An AR Document Collaboration System Combining Physical and Digital Content With Gesture Interactions

Ruichao ZHENG
44191597 - 6

Master of Engineering

Supervisor: Prof. Jiro TANAKA
Graduate School of Information, Production and Systems
Waseda University

July 2021

Abstract

Nowadays, the demand of teleworking is growing. Though working on digital documents together is much easier than before, there are still many people who want to use paper documents while collaborating with others for paper documents have many positive attributes. While at the same time, when documents are presented on digital devices, some unique functions like sharing and instant communication can be provided to users. To take advantages of both paper documents and digital documents, this paper explores the combination of physical and digital documents by using Augmented Reality.

This research first clarifies the goal, which is to implement a system that allows users to collaborate with each other between documents in digital and physical world. This goal can be divided into three points, which are to allow users to exchange comments on the documents they are reading, allow users to use pen strokes and hand actions to communicate with each other, and allow users to share contents that they are interested in with others. Then, in order to achieve these goal, four functions are proposed, which are Comment Sharing, Strokes Sharing, Hand Shape Sharing, and Board Sharing, and the design of these functions are described in detail.

Technically, this system uses Marker Recognition to recognize and track paper document and support comment capturing for Comment Sharing. Color Extracting is used to extract and rebuild users' pen strokes for Strokes Sharing. It is also used together with Template Matching to detect the cropped region for Board Sharing. By getting hand joint information from depth camera, a hand model is rebuilt and so that hand shape can be shared with other users in Hand Shape Sharing.

Overall, this system successfully accomplishes the goal. By Strokes Sharing function and Hand Shape Sharing function, this system can share users' pen strokes and hand movement in realtime, which allow users to communicate with each other by pen strokes and hand gestures and express themselves in a more freely and unlimited way. Together with Comment Sharing and Board Sharing, these functions enhance remote collaboration in some way.

Keywords: Augmented reality; Gesture interaction; Collaboration

Acknowledgements

Thanks to my supervisor Professor TANAKA Jiro for the instruction and time spent on me. He gave me a lot of help and was patient when guide me to solve problems.

Thanks to my friends in IPLab. They are very warm-hearted and gave me lots of constructive suggestions when my research hit a snag. The days spent with them are precious treasures in my memory, I will never forget the friendship with them.

Thanks to my parents. Their mental support and warm encouragement is always the inexhaustible motivation in my heart.

At last, I also want to thank myself for not giving up in the hardest time and chose to stick to the end.

Contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Introduction

Today, the demand of teleworking is growing. Though working on digital documents together is much easier than before, there are still many people who want to use paper documents while collaborating with others, for paper documents have many positive attributes, for example, natural tangibility and inherent physical flexibility, as discovered by prior research [29] [30]. At the same time, when documents are presented on digital devices, some unique functions like sharing and instant communication can be provided to users. As a result, bringing the best of physical and digital worlds together is regarded to be the next step in a technological setup [16].

To combine physical document and digital document together, Augmented Reality is a commonly used approach. Augmented Reality brings digital information and virtual objects into the physical space. With the help of Augmented Reality technology, information about the real world around becomes interactive and digital. Augmented Reality system has three prominent features [1]:

- Combines real and virtual objects in a real environment.
- Runs interactively, and in real time.
- Registers (aligns) real and virtual objects with each other.

This paper explores the combination of physical and digital documents by using Augmented Reality. In this work, a system that allows users to collaborate with each other between documents in digital and physical world is designed and implemented. This system tracks the location of physical documents, as well as user's gestures and strokes, to augment the paper documents. Users are then able to interact with system, and users' interaction data will be transmitted through network. By using functions provided by this system, users can share comments, pen strokes, hand movement and highlight contents on document with each other to achieve collaboration.

1.2 Organization of the Thesis

The rest part of this thesis is organized as follows: Chapter 2 will present the research goal and briefly outline the approach of this research. Chapter 3 will review some related work. Chapter 4 will present the system design part, where the design concept and method of each function will be introduced in detail. Chapter 5 will present the system implementation part where the detailed devices, environment and implementation will be introduced. Chapter 6 will conclude the previous content and talk about the future possibilities.

Chapter 2

Research Goal and Approach

2.1 Research Goal

The fundamental goal of this research is to propose and implement a system that allows users to collaborate with each other between documents in digital and physical world. This goal is analyzed in the following paragraphs.

Document collaboration is often inseparable from discussion, and annotating is usually required during the discussion. Marshall's extensive study [24] of annotated materials reveals that annotations provide a permanent frame of reference for guiding the reader's visual attention. According to Marshall, the major functional roles of annotations include procedural signaling for future attention, place marking and aiding memory, in situ locations for problem solving, and visible traces of reader's attention. Matthew Hong et al. [14] conducted a microanalysis of active reading behavior and also observed annotating practices that fit these descriptions. In their research, the notes and comments were written in relation to the spatial location of a specific paragraph, line, or word of interest, making manual search and navigation less difficult. Although their experiments were conducted in a single-person scenario, notes still play a similar role in a multi-person scenario. At the same time, notes also allow participants to exchange opinions with each other to reach a consensus among them under collaboration scenario. Thus, notes allowing users to exchange opinions can be one point of the goal.

Other forms of annotations, including underlining, circling and enclosing, also increase the saliency of particular areas on the document. When people are discussing paper documents face-to-face, they often use a pen to make some freeform annotations, like underlining and circling, on the paper documents to highlight specific elements, or use hand motions to assist them in expressing their ideas. The affordance of paper allows the participants to perform a natural interaction with document when collaborating with others, and they also benefit from the availability of this kind of non-verbal communication [5]. When documents are presented on digital screens, however, this kind of interaction is limited. Though the annotations user makes on digital documents can be synchronized to other users' view in a short time, they can no longer make annotations as freely as they do on paper documents, neither can they use hand motions to assist them in expressing their ideas, which will affect the efficiency of collaboration. Therefore, being able to communicate with each other in a natural way is another point of the goal, and this point plays an important role in achieving the goal well.

While underlining and note is often used for annotation in a small range, enclosing like drawing "⌈" and "⌋" around contents can be used to highlight large elements on the document such as an image or a paragraph. This kind of contents are usually important parts that require future attention and need to be organized in a proper way so that people can have frequent reference to these parts. Sharing these contents with other participants can let them pay more attention on these parts and therefore be helpful for solving the problem when collaborating with each other. Consequently, sharing highlight contents is also a point of the goal.

2.2 Research Approach

This research focuses on using augmented reality to connect digital world and real world. Based on the analysis mentioned in Section 2.1, in order to achieve the goal, the following requirements are imposed to this system.

- This system should allow users to exchange their comments about the documents they are reading.

- This system should allow users to use pen strokes and hand actions to communicate with each other.
- This system should allow users to share contents that they are interested in on the document with others.

To meet these requirements, four main functions are implemented in this system.

First, in order to allow users to exchange opinions on the documents they are reading, Comment Sharing function is implemented to let users write comments about specific elements on the documents, show comments from different users and allow users to reply comments from others.

Second, in order to allow users to use pen strokes to communicate with each other, Strokes Sharing function is implemented to capture and extract users' pen strokes and transmit them to other users' view in realtime.

Third, in order to allow users to use hand actions to communicate with each other, Hand Shape Sharing function is implemented to rebuild hand models according to the position of users' hand joints and show users' hand movement in realtime.

Last, in order to allow users to share highlight parts with others, Board Sharing function is implemented so that users can crop and save the contents that they are interested in from the physical documents they are reading by pen and choose whether to share these contents with others.

Chapter 3

Related Work

This chapter introduces related work about the proposed system. This research focuses on using augmented reality to combine the digital world and physical world and implement a document collaboration system. Therefore, related work can be divided into three kinds. The first kind of related work is work about combining physical and digital contents. The second kind of related work is work using augmented reality to combine physical and digital contents. The third kind of related work is work of collaboration system combining digital and physical contents.

3.1 Combining Digital and Physical Contents

The research that attempts to bridge the gap between paper and the digital world can be traced back to pioneering systems like DigitalDesk [32]. The approach of this work is to augment paper with video projection and other digital facilities. It projects electronic images down onto the desk and onto paper documents directly from overhead. By using camera, it responds to interaction with pens or bare fingers and can read paper documents placed on the desk. Also, EnhancedDesk [19] uses an additional pan-and-tilt infrared camera and printed matrix codes to enable better tracking of users' hand gestures, and provides various interactions to augment a textbook.

Some researches try to provide a “magic-lens” interface to visualize additional information about a physical document. For example, PACER [23] supports fine-grained paper document content manipulation through a smartphone with a touch screen and camera. Users can augment a printed document through the smartphone’s screen in order to perform actions like keyword search, copy and email part of the document, etc.

In order to visually extend paper, interactive desk surface have also been used to augment the spaces surrounding physical documents. For example, DocuDesk [8] augments physical paper with shadow menus that are displayed around the paper on a tabletop computer, which provides rich paper-associated feedback. User can use the tabletop computer to explore the links between the paper and digital documents and further manipulate the relationships between documents by using stylus.

Sketching hardware have the potential of providing enhancing sketching and note taking capabilities. For example, NiCEBook [4] focuses on supporting more structured and efficient note-taking and presents new architectures to link captured notes across applications and devices. It combines the flexibility and simplicity of taking notes on paper with the benefits of a digital representation. The resulting structured notes are accessible through an application that offers different views on the notebook.

Another research on sketching interfaces is IllumiPaper [18], which proposes a UI framework and prototype for interactive paper, focusing on visual feedback position, feedback time, and feedback types. It investigates new forms of paper-integrated feedback, which build on emerging paper-based electronics and novel thin-film display technologies. The approach of this research focuses on illuminated elements, which are seamlessly integrated into standard paper.

3.2 Combining Digital and Physical Contents by Augmented Reality

Works using augmented reality to combine physical and digital contents can be classified into three kinds. The first kind of these works is projection based augmented reality. The second kind of these works is handheld display and head mounted display based augmented reality. The third kind of these works is smartphone based augmented reality.

3.2.1 Projection Based Augmented Reality

Dachselt and Al-Saiegh point to many different attempts and examples in finding solutions for linking printed books and digital content. They suggest the concept of Projective Augmented Books (PAB) [7], which does not require fixed locations and expensive hardware and can be used almost everywhere and also combined well-known pen interaction as a natural mode of interaction with a visual augmentation of real surfaces. The Projective Augmented Book explores projection onto a physical printed book in the style of a reading lamp. A digital pen captures gestures with specially printed paper and allows virtual annotations and projected traces. In addition to enabling pen gestures within a physical book, PAB used a paper GUI palette to support mode switching. Using a pen gesture or via selecting a function from a palette, the user can perform various tasks on text such as dictionary look up, add to scrapbook, project annotations onto page and email. Any feedback is then projected onto the book.

A system called AR Lamp proposed by Kim et al. [17] is an education support system combining the projection-based AR visualization and physical objects such as a paper and a pen. The system projects a virtual image on a physical paper and a user can interactively manipulate virtual objects in the scene using a pen. By using a new interaction method with a pico-projector as an AR system to enhance the digital display on static paper, teachers, students, and experts commented that AR was useful for learning.

The possibilities of interaction by bending multiple paper-like devices are explored in PaperWindows by Holman et al. [13] where they present a system that would represent documents in an office setting. They propose using paper as an input to track the movement and shape of the paper through the Vicon motion capture system to obtain a complete set of gestural interactions for PaperWindows and show their effects on webpages projected onto paper. In this work, interaction styles suitable for digitally augmented paper in the 3D space are introduced and eight interaction styles based on the natural manipulation of paper are defined.

3.2.2 Handheld Display and Head Mounted Display Based Augmented Reality

VR and AR handheld displays and head mounted displays have also been explored for interacting with physical documents.

The Magic Book [3] is an enhanced version of a traditional pop-up book that allows users to scan AR markers on the pages to see 3D virtual models. Users can quickly switch to an immersive mode and enter the 3D model. The book is experienced through a handheld AR display, so users do not manipulate the book continuously, but rather lay it down on a table and look at it through the AR display. While inside the 3D model, the physical book object is not needed. The book is meant as a collaborative interface that allows quick transition between immersive VR and AR modes.

