Media Engineering and Technology Faculty
German University in Cairo

Optimizing Drag and Drop

Bachelor Thesis

Author: Adham Ahmed Kamel
Supervisors: Dr. Wael Aboelsadaat

Submission Date: 23 June, 2024

Media Engineering and Technology Faculty
German University in Cairo

Optimizing Drag and Drop

Bachelor Thesis

Author: Adham Ahmed Kamel
Supervisors: Dr. Wael Aboelsadaat

Submission Date: 23 June, 2024

This is to certify that:

(i) the thesis comprises only my original work toward the Bachelor Degree

(ii) due acknowlegement has been made in the text to all other material used

---

Adham Ahmed Kamel
23 June, 2024

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr. Wael Abouelsaadat, for his invaluable mentorship and guidance throughout this project. His expertise and insights have been instrumental in shaping this thesis, and I could not have achieved a fraction of this work without his support.

I am also deeply grateful to my friends and family for their unwavering support and encouragement. Their belief in me has been a constant source of motivation and strength.

Lastly, I would like to thank all the experiment test subjects who agreed to participate in my study. Your cooperation and willingness to engage in this research have been crucial to its success.

Abstract

This thesis presents an innovative drag-and-drop system which leverages natural hand gestures and spatial memory to enhance user interaction with moving files. Using MediaPipe's hands solution for hand tracking and gesture recognition, and incorporating machine learning for high accuracy, the system features secure palm authentication for cross-device functionality. An exploratory study demonstrated moderate accuracy in file placement recall, highlighting potential improvements with better user training and hardware enhancements. Despite limitations with standard webcams, the system shows promise, suggesting future work with depth cameras and sensory feedback to further improve user experience.

Contents

Acknowledgments **V**

Abstract **VII**

1 Introduction **1**
1.1 Desktop Computing Evolution 1
1.2 Motivation 1
1.3 Aim of the Project 2

2 Literature Review **3**
2.1 Related Works 3
2.1.1 Techniques for Reducing Physical Strain 3
2.1.2 Gesture-Based Data Transfer 4
2.1.3 Gesture-Based Data Transfer 5
2.1.4 Enhancing Accessibility 8
2.1.5 Improving Desktop Management 9
2.1.6 Improving Desktop Management 11
2.1.7 Tangible Interaction 12
2.1.8 Combining Gaze and Touch 14
2.2 Summary 17

3 System Design and Implementation **19**
3.1 Design Goals 19
3.2 System Overview 20
3.3 Explorative Study on Spatial Memory and Phalange Placement 21
3.3.1 Study Design and Objectives 21
3.3.2 Results 22
3.3.3 Discussion and Conclusions 22
3.4 Interaction Design 23
3.4.1 Interaction Techniques 23
3.4.2 Implementation 24
3.5 Flowchart Analysis and Detailed Interaction Process 26
3.5.1 Advanced Interaction Flow: Two-Handed Phalange Touch Detection and File Management 26

3.5.2 Single-Handed Interaction and Network Mode Operations 27
3.6 Conclusion 28

4 Conclusion **29**
4.1 Summary 29
4.2 Limitations 29
4.3 Future Work 30

Appendix **31**
List of Figures 33

References **34**

Chapter 1

Introduction

1.1 Desktop Computing Evolution

Desktop computing is always evolving, aiming to improve user experiences by making them better and easier. One common technique in modern computer interfaces is drag-and-drop, which has changed how we move data around. However, despite its widespread use, it can be challenging because it requires a lot of precision. This constant need for precision can easily get extremely tiring.

This thesis suggests a new way to make drag-and-drop better by using hand gestures. We want to use natural gestures as well as the 12 phalanges of our fingers to make it easier and more natural to use drag and drop. This idea should make things less frustrating and more efficient. Which will help users move data more seamlessly.

1.2 Motivation

The motivation for this research comes from the challenges of current drag-and-drop methods. Precise mouse control and long cursor movements can be frustrating and physically tiring. These traditional methods do not fully use the dexterity of the human hand, which is a very adaptable and expressive tool.

By adding hand gestures to the drag-and-drop experience, this research aims to overcome these problems. Using the 12 segments of our fingers for storage is a new idea that could greatly improve user experience by combining advanced technology with our natural hand skills.

While hand gestures are popular in virtual reality, gaming, and mobile devices, they are not widely used in desktop environments. This research aims to change that by using hand gestures to improve one of the most basic interactions in computing.

The decision to use the 12 finger segments is based on the detailed structure and function of the human hand. Each finger, except for the thumb, has three parts (proximal,

intermediate, and distal), while the thumb has two. This anatomy gives users 12 shelf like phalanges which they can place their data on. By using these natural abilities, the proposed system aims to provide an easy and responsive user experience that feels like part of our natural hand movements.

1.3 Aim of the Project

The main goal of this project is to design, develop, and test a gesture-based drag-and-drop system that uses the 12 phalanges of the fingers to store and move digital data. This means making a system that can correctly understand and respond to different hand gestures and put this function into a desktop environment.

The focus of this project is on the interaction itself, not going into details of hardware or advanced gesture recognition algorithms. It also does not look at compatibility with non-desktop interfaces. The main aim is to show the potential of this new interaction method, making people want to use it more and study it more in human-computer interaction.

