brandl, and W. Hartman. Authoring of a mixed reality assembly instructor for hierarchical structures. In *Mixed and Augmented Reality, 2003. Proceedings. The Second IEEE and ACM International Symposium on*, pages 237–246, Oct. 2003.

Xiang Zhang, Stephan Fronz, and Nassir Navab. Visual marker detection and decoding in ar systems: A comparative study. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, ISMAR '02, Washington, DC, USA, 2002.

Index

Typeset January 14, 2012