Grasset et al. propose The Mixed Reality Book [12] and symbiotically merge traditional text and digital media such as static or dynamic illustration, sounds. This work presents a new type of digital book which takes a traditional book and enhances it visually and aurally, while effectively combining the physical and digital world. It provides a successful integration of a mixed reality book which presents multisensory content by keeping the physical book. The user can see a live video feed of the scene and experience the spatial sound changing according to their position and the actual content on the page. This work created an enhanced immersive experience mainly focus on spatially visual-aural augmentation with a seamless approach.

HoloDoc by Li et al. [22] explores combining physical and digital papers in document-intensive tasks. HoloDoc is a mixed reality system based on Microsoft HoloLens that allows users to make use of physical documents, while preserving the advantages of digital content, such as the ability to search, hyperlink, and access rich media. It tracks the location of physical artifacts, as well as a user's gestures and strokes, to augment the artifacts with multimedia content in mixed reality. Users are then not limited by the field of view of a camera or projector and able to directly interact with content in 3D space to take notes and look up words. This work shows that working with a combination of physical and digital documents through AR is feasible and accepted by users.

Reipschlager and Dachselt combine an interactive surface with an AR HMD for a stereoscopic 3D modeling tool and propose DesignAR [27], which demonstrates the use of a Microsoft HoloLens with an interactive surface for the creation of 3D models. In this work, there is a strong connection between the display and the AR content and a particular focus is on how the borders of the display can be utilized to place additional views and how to interact with them. They defined three levels of spatial proximity that relate to design space, and the way information is positioned in AR can rely on physical properties, such as the position, orientation, or size of a mobile device as well as the number and location of users, which allows DesignAR nicely show how such user interface components can be attached around a conventional display using AR. They have also coined the term Augmented Displays for seamlessly combining interactive displays with head mounted AR.

3.2.3 Smartphone Based Augmented Reality

Y. Li et al. present ARCritique [21], a mobile augmented reality application running on smartphone that combines KinectFusion and ARKit. It allows users to scan artifacts and share the resulting 3D models, view the model simultaneously in a shared virtual environment from remote physical locations, and point to and draw on the model to aid communication.

Daiki Yamaji et al. [34] propose a system for users of general paper media, such as newspaper, books, publications, etc., which uses recognition of written cues made by handwritten entries and performs digital processing. Users are able to use this system by a smartphone and on paper-media to save favorite paragraphs or images on the paper, illustrate data associations, and search for English translations, all the while being able to use the paper-media in a natural way. Moreover, users are able to browse the interaction results from both the paper-media and smartphone.

3.3 Collaboration System Combining Digital and Physical Contents

Mads Møller Jensen et al. [16] compare the use of physical and digital sticky notes in collaborative ideation, and find that the next step in a technological setup is not an either or, but should bring the best of physical and digital worlds together. Also, X. Wang et al. [31] executed experiments to test the capabilities of MR systems in realistic environment and collaborative tasks against prevalent methods. Results indicated that MR systems significantly reduced the performance time for the collaborative design error detection task.

Danakorn Nincarean Eh Phon et al. [25] review the literatures concerning collaborative augmented reality, its previous usages and its potential. Damien Clergeaud et al. [6] examined VR functionality in relation to shared physical and virtual workspaces.

Susan R. Fussell et al. [9] demonstrate the value of a shared view of the work environment for remote collaboration on physical tasks. Jeffrey W. Chastine et al. [5] also find, for collaborative environments to be successful, a fundamental requirement is that these environments provide support for interreferential awareness or the ability for one participant to refer to a set of objects, and for that reference to be understood by others.

Andreea-Carmen Ifrim et al. [15] present the concept of LibrARy, an AR system that intends to create innovative collaborative environments where multiple users can interact simultaneously with computer-generated content. Joan Sol Roo and Martin Hachet [28] present an integrated Mixed Reality ecosystem. In this system, multiple observer can use the same through-the-lens image, which reinforces collaborative and social interaction. Other systems that capture local scene and wirelessly stream to remote side for the helper to view have been explored [20] [10]. These studies have confirmed Jeffrey W. Chastine's view.

Also, Bacca et al. [2] point out that in the recent years there has been an increased interest for researching the benefits of AR in the educational context.

There are also some studies about asymmetrical interaction and collaboration using immersive technologies which provide insights and design considerations in regard to aspects such as co-presence, awareness, leadership roles and task contribution, communication and collaboration, and co-manipulation of objects [6] [11] [26] [33].

Chapter 4

System Design

This chapter introduces the design of this system and explains each part. At first, the system overview and how the system works will be introduced. Then, each part of this system will be explained, followed by a usage scenario, which shows how these functions can be used together and be helpful to collaboration.

4.1 System Overview

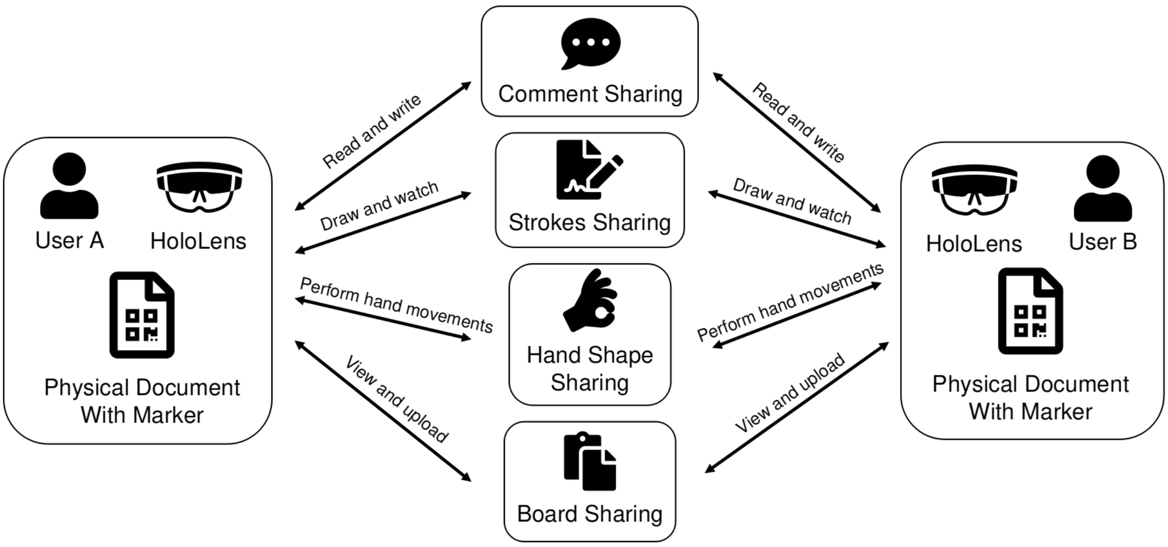


Fig. 4.1 System overview

Fig. 4.1 shows the overall structure of this system.

There are multiple users in this system, for example, User A and User B. Each of the user wears a head mounted device, which is Microsoft HoloLens 2 in this system. In front of users are paper documents. A marker is printed to each paper document so that the document can be recognized and the movement of the document can be tracked by the system. Users' interaction information will be transmitted through network. Four main functions are provided for users to interact with this system, which are Comment Sharing, Strokes Sharing, Hand Shape Sharing, and Board Sharing. Each of the four main functions will be explained in detailed in the following sections. Even though two-user case is took as an example in this figure, this system can be extended to cases with three or more users, and each user in this system is equivalent.

4.2 Comment Sharing

Before using this system, it is necessary to scan the marker printed on the physical document, so that the document can be recognized and the movement of the document can be tracked by the system. After the system recognize the document, user can see existing comments shown as virtual objects.

User can add comments by writing on note, which is a small piece of paper with marker printed besides comment part. To add a comment, user first needs to put the note to the place where he or she want to comment about. Then, users can write down the comment on the comment part of the note. Last, users need to uncover the marker on the note, so that the system can recognize the marker. After recognizing the marker on the note, the system will use camera provided by head mounted device to capture the comment and upload the comment so that each user can see this comment. User can also reply other users' comments in the same way. Above the comment, user name with different color is used to distinguish between users.

In this way, users are able to exchange ideas with each other.

For example, suppose Alice and Bob are reading the same document in different place at the same time. As Fig. 4.2 shows, at the very beginning, there is no comment on the

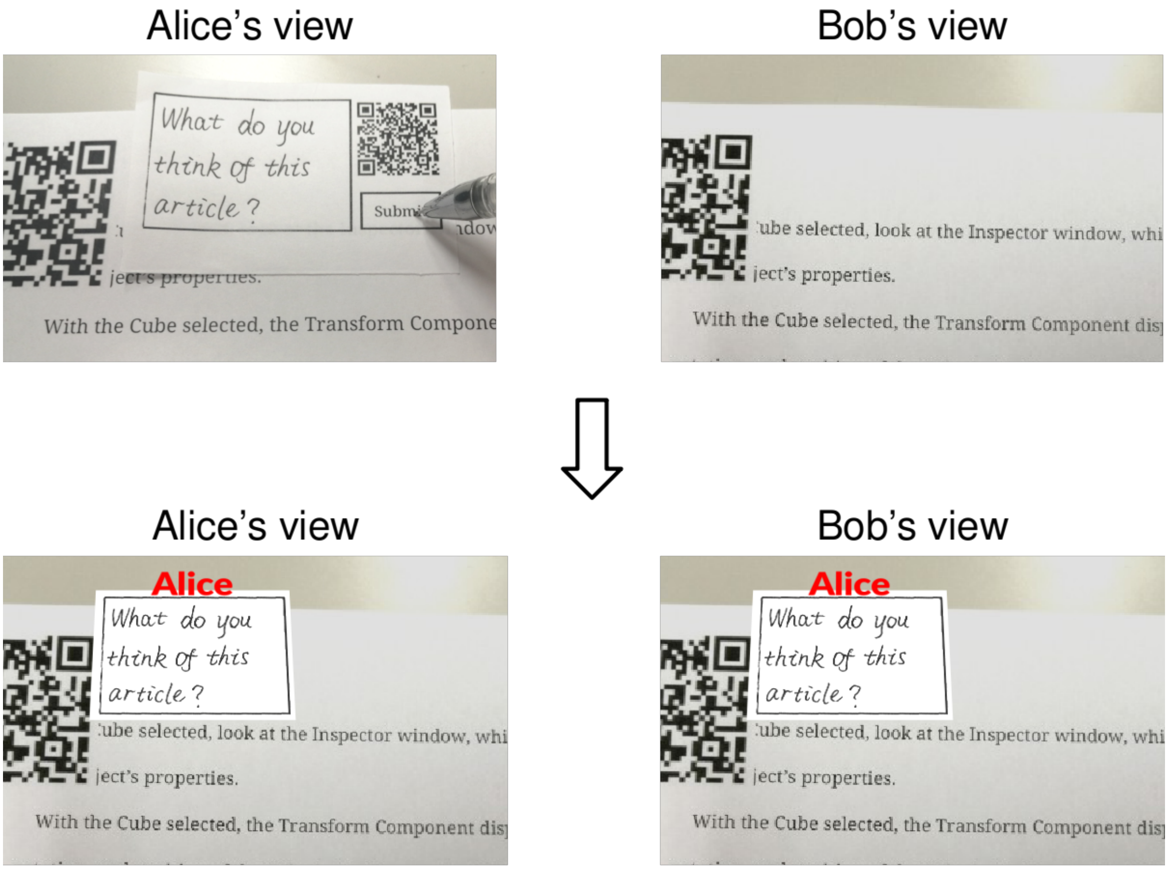


Fig. 4.2 Alice adds a comment, saying “What do you think of this article?”, to the document

document. Then Alice can add a comment, saying "What do you think of this article?", to the document and upload this comment. After this, both Alice and Bob will be able to see this comment, and Alice's name will appear at the top of the comment she just uploaded in a specific color (here it is red).