To reach this goal, we will use a "black box" system architecture. This will separate the gestures from the main operating system and give a test environment for the interaction technique. The system will also use a layered approach, letting each phalange store many files at the same time. An experiment will test how many layers users can handle well, finding out the limits of this layered approach.

Chapter 2

Literature Review

This section discusses Related Works in order to showcase how far technology have come in terms of Human-Computer-Interaction

2.1 Related Works

2.1.1 Techniques for Reducing Physical Strain

Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-Operated Devices
Baudisch et al. [1] developed "Drag-and-Pop" and "Drag-and-Pick" methods to help users reach remote content on large touchscreens. These methods reduce physical effort when using large screens. With "Drag-and-Pop," far objects move closer when users start to drag them. "Drag-and-Pick" lets users pick objects from a distance. These techniques can lower physical strain but might cause screen clutter and unexpected movements, making the interface more complex. Also, setting up these techniques needs careful adjustment to ensure the moving parts do not interfere with other on-screen activities.

Drag-and-Pop: This technique improves upon traditional drag and drop such that it brings potential target locations closer to the user's cursor once the user starts dragging an item.

Drag-and-Pick: This interaction allows a user to perform an action like opening a folder or pressing an icon by just dragging across the screen. Once the user drags across an empty portion of the screen, all files on the screen are activated and move towards the cursor, allowing the user to select whichever icon he wishes to activate. Then, everything goes back to place.

Evaluation Results

The techniques were evaluated on a 15-foot wide interactive display wall. The user study compared Drag-and-Pop with traditional drag-and-drop methods. The findings

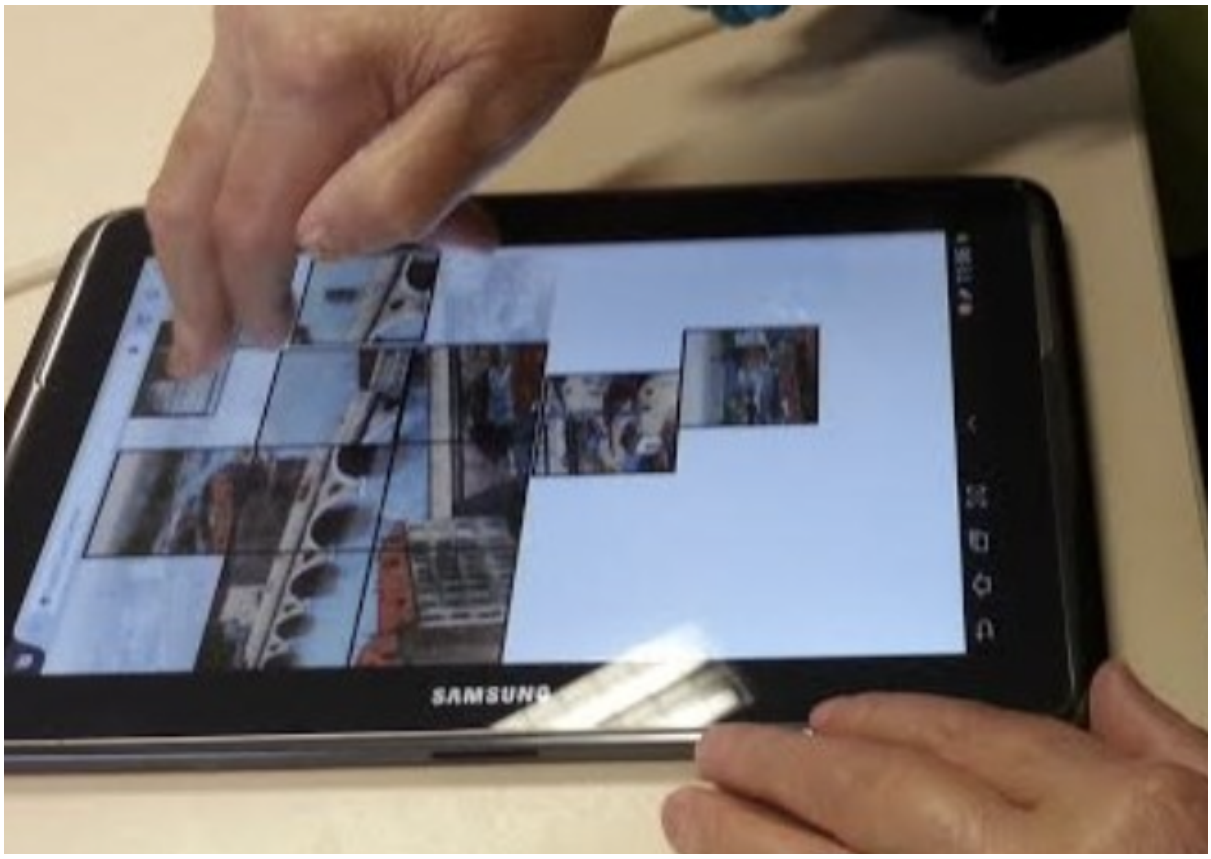

Figure 2.1: Older adult playing a tactile puzzle on a tablet. This figure demonstrates the practical application of touch-based interactions which can be extended to large displays. [1]

were that users were able to place items in files 3.7 times faster using Drag-and-Pop compared to traditional drag-and-drop. Drag-and-Pop had a significant error rate (6.7 percent). The errors were mainly due to how close the popped targets were, making it easy to accidentally drop an item in the wrong target. Participants mainly preferred Drag-and-Pop as it is less physically demanding and very efficient on large screens.