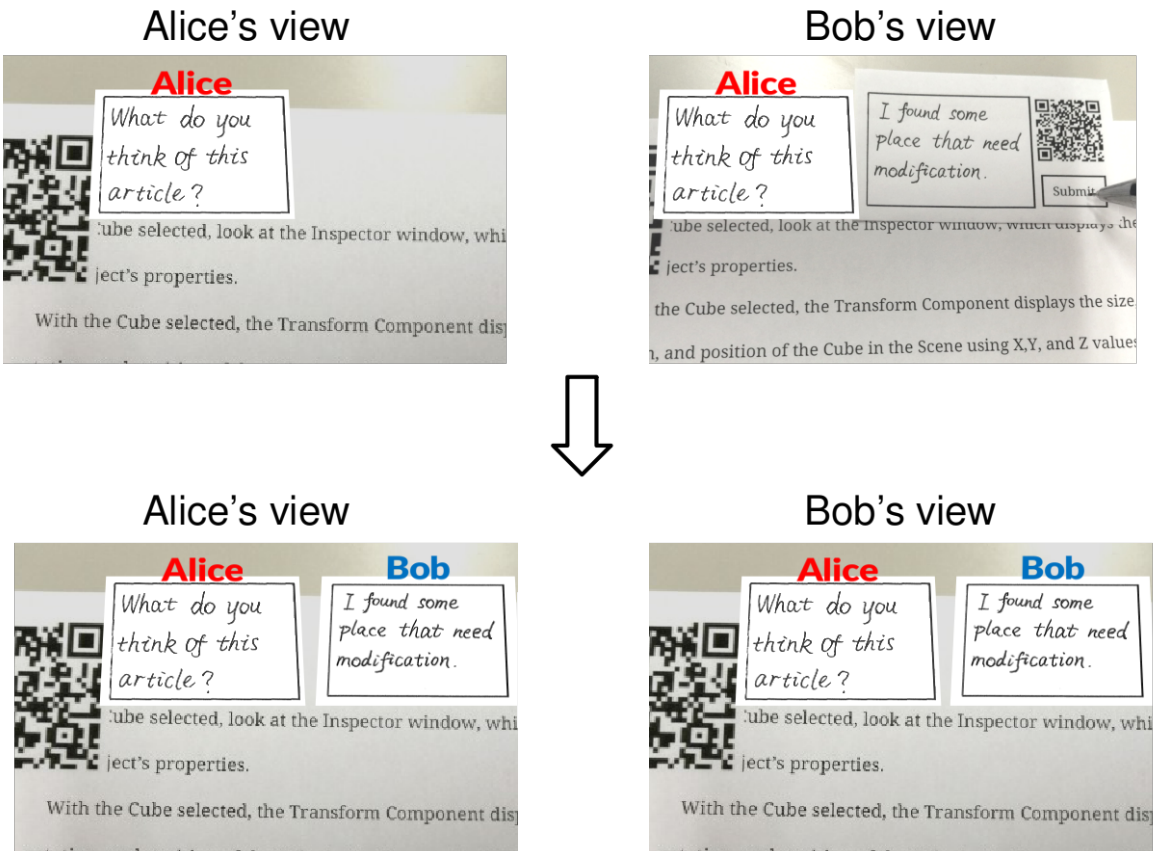


Fig. 4.3 Seeing Alice's comment, Bob can reply to it, saying "I found some place that need modification."

As Fig. 4.3 shows, seeing Alice's comment, Bob can choose to reply to Alice's comment. So he adds a comment just besides Alice's comment, saying "I found some place that need modification.", to reply to Alice. After this, both Alice and Bob will be able to see this comment, and Bob's name will appear at the top of the comment he just uploaded in another specific color (here it is blue).

4.3 Strokes Sharing

When one user draws pen strokes, for instance, a circle or an underline, on the document, the pen strokes can be synchronized to other users' view.

To share pen strokes, user needs to choose a color as pen stroke color for the system to track. When users are sharing their pen strokes, the system will use camera provided by head mounted device to capture the strokes every short period of time and upload the strokes to update every user's view. Pen strokes are rendered in different colors in order to distinguish between different users.

Since the users can draw pen strokes in any way as they like, this function allows users to express themselves in a more freely and unlimited way. The position of pen nib in the end of pen stroke flow can also serve as a temporary visual focal point, suggesting users to pay more attention and focus on this point.

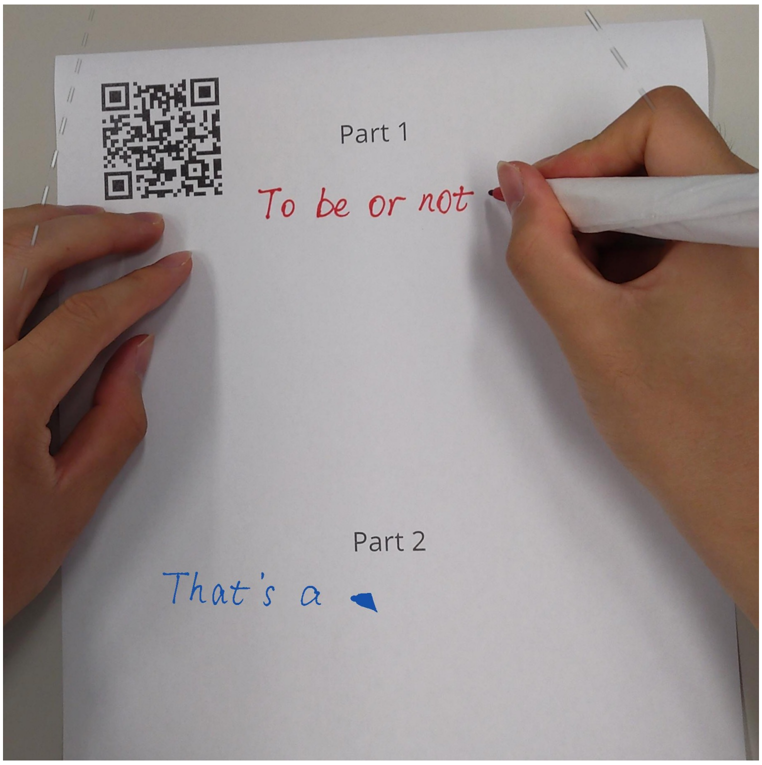


Fig. 4.4 Alice's view when she is sharing strokes with Bob

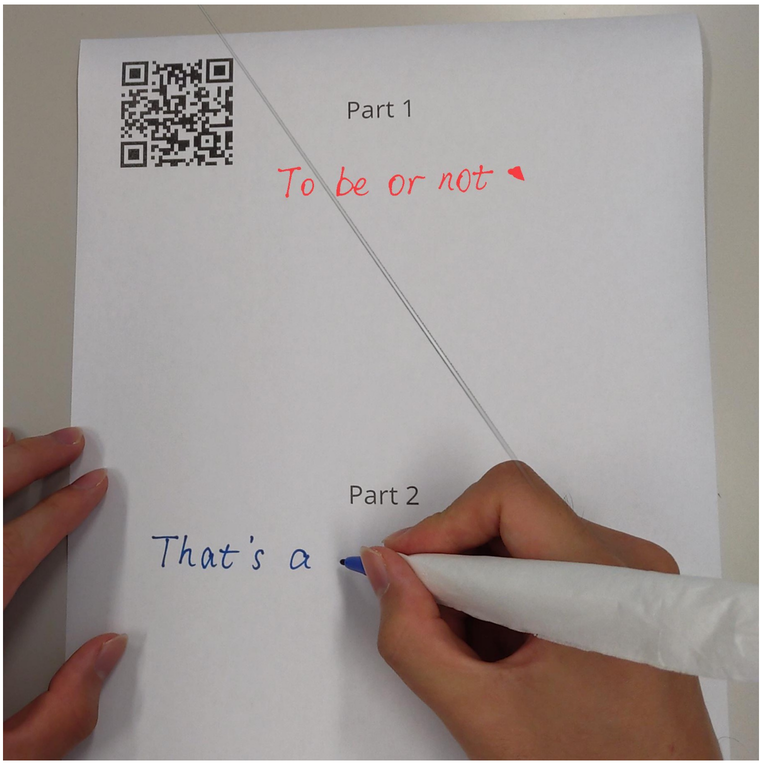


Fig. 4.5 Bob's view when he is sharing strokes with Alice

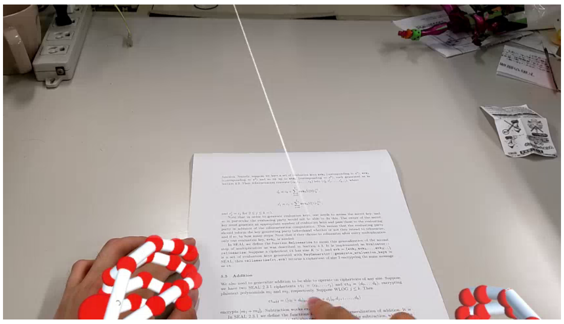
For example, suppose Alice and Bob need to write a report jointly. Alice is responsible for Part 1 and uses red pen, and Bob is responsible for Part 2 and uses blue pen. As shown in Fig. 4.4 and Fig. 4.5, Alice and Bob can use Strokes Sharing function to share their strokes with each other, so that they can see each other's strokes and make sure they have similar ideas when trying to solve problem.

4.4 Hand Shape Sharing

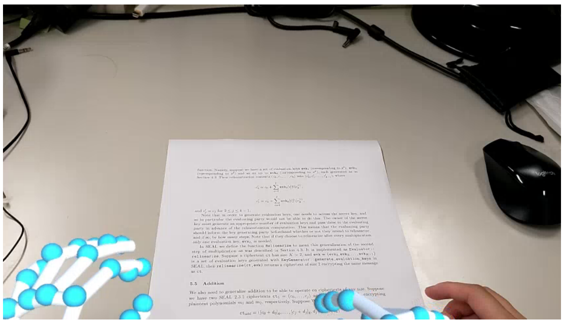
When one user is performing a hand movement, the hand shape can be synchronized to other users' view. In this way, every user can see other users' hand shape and hand movement even if they are not face to face.

When users are sharing their hand shapes, the system will use depth camera provided by head mounted device to capture the joints of users' hands and update the positions of hand joints every single frame. User's hand joints are rendered in different colors in order to distinguish between different users. In this system, only other users' hand shape will be rendered in one's view, and the current user's own joints will not be rendered, reducing redundant superimposition in user's view.

Hand shape and hand movement is an important way to express oneself, especially in some cases that are related to geometry and actions. By using hand shape, user can also grab other users' attention and communicate with each other. Since the users can perform hand movement in any way as they like, this function also allows users to express themselves in a more freely and unlimited way.



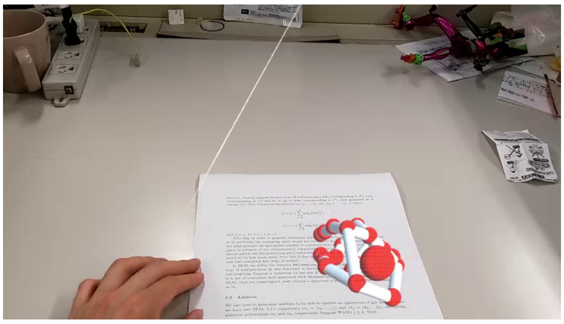
(a) Alice's view



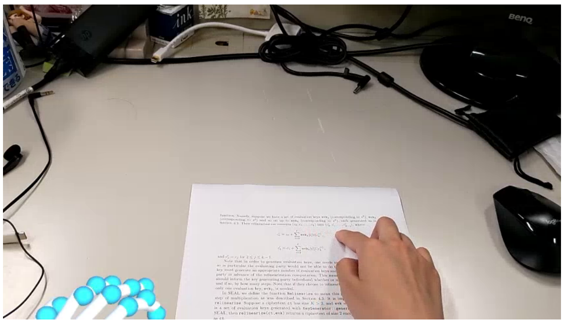
(b) Bob's view

Fig. 4.6 Alice points out a formula to grab Bob's attention

Suppose Alice and Bob are reading the same document. As shown in Fig. 4.6, Alice forgets where the definition of a formula is, so she uses hand shape sharing to grab Bob's attention. After Bob noticed, he can use finger to highlight the definition part, as Fig. 4.7 shows. Last, Alice can use an OK posture to tell Bob that she has understood, as Fig. 4.8 shows.

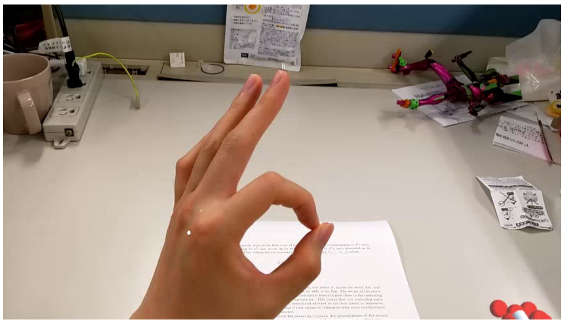


(a) Alice's view

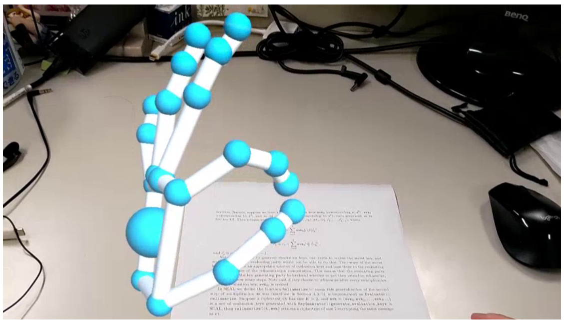


(b) Bob's view

Fig. 4.7 Bob uses finger to highlight the definition part



(a) Alice's view



(b) Bob's view

Fig. 4.8 Alice uses OK posture to tell Bob that she has understood

4.5 Board Sharing

When reading document, user can save contents, such as an image or a paragraph, that he or she is interested in by drawing “⌈” and “⌋” at the diagonal ends of the desired region, as shown in Fig. 4.9.

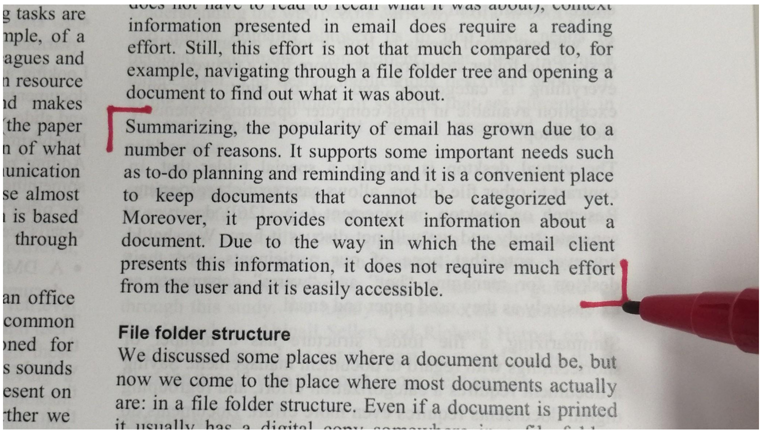


Fig. 4.9 User can save contents by drawing “⌈” and “⌋”

After drawing “⌈” and “⌋” to enclose the region, the system will use camera provided by head mounted device to capture the document and crop the enclosed region of the document. User can choose to save contents privately or publicly, as Fig. 4.10 shows, and the cropped region will be uploaded. Privately saved contents will only be available to the user who cropped them, while publicly saved contents are accessible to every other user.

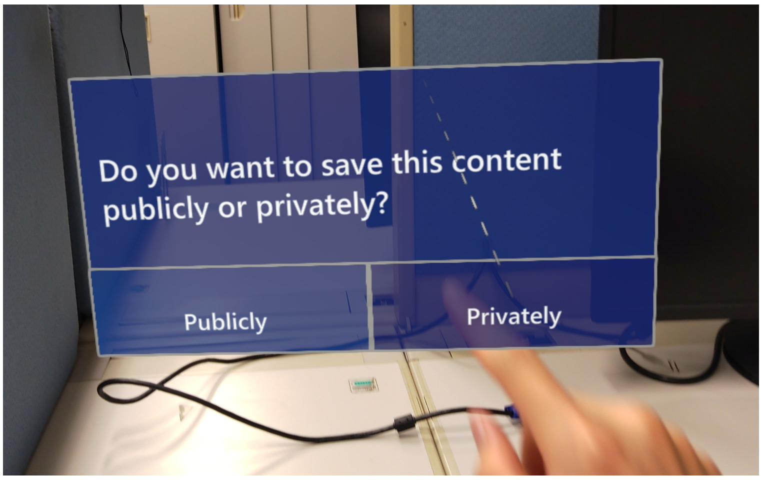


Fig. 4.10 User can choose to save contents privately or publicly

The saved contents are saved in two lists, one is for private saved contents, and another is for publicly saved contents, as shown in Fig. 4.11. User can touch on saved contents to view it in detail, as Fig. 4.12 shows. Above the contents, user name with different color is used to distinguish between uploaders.

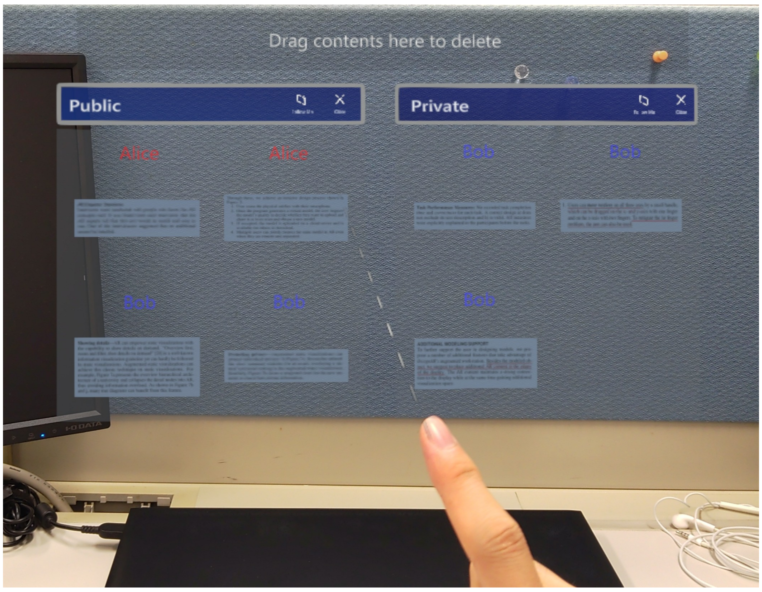


Fig. 4.11 Overview of Board Sharing

User can modify the sharing access right by grabbing and dragging the contents in the lists. User can drag contents created by himself or herself from private part to public part to set contents from private to public. Also, user can drag content created by himself or herself from public part to private part to set contents from public to private, as Fig. 4.13 shows. Above the public part and private part is the delete part. If user finds he or she is not interested in some contents created by himself or herself any more, he or she can grab and drag the contents to delete part to delete the contents.

Suppose Alice and Bob continue reading the same paper. Bob finds some exciting parts. He crops contents that he is interested in and saves them privately, and crops contents that are helpful for both his and Alice's research and saves them publicly. Both Alice and Bob can touch on the saved contents in their view to read the contents in detail. Now, Alice and Bob have collected some contents, and some contents are saved publicly and some are private, as Fig. 4.14 shows. If Alice published some contents she is interested in but not so helpful for

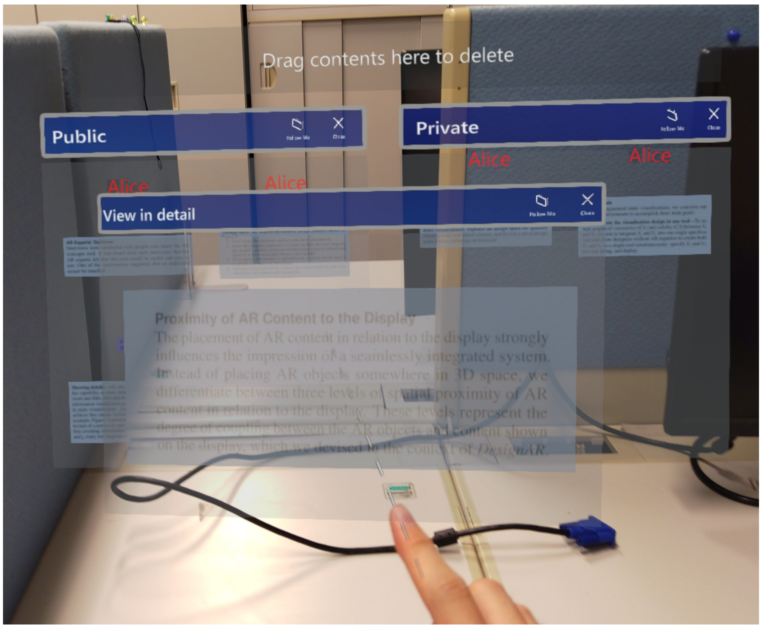


Fig. 4.12 Touch on saved contents to view in detail

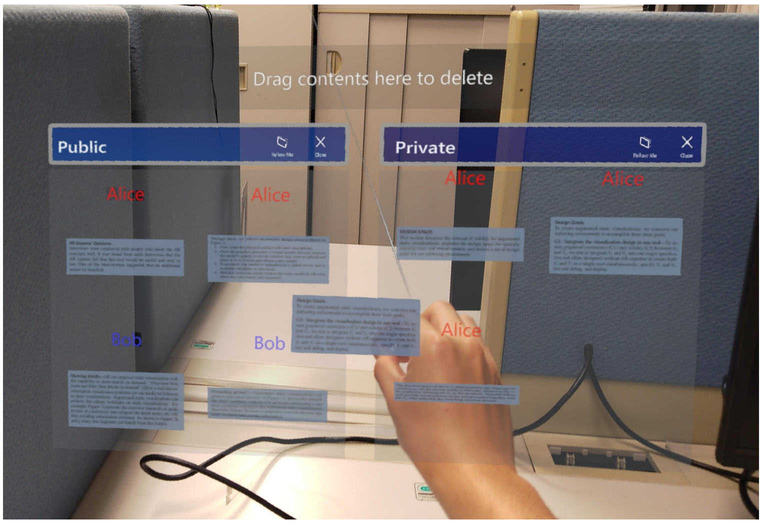
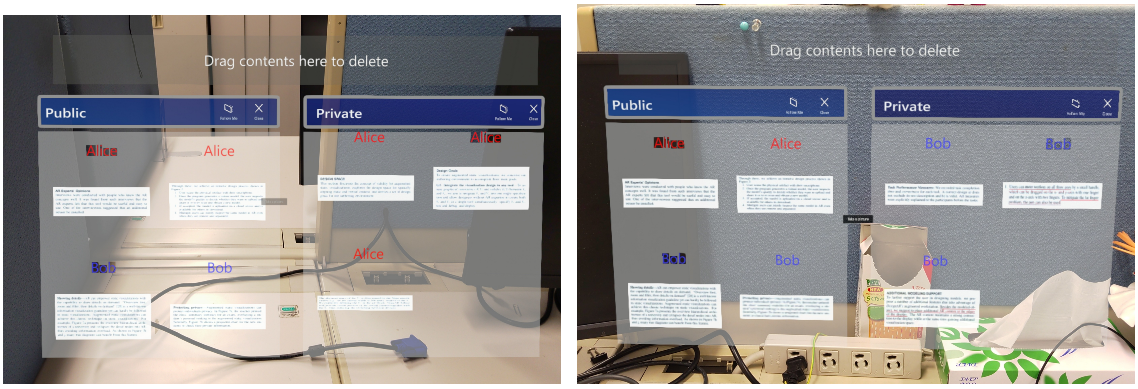