The evaluation demonstrated significant improvements in interaction speed, especially when multiple display units or large distances were involved. However, some participants suggested increasing the angle for destination targets to better accommodate their natural arm movements on large touch-sensitive displays.

2.1.2 Gesture-Based Data Transfer

Toss-It: Intuitive Information Transfer Techniques for Mobile Devices
Yatani et al. [8] created the "Toss-It" system. This system lets people share data between mobile devices using simple gestures like tossing. It feels natural to use, but it needs very precise gesture recognition, which is difficult to get right. It also requires special hardware,

which limits its use case. The system mimics the physical action of tossing an object, making the interaction feel more straightforward. The success of this method depends on how well it can recognize these gestures, and any mistakes in recognition can make users feel frustrated.

System Implementation

Toss-It: This system lets users send information from their mobile device to other devices with a simple toss gesture. To make this work, sensors are attached to the mobile device to accurately capture gestures. When the user starts a toss action using their device, sensors like accelerometers capture this movement. Then, the algorithm analyzes the data from the sensors to determine the direction and target device, and the information is sent to said device wirelessly.

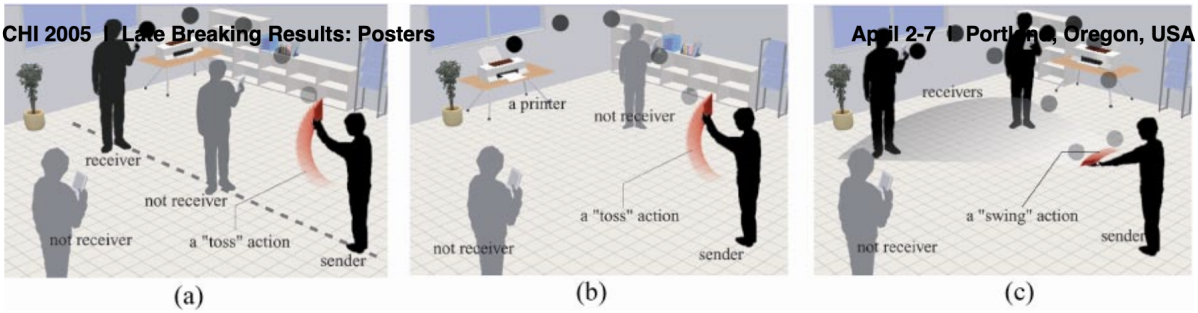

Figure 2.2: Intuitive information transfer techniques with Toss-It. (a) from a PDA to another PDA (b) from a PDA to a printer (c) from a PDA to multiple PDAs. [8]

Evaluation Results

The Toss-It system was tested to see how well it worked for transferring information with toss and swing actions. The main findings are that Toss-It could actually tell the difference between toss and swing actions, but the accuracy of detection changed based on how far the target was and the angle of the toss or swing. A user study was conducted where users were asked to use toss or swing actions to complete assigned data transfer tasks. The results were that for toss actions, transferring had a success rate of 70 percent to 85 percent depending on the distance of the target, and swing actions actually had variable success rates, mainly depending on the angle used, with the bigger angle having a higher probability of success.

2.1.3 Gesture-Based Data Transfer

SPARSH: A System for Touch-Based Data Transfer
SPARSH, created by Pranav Mistry et al. [5] uses the human body to transfer data. This system lets users touch a device to store data and then touch another device to transfer it. SPARSH has issues with data security and accidental activations, raising

privacy concerns. The system's reliance on touch for data transfer also raises suspicions about its effectiveness in different conditions.

System Implementation

SPARSH: The system uses a touch interface to know what to copy and where to pass it. Technically, the actual transfer of media happens via the information cloud, such as Dropbox or Google Drive. The user touches the data they wish to transfer, and once they touch said data, this data is stored in the cloud. Then the user touches another device to paste the selected data.