Fig. 4.13 Drag content to modify access right



(a) Alice's view (b) Bob's view

Fig. 4.14 Alice and Bob have collected some contents

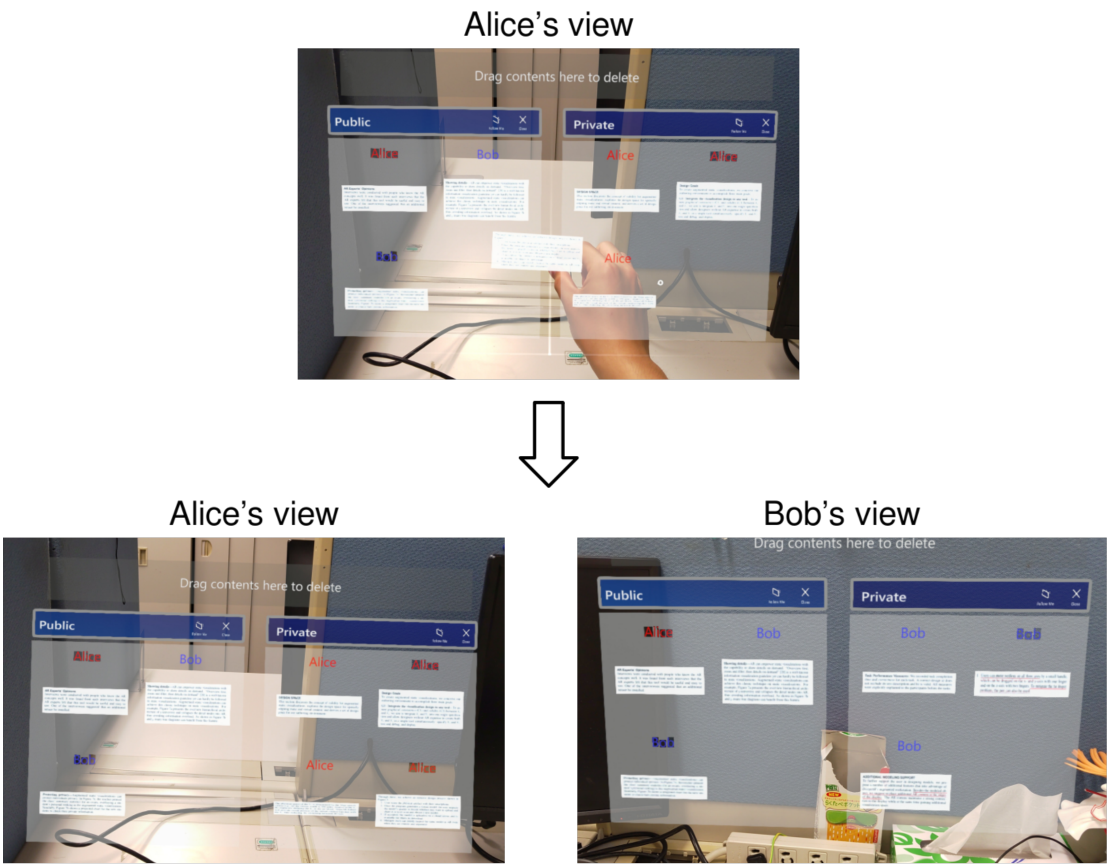


Fig. 4.15 Alice drags public content to private part

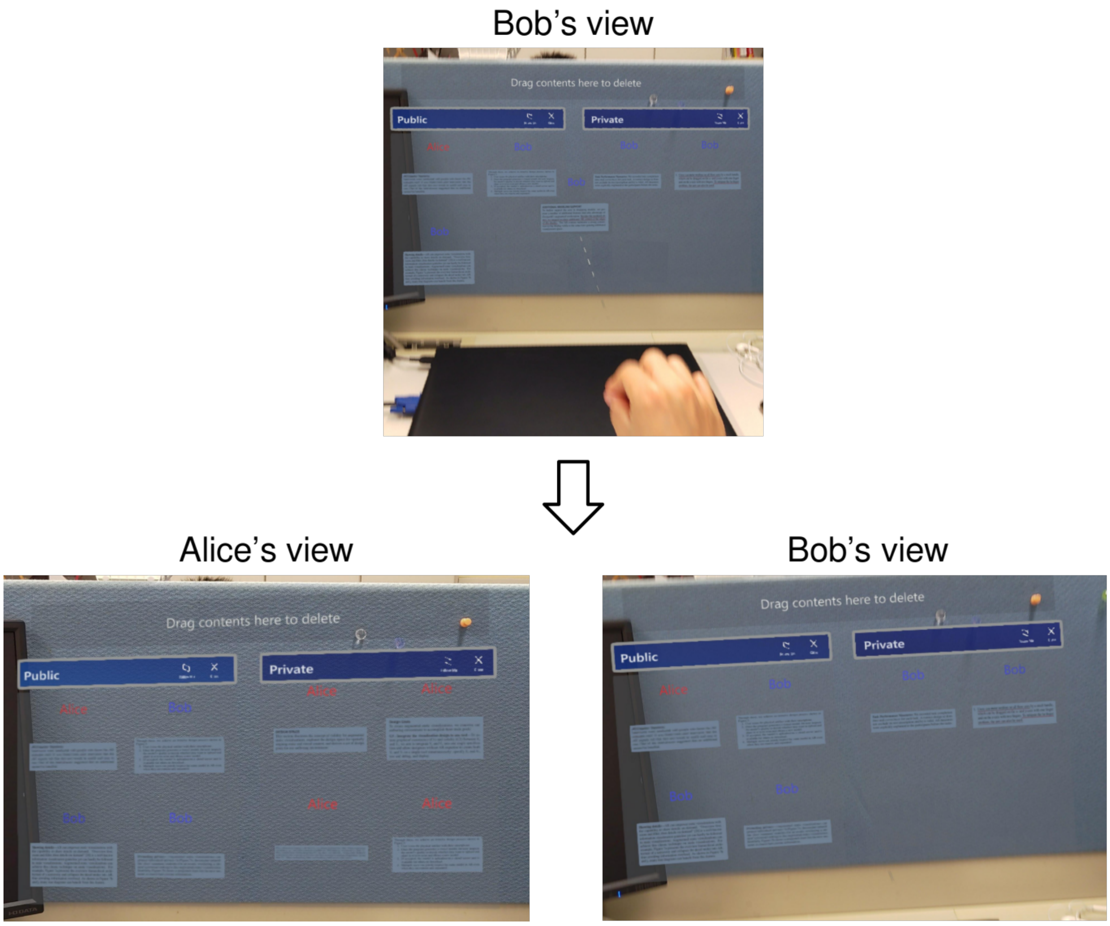


Fig. 4.16 Bob drags private content to public part

their research, she doesn't need to delete these contents. Instead, she can drag these contents from public part to private part, as Fig. 4.15 shows. If Bob finds some contents he saved privately are also helpful for Alice's research, he can drag these contents from private part to public part, as shown in Fig. 4.16. And if Alice finds some contents she saved are not so interesting, she can drag these contents to delete part to delete them.

4.6 Usage Scenario

Suppose Alice and Bob are two college students and they need to read a paper together. But for some reason, they cannot meet each other face to face, so they choose to use this system to finish this task.

When they are reading the paper, Alice meets a formula she doesn't understand. However, Alice isn't good at mathematics, so she writes down a comment, says "I don't quite understand this formula", and put this comment next to the formula. After seeing comment from Alice, Bob can write down a comment saying "I can show you the derivation process of this mathematical formula." besides Alice's comment to reply Alice.

Then, Bob can write down the derivation process of this mathematical formula with Strokes Sharing enabled, so that Alice can see Bob's pen strokes in real time and keep up with Bob's thoughts. If Alice has any doubt about the derivation process of this mathematical formula, she can also use Strokes Sharing to underline or encircle the part she doesn't fully understand, so that Bob can give explanation in more detail.

Alice finds this derivation process of the mathematical formula is a little complex for her, and she forgets some definitions of the variables in the derivation process. So Alice enables Hand Shape Sharing function, and uses her finger to highlight the formulas that contain these variables. Seeing this, Bob can also enable Hand Shape Sharing function and point out the definitions of these variables. After seeing Bob's instruction and thinking about it, Alice has understood the idea of solving the problem, so she shows Bob an OK gesture, telling Bob that she has understood this part.

As they continue reading the paper, they find some exciting parts. Bob uses his pen to draw "⌈" and "⌋" on the paper document, crops the contents that he is interested in and saves them privately so that only himself can see these contents. Alice also finds some parts that she thinks are helpful for both her and Bob's research, so she uses his pen to draw "⌈" and "⌋" on the paper document around these parts, crops these contents and saves them publicly so that both Bob and her can see these contents. If Bob finds some contents he saved privately are also helpful for Alice's research, he can use hands to drag these contents from private part to public part so that both Alice and him can share these contents together. If Alice published some contents she is interested in but not so helpful for their research, she can drag these contents from public part to private part so that only herself can see these contents.

By exchanging ideas and collaborate with each other by using this system, Alice and Bob get a good understanding about this paper and finish the task brilliantly.

Chapter 5

System Implementation

This chapter explains the implementation of each part of this system. At first, the hardware used in this system and software environment for development of this system will be introduced. Then, technical points of each part of this system will be explained.

5.1 System Hardware

Some hardware are needed to create this AR Document Collaboration System.

In this system, Microsoft HoloLens 2 is used as the mixed reality head mounted device, as shown in Fig. 5.1. The depth camera installed on Microsoft HoloLens 2 can detect user's hand gesture and position to collect information that is needed in this system.

Also, a laptop is needed to support the programming environment. Table 5.1 shows the information of this laptop.

Category	**Information**
Operation System	Microsoft Windows 10 Pro
CPU	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99GHz
RAM	8.00GB

Table 5.1 The information of laptop

Fig. 5.1 Microsoft HoloLens 2

5.2 Development Environment

Some development tools are also used to complete code writing. The development environment of this system is as follows.

- Windows 10 SDK. It provides some libraries and tools for building Universal Windows Platform applications that run on Microsoft HoloLens 2.
- Unity 2019.4.28f1 LTS. It provides the development platform for applications based on Universal Windows Platform which can be run on Microsoft HoloLens 2. Unity is also implemented with an XR SDK which helps this system to handle input and output events.
- Mixed Reality Toolkit 2.7.0 (MRTK v2). It provides a set of components and features, interface user experiences and some basic configuration modules to accelerate cross-platform mixed reality application development in Unity.
- Visual Studio 2019. It is used for writing and debugging C# scripts in this system.

5.3 Comment Sharing

In order to upload a comment created by user, the comment needs to be captured first. To capture the comments, the camera provided by the head mounted device will be used. After capturing the photo, it is necessary to crop the comment from the photo. Since markers are printed to every piece of comment note, the way to crop the comment from the photo is to crop a fixed size image at a fixed distance from the marker. After cropping comment from the photo, the comment can be uploaded so that every user in this system can see this comment.

Consequently, Comment Sharing can be divided into three main processes, i.e. Marker Recognition, Photo Capturing, and Photo Cropping. The following subsections will explain these processes one by one.

5.3.1 Marker Recognition

QR code is used as marker in this system. In order to track the marker, a package called `Microsoft.MixedReality.QR` is used. The class `QRCodeWatcher` provided by package `Microsoft.MixedReality.QR` plays an important role in QR code tracking. First of all, it is necessary to get whether QR code detection is supported on the current device by calling `IsSupported()`. Then, call `RequestAccessAsync()` to request user consent before using web camera for QR code detection. Once these are checked and there are no errors, the function `Start()` can be called to start detect QR codes. The code is shown in Fig. 5.2.

```
async protected virtual void Start() {
  // Get whether QR code detection is supported on current device.
  IsSupported = QRCodeWatcher.IsSupported();

  // Request user consent before using QR code detection.
  capabilityTask = QRCodeWatcher.RequestAccessAsync();

  accessStatus = await capabilityTask;
  capabilityInitialized = true;
}
```

Fig. 5.2 QR code initialization

The marker in this system has two functions, one is to track the position of physical document, and another is to recognize and trigger specific functions. For the first function, package named `Windows.Perception.Spatial` are used to percept the spatial information around the head mounted device and build a spatial graph coordinate system. In the source code of this system, at first, an object named `CoordinateSystem` of class `SpatialCoordinateSystem` are initialized. After this, `CoordinateSystem` should call `Windows.Perception.Spatial.Preview.SpatialGraphInteropPreview.Create CoordinateSystemForNode()` to update the coordinate system every single frame. If this function doesn't return `null`, the coordinate system is acquired successfully. Then, location information of virtual objects in augmented reality environment can be updated to meet the position of objects in real environment, and the function of tracking of the position of physical document is achieved. The framework of code is shown in Fig. 5.3.

```
public class SpatialGraphCoordinateSystem : MonoBehaviour {
  private SpatialCoordinateSystem CoordinateSystem = null;
  void Start() {
    if (CoordinateSystem == null) {
      CoordinateSystem = Windows.Perception.Spatial.Preview.
        SpatialGraphInteropPreview.CreateCoordinateSystemForNode(id);
      if (CoordinateSystem == null)
        Debug.Log("Failed to acquire coordinate system");
    }
  }
  void Update() {
      UpdateLocation();
  }
  private void UpdateLocation();
}
```

Fig. 5.3 Framework of `SpatialGraphCoordinateSystem`

For the second function, when setting up QR code tracking function, some functions need to be registered so that these functions can be called automatically when events like adding a new QR code happen, as Fig. 5.4 shows.

```
private void SetupQRTracking() {
  try {
    // Get reference for QRCodeWatcher
    qrTracker = new QRCodeWatcher();
    // Disable tracker first.
    IsTrackerRunning = false;
    // Event representing the addition of a QR Code.
    qrTracker.Added += Watcher_Added;
    // Event representing the update of a QR Code.
    qrTracker.Updated += Watcher_Updated;
    // Event representing the removal of a QR Code.
    qrTracker.Removed += Watcher_Removed;
    // Event representing the enumeration of QR Codes completing after
    // a Start call.
    qrTracker.EnumerationCompleted += Watcher_EnumerationCompleted;
  }
  catch (Exception ex) {
    Debug.Log("Exception starting the tracker " + ex.ToString());
  }
}
```

Fig. 5.4 Register functions for every event

`Watcher_Added`, `Watcher_Updated`, `Watcher_Removed` and `Watcher_EnumerationCompleted` are all names of functions. As a result, function to capture and upload comment can be wrapped in these functions.

5.3.2 Photo Capturing

After the system detects the marker on the physical document, the document can be recognized and the position of this document can be tracked. After the system detects the marker on the comment note, the photo of the comment on the note can be captured.

To capture a photo by camera on head mounted device, package `UnityEngine.Windows.WebCam` is used. Class `PhotoCapture` provided by this package is used to capture a photo from the web camera and stores it in memory. First, some basic information of camera, like width, height, pixel format and so on are initialized to an object of class `Resolution`.

Then, the camera on head mounted device is activated to take a picture. At last, it is very important to deactivate the camera and shutdown the photo capture resource, otherwise the camera on head mounted device will always be occupied, resulting in the wasting of resource.

The flow chart of Photo Capturing function is shown in Fig. 5.5.

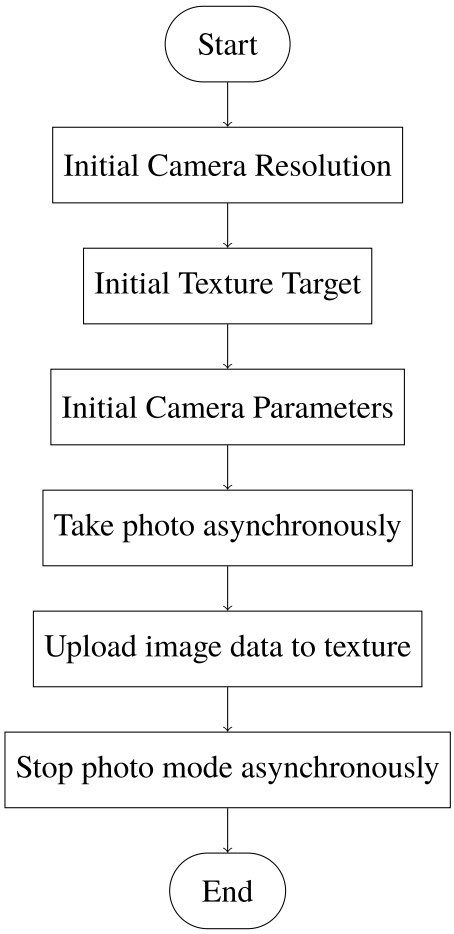


Fig. 5.5 Flow chart of Photo Capturing

What is noticeable is that if Marker Recognition part and Photo Capturing part are combined directly, Photo Capturing part will not work correctly. The reason is that both Marker Recognition part and Photo Capturing part need to use camera on head mounted device. If Marker Recognition is enabled when the system launches, when the system is trying to take a photo for Photo Capturing, camera is occupied by Marker Recognition function. Consequently, to solve this problem, Marker Recognition function should be disabled temporarily when system is capturing image.

5.3.3 Photo Cropping

After the system captures the photo of the comment note, the comment part on the note can be cropped. The way to crop comment from the photo is to crop a fixed size image at a fixed distance from the marker. Package `Microsoft.MixedReality.QR` provides a class named `QRCode`, which implemented `PhysicalSideLength` member variable. `PhysicalSideLength` returns the physical side length of detected QR code. Since the size of every comment note is the same, and the relative position of every element on the comment note is fixed, after getting the position and the physical side length of QR code, the start point and end point for cropping can be easily got by ratio calculation.

After cropping comment from the photo, the comment can be uploaded so that every user in this system can see this comment.

5.4 Strokes Sharing

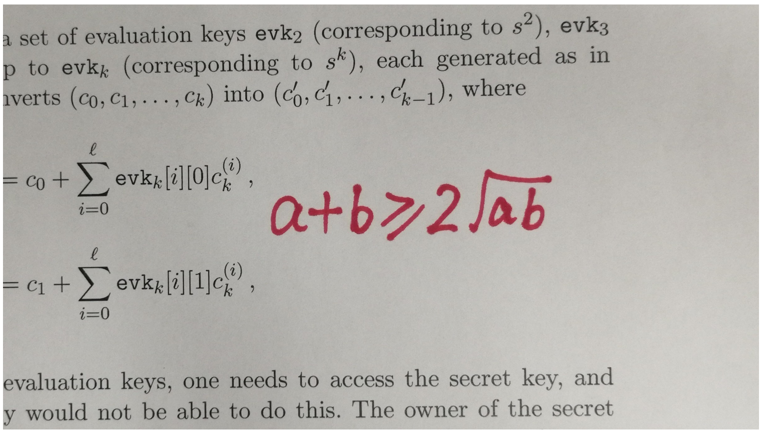
In order to share strokes created by user, the physical document needs to be captured first. To capture the physical document, the camera provided by the head mounted device will be used. After capturing the photo, it is necessary to extract the pen strokes by color from the photo. After extracting the pen strokes from the photo, the pen strokes can be uploaded and rendered with different colors according to the user who created them so that every user in this system can see these pen strokes.

Consequently, Strokes Sharing can be divided into two main processes, i.e. Photo Capturing and Color Extracting. The main process of Photo Capturing is roughly the same with Section 5.3.2, which will not be repeated here. The following subsection will explain Color Extracting process.

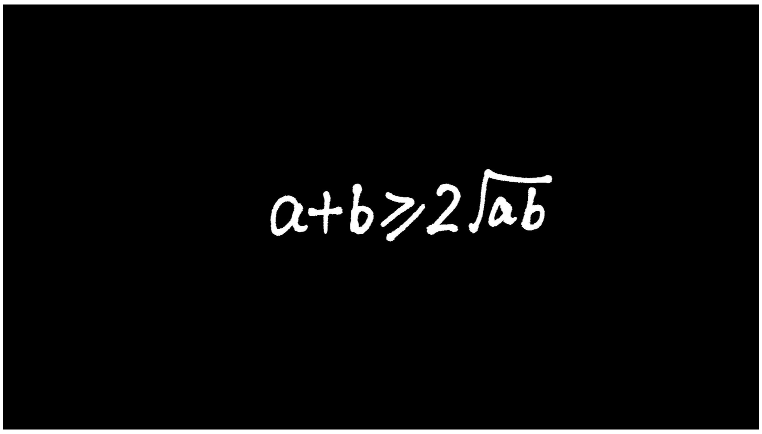
5.4.1 Color Extracting

This function uses two colors as standard colors to extract, one is blue (Color code: #1851AA), and another one is red (Color code: #F73E43). However, users can customize the color to extract as they like theoretically. Here, the above-mentioned two colors are used as an example.

When extracting color, the color in RGB color space should be converted to HSV color space. This is because by converting color from RGB color space to HSV color space, the extracting result will not be easily affected by ambient brightness if a threshold value of hue is set.



(a) Original captured image for Strokes Sharing



(b) Binarized image for Strokes Sharing

Fig. 5.6 Original captured image and binarized image for Strokes Sharing

In order to transmit the pen strokes in an effective way, the pen strokes extracted by color will be binarized. By doing this, the strokes data can be saved in a two-dimensional array of bool type. The written part is represented as `1`, and the blank part is represented as `0`. The original image captured by camera on head mounted device is shown in Fig. 5.6(a).

The binarized image after extracting color is shown in Fig. 5.6(b), where the written part is white, and the blank part is black.

The Color Extracting will be performed every short period of time, and the generated two-dimensional array of bool type containing pen stroke data will be synchronized to other users' device. When receiving the pen stroke data, the system will render the pen strokes according to the data and show these pen strokes in user's view, which allows user to see others' pen strokes in realtime.

5.5 Hand Shape Sharing

In order to share the hand shape between users, the system will first track the position of every hand joint of user. The position of every joint will be transmitted to other users' devices. Small balls will be rendered in the position of hand joints of remote users, and bones will be rendered between joints to rebuild the model of hands. The rotation and position of every joint and bone will be updated through network in real time, so that Hand Shape Sharing function can be achieved.

Consequently, Hand Shape Sharing can be divided into two main processes, i.e. Joint Rendering and Bone Rendering. The following subsections will explain these processes one by one.

5.5.1 Joint Rendering

The depth camera installed on head mounted device can detect the position of user's hand, and some APIs provided by Microsoft can track user's hand joint. The enumerated type `TrackedHandJoint` lists tracked hand joints that are supported. Fig. 5.7 shows the names of these joints.

To superimpose joint prefab on joint, it is necessary to disable each hand joint marker object so there aren't any errors if HoloLens can't locate a joint. After each hand joint marker object is disabled, `HandJointUtils.TryGetJointPose()` function can be used to look for the hand joints need to track. If the desired joint is found, the hand joint ball can be enabled,

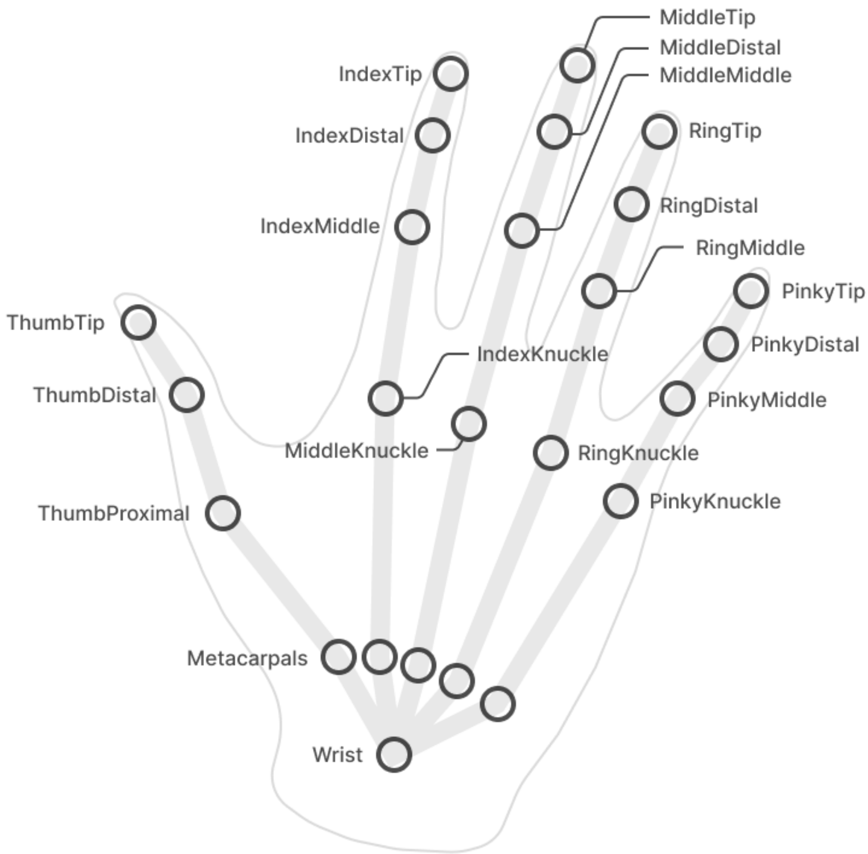


Fig. 5.7 Hand Joint Names

and it's position can be set to right place. To update the position and rotation of hand joint balls, the above mentioned process should be called every single frame. The pseudo code of this process is shown in Fig. 5.8.

```
void UpdateJoints() { // Will be called every single frame.
  for (every joint need to track) {
    // Disable each hand joint marker object.
    joint.GetComponent<Renderer>().enabled = false;
    if (HandJointUtils.TryGetJointPose(joint, handedness, out pose)) {
      // If desired joint is found, enable the hand joint ball.
      joint.GetComponent<Renderer>().enabled = true;
      // Set its position to right place
      joint.transform.position = pose.Position;
    }
  }
}
```

Fig. 5.8 Pseudo code of Joint Rendering

Hand joints in left hand and right hand need to be dealt with separately, and every single joint need to be dealt with separately. It is not intuitive to render every joint on user's hand, and to make the hand model more intuitive, not all hand joints need to be rendered. As a result, the target hand model is shown in Fig. 5.9, and the Index Metacarpal, Middle Metacarpal, Ring Metacarpal and Wrist are not rendered in implementation of this system.