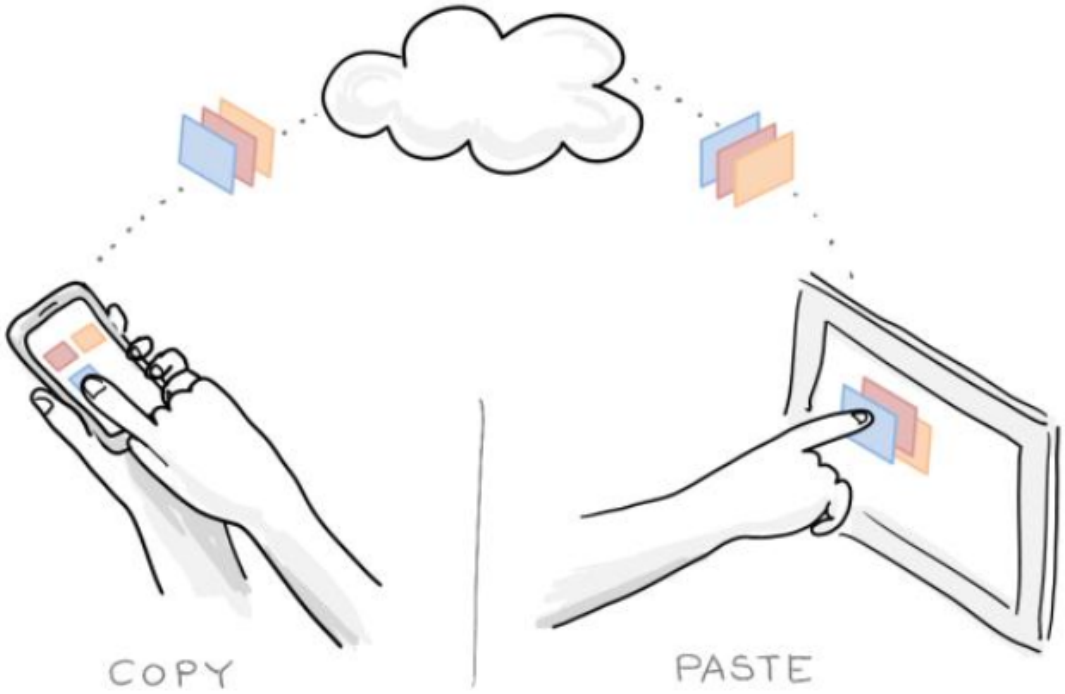

Figure 2.3: SPARSH – Touch to Copy, Touch to Paste.[5]

Figure 2.4: Moving a video link from a mobile screen to a larger digital display.[5]

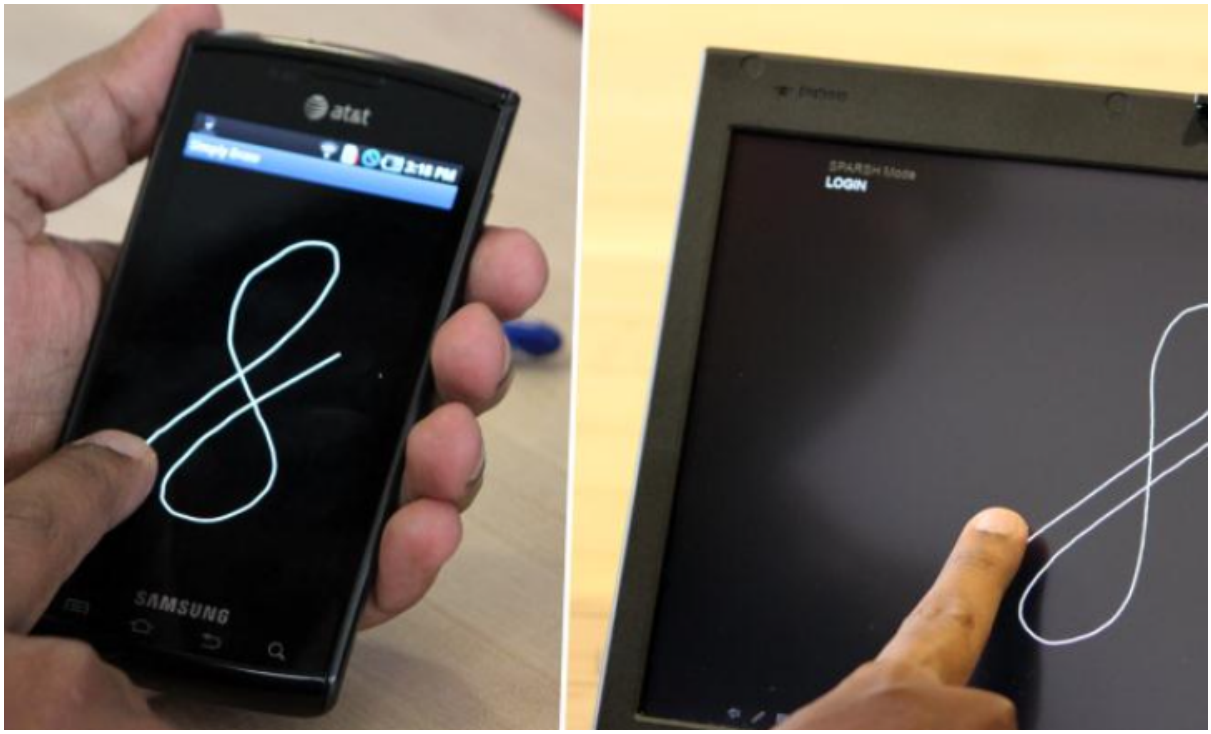

Figure 2.5: Identifying user by using a unique gestural sign.[5]

Evaluation Results

The SPARSH system was evaluated through various scenarios and user studies. The scenarios included transferring an address from a mobile phone to Google Maps on a computer, copying a phone number from a computer to a mobile phone, copying pictures from a phone to a tablet, and moving a video link from a mobile phone to a large display. The success rate of said scenarios varied according to the type of data and devices used. The participants found the touch interaction very easy to use, albeit there were some accidental activations reported. However, there are privacy concerns since the system stores the data for transfer in the cloud. All in all, users liked how easy SPARSH was to use and were very accustomed to the interaction after a very short practice period.

2.1.4 Enhancing Accessibility

Accessible Auditory Drag and Drop
Winberg and Hellstrom developed [7] "Accessible Auditory Drag and Drop," a system that help blind users by using sound for drag-and-drop tasks. This system offers a useful alternative to traditional methods for visually impaired users. However, it is hard to learn and can overwhelm some users. The system uses sounds to guide users through the drag-and-drop process, so they can do these tasks without having to see the screen.