5.5.2 Bone Rendering

After the positions of hand joints are detected, a cylinder can be drawn between two hand joints to connect them as bone to make the rebuilt hand model more intuitive, as Fig. 5.9 shows. First, it is necessary to determine which two joints need a bone to connect between them. This data can be saved in a two-dimensional array. Next is to draw bones between these hand joints pairs. The process of drawing a bone between two hand joints can be roughly divided the following four steps.

1. Get the position of the two hand joints to calculate the center position of the bone.

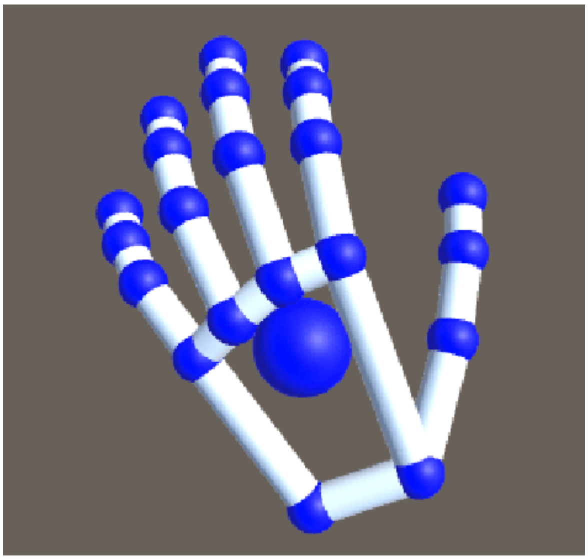


Fig. 5.9 Hand Model

2. Calculate the rotation degree according to the position of the two hand joints.

3. Calculate the distance between the two hand joints according to their position as the length of the bone.

4. Apply the value of position, rotation and length to the bone object.

The pseudo code of this process is shown in Fig. 5.10.

What is noticeable is that in the first step of drawing bones between hand joints, it is necessary to use `HandJointUtils.TryGetJointPose()` function to check whether both of the desired two hand joints can be found or not. If one of the desired hand joints cannot be found, this process should be terminated, since if the position of one of the hand joints is unknown, there is no way to continue drawing bone. Since the position of hand joint is saved as a three-dimensional vector, if the bone starts from the position of one hand joint and ends at the position of another hand joint, the position of bone should satisfy

$$\overrightarrow{\text{bonePos}} = (\overrightarrow{\text{startPos}} + \overrightarrow{\text{endPos}}) \times \frac{1}{2} \tag{5.1}$$

```
void UpdateBones() { // Will be called every single frame.
  for (every bone need to draw) {
    if (HandJointUtils.TryGetJointPose(startJoint, handedness,
        out pose))
      // If desired joint is found, record its position.
      startPos = startJoint.transform.position;
    else
      // Otherwise, this process should be terminated.
      continue;
    if (HandJointUtils.TryGetJointPose(endJoint, handedness,
        out pose))
      endPos = endJoint.transform.position;
    else
      continue;
    bonePos = CalculateBonePos();
    boneRotation = CalculateBoneRotation();
    boneLength = CalculateBoneLength();
    bone.Apply(bonePos, boneRotation, boneLength);
  }
}
```

Fig. 5.10 Pseudo code of Bone Rendering

In the second step, according to the subtraction of vector, the rotation of bone should satisfy

$$\overrightarrow{\text{boneRotation}} = \overrightarrow{\text{endPos}} - \overrightarrow{\text{startPos}} \tag{5.2}$$

In the third step, the distance between the two hand joints can be calculated by `Vector3.Distance(startPos, endPos)`.

Bones between joints in left hand and right hand need to be dealt with separately, and every single bone between two hand joints need to be dealt with separately. After the position, rotation and scale information of hand joints and bones are transmitted through network, users can share their hand shape between each other.

5.6 Board Sharing

In order to share the contents cropped from physical document among users, the physical document needs to be captured first. To capture the physical document, the camera provided by the head mounted device will be used. After capturing the photo, it is necessary to extract the two characters ("⌈" and "⌋") by color from the photo. If "⌈" and "⌋" are extracted from the photo, the enclosed region from the photo is ready to crop. After the system crops the enclosed region from the photo and user sets the accessibility, the content can be uploaded.

Consequently, Board Sharing can be divided into two main processes, i.e. Photo Capturing, Color Extracting, Template Matching, and Photo Cropping. The main process of Photo Capturing, Color Extracting, and Photo Cropping are roughly the same with Section 5.3.2, Section 5.4.1, and Section 5.3.3 respectively, which will not be repeated here. The following subsection will explain Template Matching process. Also, to prevent system from cropping the content which has already be saved, Duplicate Checking is necessary.

5.6.1 Template Matching

Template matching is a method to check whether the small area of image, called pattern, exists in the whole image, called a template, by raster scanning. This algorithm is often used for detecting the position of a target object in an image, counting the number of objects, and detecting object movement. There are some methods for template matching, which are listed in Table 5.2.

Method Name	**Meaning**
TM_SQDIFF	Square Difference Matching Method
TM_SQDIFF_NORMED	Normalized Square Difference Matching Method
TM_CCORR	Correlation Matching Methods
TM_CCORR_NORMED	Normalized Cross-Correlation Matching Method
TM_CCOEFF	Correlation Coefficient Matching Methods
TM_CCOEFF_NORMED	Normalized Correlation Coefficient Matching Method

Table 5.2 Methods used for Template Matching

Though TM_CCOEFF method provides high accuracy, the amount of calculation is large. Since this system is implemented on a head mounted device, TM_CCORR method family will be used for it has lower load. In order to reduce the impact of ambient light on the system, the normalized version of TM_CCORR, TM_CCORR_NORMED, will be used.

The mathematical representation of TM_CCORR_NORMED method is

$$R(x,y) = \frac{\sum_{x',y'}(T(x',y') \cdot I(x+x',y+y'))}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}} \tag{5.3}$$

Here, T is the template image which will be compared to the source image, I is the source image in which is expected to find a match to the template image. $T(x,y)$ is the brightness of the template image at pixel (x,y), $I(x,y)$ is the brightness of the source image at pixel (x,y), and $R(x,y)$ is the normalized value of the brightness correlation at pixel (x,y). Let w and h be the width and height of template image, and W and H be the width and height of source image, then the value ranges of x and y are $[0, W-w]$ and $[0, H-h]$ respectively.

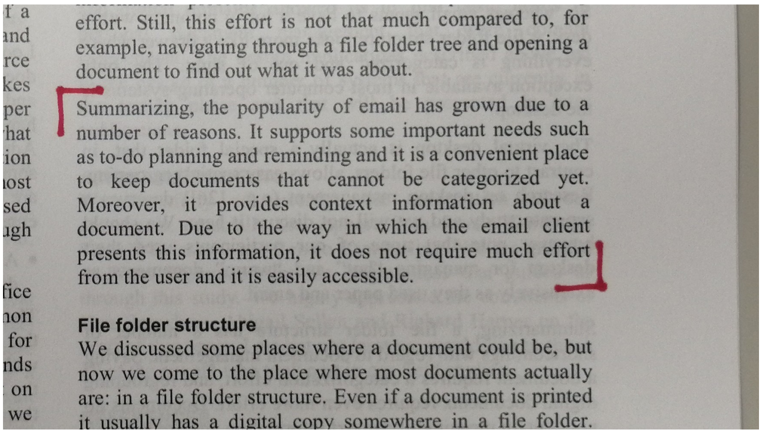
For higher efficiency, the captured photo will be binarized before Template Matching. Fig. 5.11 shows an example. Fig. 5.11(a) shows the original captured image, and Fig. 5.11(b) shows the binarized image. The template images used for Template Matching are shown in Fig. 5.12.

After Template Matching, the coordinates of "⌈" and "⌋" can be found, and the contents enclosed will be cropped.

Sometimes, a photo containing "⌈" and "⌋" which are not matched may be captured, Fig. 5.13 gives an example. In such case, the content shouldn't be cropped. To handle this situation, only when "⌈" is 70px or more to the left of "⌋", or 70px or more above "⌋" will the crop function be called. Here, 70px is a threshold and can be modified if necessary.

5.6.2 Duplicate Checking

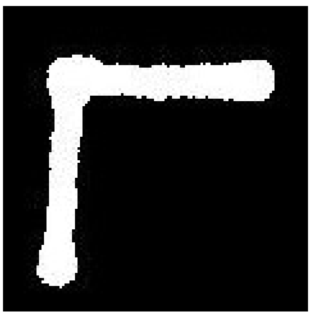
When user is reading the physical documents, some contents that have already be saved may be captured by system. In this case, the captured content shouldn't be saved again. To recognize whether the captured content has be saved or not, two steps are performed here.



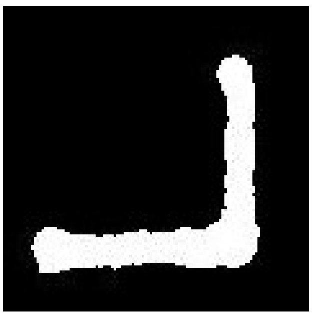
(a) Original captured image for Board Sharing

(b) Binarized image for Board Sharing

Fig. 5.11 Original captured image and binarized image for Board Sharing

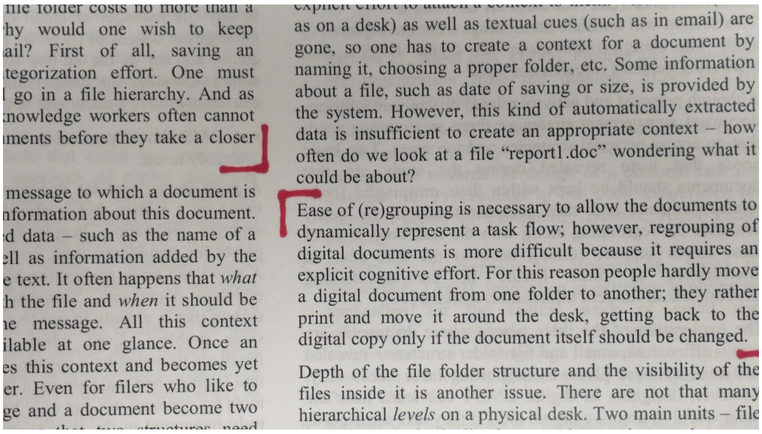


(a) Template image for "⌈"



(b) Template image for "⌋"

Fig. 5.12 Images used for Template Matching



'hy would one wish to keep
ail? First of all, saving an
tegorization effort. One must
l go in a file hierarchy. And as
nowledge workers often cannot
ments before they take a closer

message to which a document is
nformation about this document.
d data – such as the name of a
ell as information added by the
e text. It often happens that *what*
h the file and *when* it should be
e message. All this context
ilable at one glance. Once an
es this context and becomes yet
er. Even for filers who like to
ge and a document become two

as on a desk) as well as textual cues (such as in email) are gone, so one has to create a context for a document by naming it, choosing a proper folder, etc. Some information about a file, such as date of saving or size, is provided by the system. However, this kind of automatically extracted data is insufficient to create an appropriate context – how often do we look at a file "report1.doc" wondering what it could be about?

Ease of (re)grouping is necessary to allow the documents to dynamically represent a task flow; however, regrouping of digital documents is more difficult because it requires an explicit cognitive effort. For this reason people hardly move a digital document from one folder to another; they rather print and move it around the desk, getting back to the digital copy only if the document itself should be changed.

Depth of the file folder structure and the visibility of the files inside it is another issue. There are not that many hierarchical *levels* on a physical desk. Two main units – file

Fig. 5.13 A photo containing "⌈" and "⌋" which are not matched

Every time a content is cropped, the aspect ratio and the binarized data of this content will be saved and attached to this content. When "⌈" and "⌋" are captured, the aspect ratio of the enclosed region will be calculated first. If there is no aspect ratio in the saved content that is the same as the detected content, the detected content will be regarded as a new content and saved. Otherwise, if the aspect ratio is the same as an existing content, first crop a square of 70px $\times$ 70px at a random position from existing content, then crop a square in the same relative position from captured image and scale it to 70px $\times$ 70px, next calculate the result using Equation 5.3. If the average value of the result is lower than a threshold (e.g., 0.8), then the detected content will be regarded as a new content and saved.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this research, the combination of physical and digital documents is explored, and a system that allows users to collaborate with each other between documents in digital and physical world is designed and implemented.

This system leverages the benefits of digital technologies while preserving the advantages of using physical documents to help users in collaboration scenarios. Specifically, this system tracks the location of physical documents, as well as user's gestures and strokes, to augment the paper documents by making use of Augmented Reality. Users are then able to interact with system, and users' interaction data will be transmitted through network. By using functions provided by this system, users can share comments, pen strokes, hand movement and highlight contents on document with each other to achieve collaboration. A usage scenario is represented to show how these functions can be used together and be helpful to collaboration.

Technically, this system uses Marker Recognition to recognize and track paper document and support comment capturing for Comment Sharing. Color Extracting is used to extract and rebuild users' pen strokes for Strokes Sharing. It is also used together with Template Matching to detect the cropped region for Board Sharing. By getting hand joint information from depth camera, a hand model is rebuilt and so that hand shape can be shared with other users in Hand Shape Sharing.

Overall, this system has successfully accomplished the following requirements.

- Allow users to exchange their comments about the documents they are reading.
- Allow users to use pen strokes and hand actions to communicate with each other.
- Allow users to share contents that they are interested in on the document with others.

6.2 Future Work

Although an AR document collaboration system has been proposed and implemented in this research, there are still some limitations to be improved.

For example, in Comment Sharing, the comment uploading is triggered when the system detects the marker besides the comment part on note, which is not so natural. The expected way is uploading comment when user tap a "Submit" button printed on the note with pen nib. Also, since this system uses color to track user's pen strokes, the pen nib should be thicker than the pen normally used. If the pen strokes are not obvious enough, it is difficult for the system to distinguish between pen strokes and document background. These problems can be solved by using smart pen.

References

[1] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, 2001.

[2] Jorge Luis Bacca Acosta, Silvia Margarita Baldiris Navarro, Ramon Fabregat Gesa, Sabine Graf, et al. Augmented reality trends in education: a systematic review of research and applications. *Journal of Educational Technology and Society, 2014, vol. 17, núm. 4, p. 133-149*, 2014.

[3] Mark Billinghurst, Hirokazu Kato, and Ivan Poupyrev. The MagicBook - moving seamlessly between reality and virtuality. *IEEE Computer Graphics and Applications*, 21(3):6–8, 2001.

[4] Peter Brandl, Christoph Richter, and Michael Haller. *NiCEBook: Supporting Natural Note Taking*, page 599–608. Association for Computing Machinery, New York, NY, USA, 2010.

[5] Jeffrey W. Chastine, Ying Zhu, and Jon A. Preston. A Framework for Inter-referential Awareness in Collaborative Environments. In *2006 International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 1–5, 2006.

[6] Damien Clergeaud, Joan Sol Roo, Martin Hachet, and Pascal Guitton. Towards Seamless Interaction between Physical and Virtual Locations for Asymmetric Collaboration. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, VRST '17, New York, NY, USA, 2017. Association for Computing Machinery.

[7] Raimund Dachselt and Sarmad Al-Saiegh. Interacting with printed books using digital pens and smart mobile projection. In *Proc. of the Workshop on Mobile and Personal Projection (MP2)@ ACM CHI*, volume 11, 2011.

[8] Katherine M. Everitt, Meredith Ringel Morris, A.J. Bernheim Brush, and Andrew D. Wilson. DocuDesk: An interactive surface for creating and rehydrating many-to-many

linkages among paper and digital documents. In *2008 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, pages 25–28, 2008.

[9] Susan R. Fussell, Leslie D. Setlock, and Robert E. Kraut. Effects of Head-Mounted and Scene-Oriented Video Systems on Remote Collaboration on Physical Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, page 513–520, New York, NY, USA, 2003. Association for Computing Machinery.

[10] Lei Gao, Huidong Bai, Weiping He, Mark Billinghurst, and Robert W. Lindeman. Real-Time Visual Representations for Mobile Mixed Reality Remote Collaboration. In *SIGGRAPH Asia 2018 Virtual & Augmented Reality*, SA '18, New York, NY, USA, 2018. Association for Computing Machinery.

[11] Jerônimo Gustavo Grandi, Henrique Galvan Debarba, and Anderson Maciel. Characterizing Asymmetric Collaborative Interactions in Virtual and Augmented Realities. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 127–135, 2019.

[12] Raphael Grasset, Andreas Duenser, Hartmut Seichter, and Mark Billinghurst. *The Mixed Reality Book: A New Multimedia Reading Experience*, page 1953–1958. Association for Computing Machinery, New York, NY, USA, 2007.

[13] David Holman, Roel Vertegaal, Mark Altosaar, Nikolaus Troje, and Derek Johns. *Paper Windows: Interaction Techniques for Digital Paper*, page 591–599. Association for Computing Machinery, New York, NY, USA, 2005.

[14] Matthew Hong, Anne Marie Piper, Nadir Weibel, Simon Olberding, and James Hollan. Microanalysis of Active Reading Behavior to Inform Design of Interactive Desktop Workspaces. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, page 215–224, New York, NY, USA, 2012. Association for Computing Machinery.

[15] Andreea-Carmen Ifrim, Florica Moldoveanu, Alin Moldoveanu, and Alexandru Grădinaru. LibrARy – Enriching the Cultural Physical Spaces with Collaborative AR Content. In Jessie Y. C. Chen and Gino Fragomeni, editors, *Virtual, Augmented and Mixed Reality*, pages 626–638, Cham, 2021. Springer International Publishing.

[16] Mads Møller Jensen, Sarah-Kristin Thiel, Eve Hoggan, and Susanne Bødker. Physical Versus Digital Sticky Notes in Collaborative Ideation. *Computer Supported Cooperative Work (CSCW)*, 27(3):609–645, Dec 2018.

[17] Jeongyun Kim, Jonghoon Seo, and Tack-Don Han. AR Lamp: Interactions on Projection-Based Augmented Reality for Interactive Learning. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, IUI '14, page 353–358, New York, NY, USA, 2014. Association for Computing Machinery.

[18] Konstantin Klamka and Raimund Dachselt. *IllumiPaper: Illuminated Interactive Paper*, page 5605–5618. Association for Computing Machinery, New York, NY, USA, 2017.

[19] Hideki Koike, Yoichi Sato, and Yoshinori Kobayashi. Integrating Paper and Digital Information on EnhancedDesk: A Method for Realtime Finger Tracking on an Augmented Desk System. *ACM Trans. Comput.-Hum. Interact.*, 8(4):307–322, December 2001.

[20] Youngho Lee, Katsutoshi Masai, Kai Kunze, Maki Sugimoto, and Mark Billinghurst. A Remote Collaboration System with Empathy Glasses. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pages 342–343, 2016.

[21] Yuan Li, David Hicks, Wallace S. Lages, Sang Won Lee, Akshay Sharma, and Doug A. Bowman. ARCritique: Supporting Remote Design Critique of Physical Artifacts through Collaborative Augmented Reality. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 585–586, 2021.

[22] Zhen Li, Michelle Annett, Ken Hinckley, Karan Singh, and Daniel Wigdor. *HoloDoc: Enabling Mixed Reality Workspaces That Harness Physical and Digital Content*, page 1–14. Association for Computing Machinery, New York, NY, USA, 2019.

[23] Chunyuan Liao, Qiong Liu, Bee Liew, and Lynn Wilcox. *Pacer: Fine-Grained Interactive Paper via Camera-Touch Hybrid Gestures on a Cell Phone*, page 2441–2450. Association for Computing Machinery, New York, NY, USA, 2010.

[24] Catherine C. Marshall. Annotation: From Paper Books to the Digital Library. In *Proceedings of the Second ACM International Conference on Digital Libraries*, DL '97, page 131–140, New York, NY, USA, 1997. Association for Computing Machinery.

[25] Danakorn Nincarean Eh Phon, Mohamad Bilal Ali, and Noor Dayana Abd Halim. Collaborative Augmented Reality in Education: A Review. In *2014 International Conference on Teaching and Learning in Computing and Engineering*, pages 78–83, 2014.

[26] Márcio S. Pinho, Doug A. Bowman, and Carla M.D.S. Freitas. Cooperative Object Manipulation in Immersive Virtual Environments: Framework and Techniques. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST ’02, page 171–178, New York, NY, USA, 2002. Association for Computing Machinery.

[27] Patrick Reipschläger and Raimund Dachselt. DesignAR: Immersive 3D-Modeling Combining Augmented Reality with Interactive Displays. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, ISS ’19, page 29–41, New York, NY, USA, 2019. Association for Computing Machinery.

[28] Joan Sol Roo and Martin Hachet. One Reality: Augmenting How the Physical World is Experienced by Combining Multiple Mixed Reality Modalities. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST ’17, page 787–795, New York, NY, USA, 2017. Association for Computing Machinery.

[29] Kentaro Takano, Hirohito Shibata, and Kengo Omura. Effects of Paper on Cross-Reference Reading for Multiple Documents: Comparison of Reading Performances and Processes between Paper and Computer Displays. In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction*, OzCHI ’15, page 497–505, New York, NY, USA, 2015. Association for Computing Machinery.

[30] Craig S. Tashman and W. Keith Edwards. *Active Reading and Its Discontents: The Situations, Problems and Ideas of Readers*, page 2927–2936. Association for Computing Machinery, New York, NY, USA, 2011.

[31] Xiangyu Wang and Phillip S. Dunston. Comparative Effectiveness of Mixed Reality-Based Virtual Environments in Collaborative Design. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(3):284–296, 2011.

[32] Pierre Wellner. Interacting with Paper on the DigitalDesk. *Commun. ACM*, 36(7):87–96, July 1993.

[33] Josef Wideström, Ann-Sofie Axelsson, Ralph Schroeder, Alexander Nilsson, Ilona Heldal, and Åsa Abelin. The Collaborative Cube Puzzle: A Comparison of Virtual and Real Environments. In *Proceedings of the Third International Conference on Collaborative Virtual Environments*, CVE ’00, page 165–171, New York, NY, USA, 2000. Association for Computing Machinery.

[34] Daiki Yamaji and Jiro Tanaka. Recognition of Written Cues System for Users of General Paper Media. In Sakae Yamamoto, editor, *Human Interface and the Management of Information. Information and Knowledge Design*, pages 466–476, Cham, 2015. Springer International Publishing.