System Implementation

Accessible Auditory Drag and Drop: The system is designed to enable blind users to perform drag-and-drop operations through auditory feedback. It works through the use of auditory zooming, where the system gives the user different levels of information based on how close the user is to the objects on screen. When the user is far from said object, they only hear a chime. Then, when they get close, they start hearing information about said object. The system works by dividing the screen into quadrants. The quadrants are recognized through a combination of their tone and audio orientation. For example, high tone and right audio orientation indicate the top right quadrant, while low tone and left audio orientation indicate the bottom left quadrant, and so on. The user interacts with the system using a stylus pen, which gives the exact position of the cursor on screen. Also, actions such as picking up, dragging, or dropping give haptic feedback to the user.

Evaluation Results

Two versions of the auditory interface were evaluated through user studies. The first was where four blind subjects learned the interface and got used to it. Then, the second study was that three of the original four came back and were asked to use a modified version of the system based on feedback from the first study. At first, users reported a steep learning curve, but they quickly learned how to locate and move objects, then found the system very easy and intuitive to use despite it seeming very complex at first.

2.1.5 Improving Desktop Management

Copy-and-Paste Between Overlapping Windows
Chapuis and Roussel [3] created techniques to make copy-and-paste tasks between overlapping windows easier. This method reduces the need to manage windows during these tasks. While it simplifies desktop tasks, it might disrupt familiar workflows because of automatic window adjustments. The system transfers information between applications without manually moving windows.

System Implementation

Restack and Roll Techniques: These techniques were designed to facilitate copy-paste operations by managing window overlaps. Restack works such that when a user starts a drag operation in an overlapping window setup, the potential target window is automatically brought into the foreground, enabling the user to smoothly drop the data into said window. Then it goes back into its original place. Roll works by, instead of changing the stacking order, rolling the overlapping non-target window to the side, allowing you to place the item wherever you want, then unrolling the window, making everything return back to what it was.

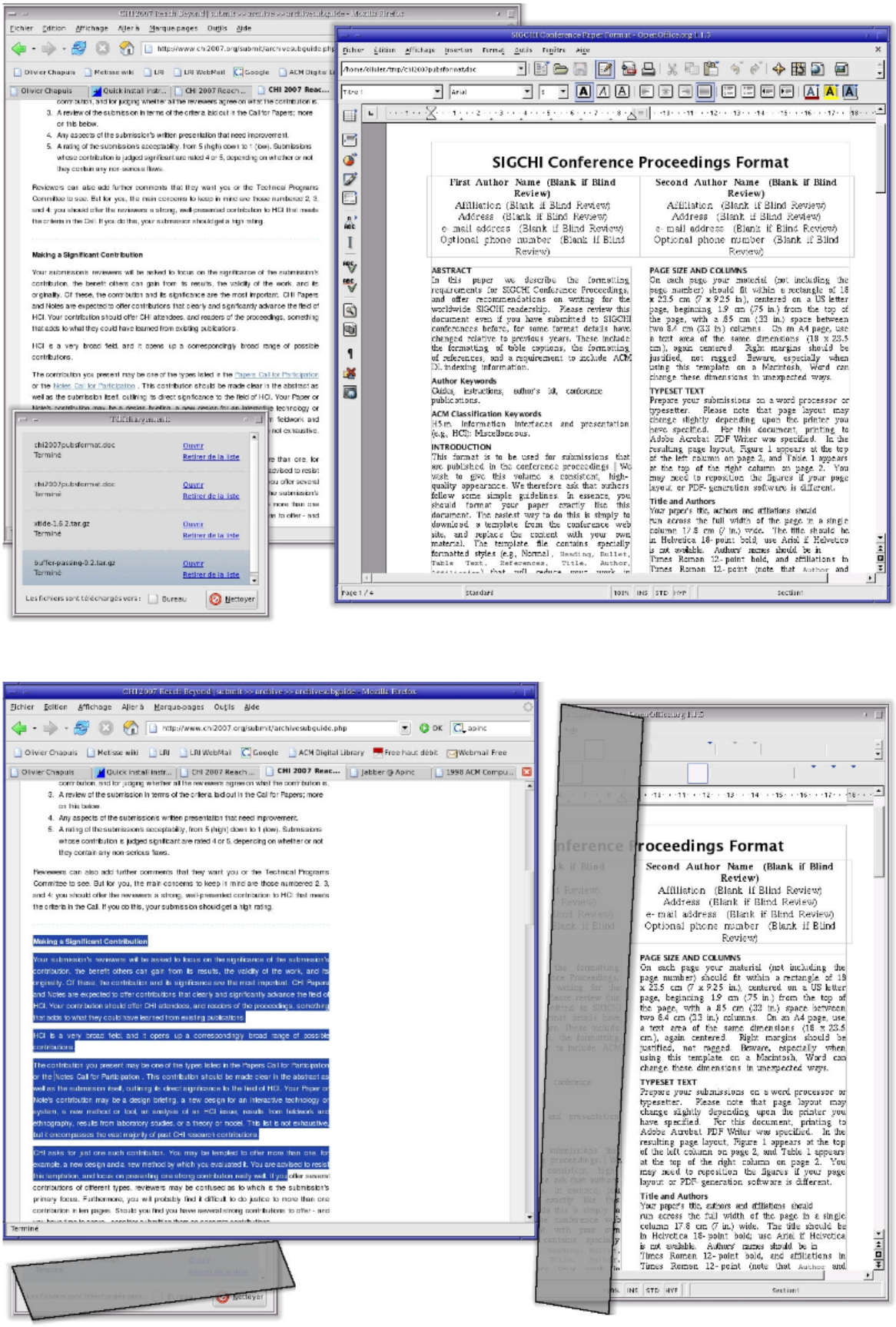

Figure 2.6: Figure 2.6: Rolling windows to reveal an overlapped one.[3]

Evaluation Results

These techniques were designed to facilitate copy-paste operations by managing window overlaps. Restack works such that when a user starts a drag operation in an overlapping window setup, the potential target window is automatically brought into the foreground, enabling the user to smoothly drop the data into said window, then it goes back into its original place. Roll works by, instead of changing the stacking order, rolling the overlapping non-target window to the side, allowing you to place the item wherever you want, then unrolling the window, making everything return back to what it was.

2.1.6 Improving Desktop Management

Multiblending: Displaying Overlapping Windows Simultaneously without the Drawbacks of Alpha Blending
Baudisch and Gutwin developed "Multiblending" [2] to improve how overlapping windows look. Instead of mixing everything together like alpha blending does, Multiblending lets users focus on what they find most important. It uses different blending levels for different parts of the screen to keep things clear and easy to read.

But Multiblending needs careful settings and a lot of computer power. This could be tricky for people who aren't tech-savvy to use.

System Implementation

Multiblending extends alpha blending by applying various blending weights to different visual features. It works by dealing with brightness, red-green contrast, blue-yellow contrast, and detailed data like edges or high-contrast areas. Each feature is separately blended using a different weight, which allows the program to protect the more important elements to the user while reducing their less important counterpart. The system automatically optimizes its blending weights in order to deal with different scenarios. Manual switching of the weights is also possible.

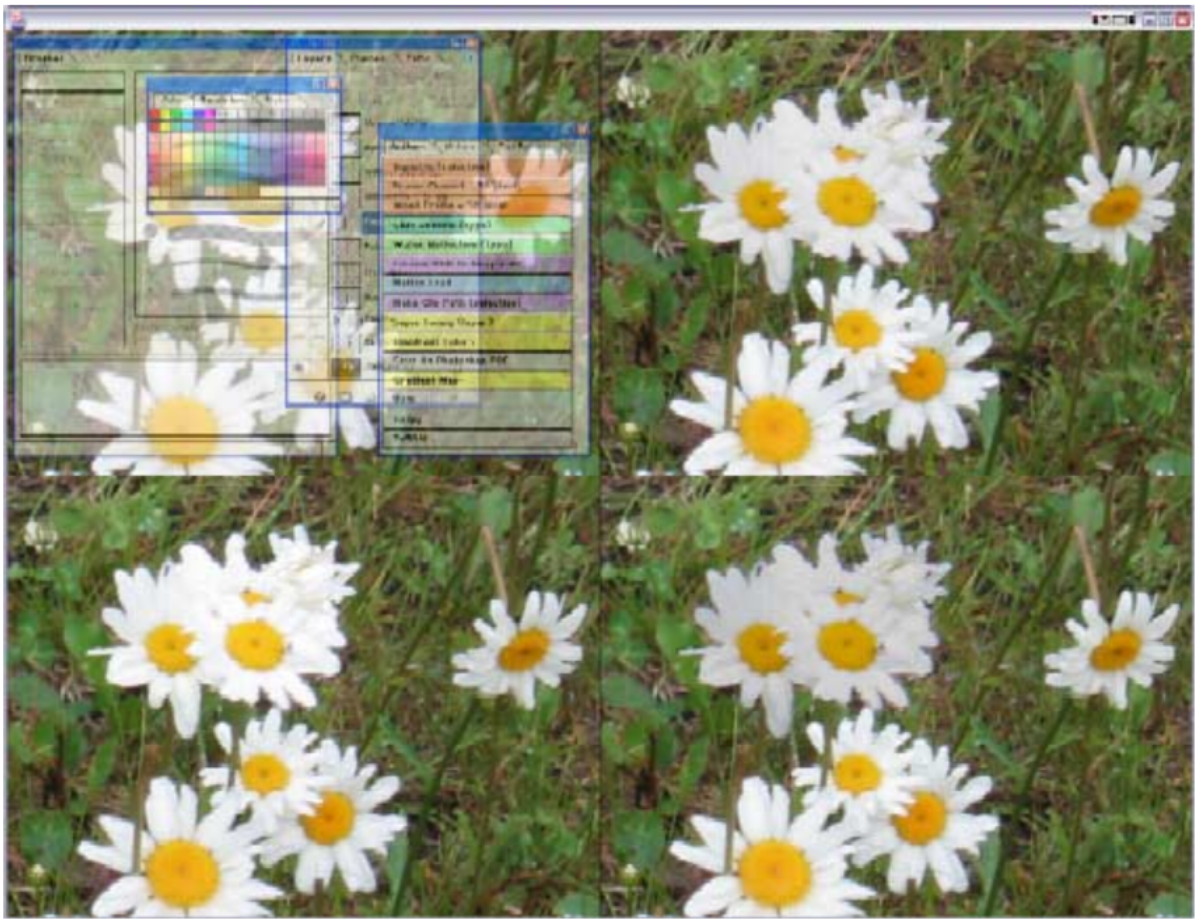

Figure 2.7: Background recognizability task (alpha-50 condition). A source image with overlaid palettes is shown at the top left, and three candidate images—near copies of the source image—are shown in the other three quadrants.[2]

Evaluation Results

Multiblending was evaluated through user studies comparing it with traditional alpha blending at different levels of opacity. In a task where participants matched a source image covered by palettes to one of three potential images, it was found that Multiblending made it easier to recognize the background and palette compared to alpha blending. Time-wise, the test participants took the same time to complete tasks when using Multiblending and alpha blending. Test subjects preferred Multiblending over alpha blending as, at any given time, they could see and interact with all programs on screen no matter their orientation.

2.1.7 Tangible Interaction

Slurp: Tangibility, Spatiality, and an Eyedropper
Zigelbaum et al. [9] developed "Slurp," an interaction system that uses a physical device

to transfer digital content. This system aims to make data transfer interactions more natural by relating them to physical actions. The system allows users to extract digital objects from physical items and inject them onto digital displays.

System Implementation

Slurp consists of a physical eyedropper-looking device that can interact with digital content. Users are able to extract digital content and store it in their Slurp device by taking said Slurp device and touching it to the place where the digital content was initially stored. This action is very similar to sucking up liquid using an eyedropper. The Slurp then holds the data until the user decides to transfer said data somewhere else by taking the Slurp device, which is now filled with data, and touching it to whichever medium the user wishes to transfer data to, again very similar to using an eyedropper.

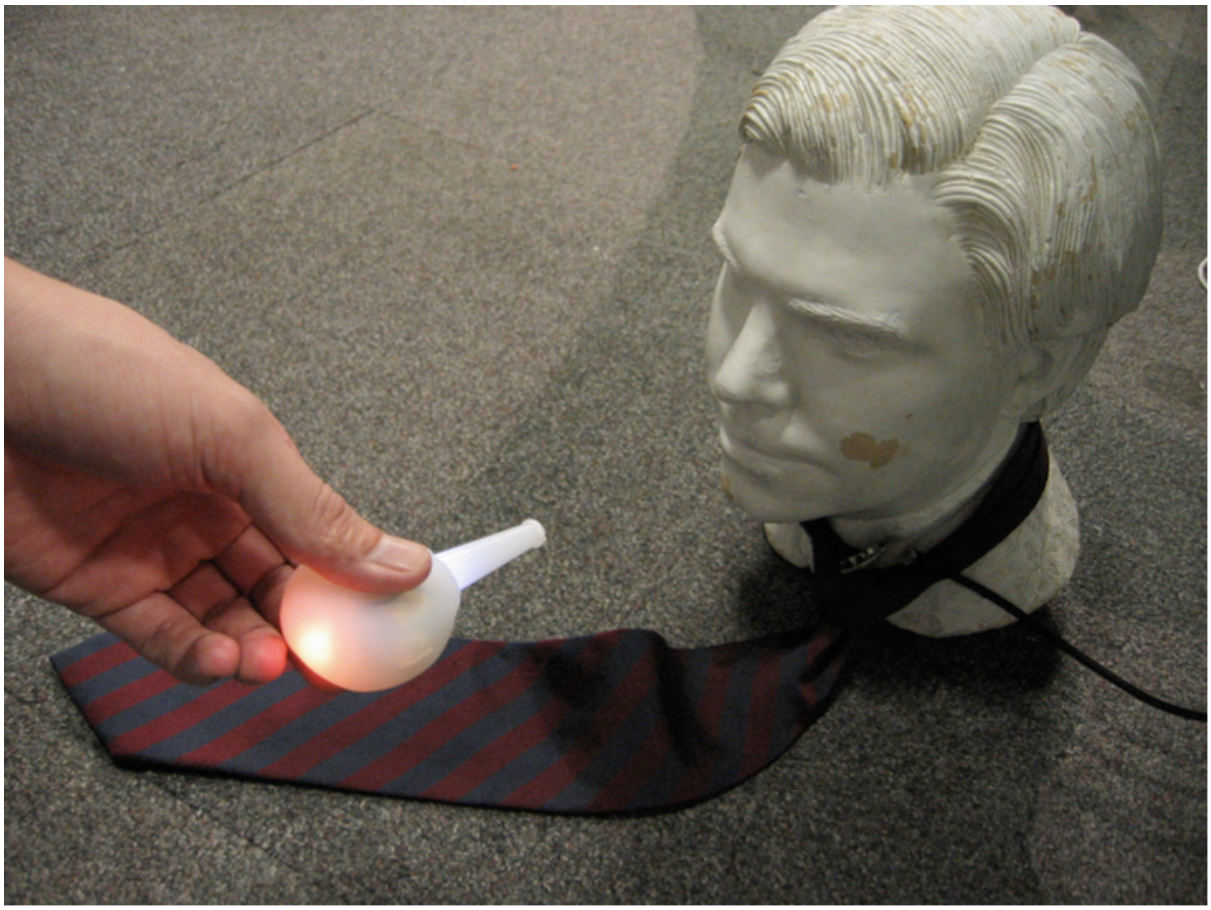

Figure 2.8: Slurp extracting a digital object from a sculpture.[9]

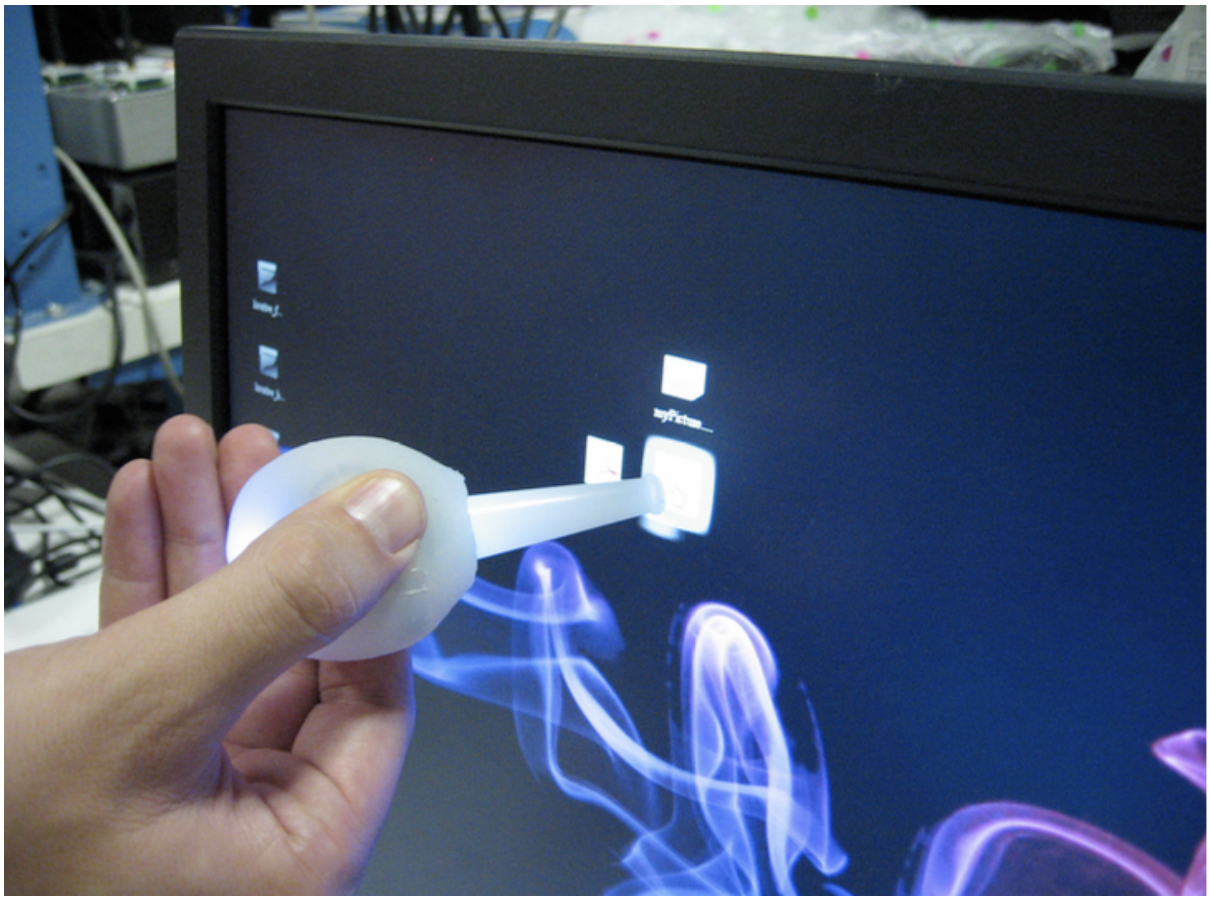

Figure 2.9: Slurp injecting a digital object onto a screen.[9]

Evaluation Results

The Slurp system went through various evaluation scenarios in order to demonstrate its ability to interact with both physical and digital objects. Users were instructed to use Slurp in various scenarios, such as extracting a video from a physical sculpture and injecting said video onto a digital display. These tasks aimed to test the system's practicality, versatility, and ease of use. Participants immediately found Slurp easy to use due to its resemblance to the traditional eyedropper. However, they still reported difficulties, especially when it came to touching Slurp to target objects, as that action requires immense precision and was also prone to accidental activations if a user wasn't paying enough attention.

2.1.8 Combining Gaze and Touch

Eye Pull, Eye Push: Techniques Combining Gaze and Touch for Content Transfer Turner et al. developed "Eye Pull, Eye Push" [6] which combines eye tracking and touch for moving content between devices. This method makes moving content more

natural by utilizing how eyes and hands move in unison. It involves three main techniques: Eye Cut Paste, Eye Drag Drop, and Eye Summon Cast.

System Implementation

Eye Pull, Eye Push system allows users to transfer content between large screens and personal devices using a combination of gaze and touch. For 'Eye Cut and Paste,' the user would look at the object he or she wishes to cut, then simultaneously tap their device to cut it. Then they would direct their gaze towards their device and tap again to paste it. 'Eye Drag and Drop' works in a similar manner, but instead of just tapping their device, users would hold their touch to start dragging the object, then while holding their touch, direct their gaze towards where they want to drop said object. Finally, in 'Eye Summon and Cast,' the user would look at an object, then swipe down on their device to summon it, then look at where they wish to send the summoned object and swipe up on their device.

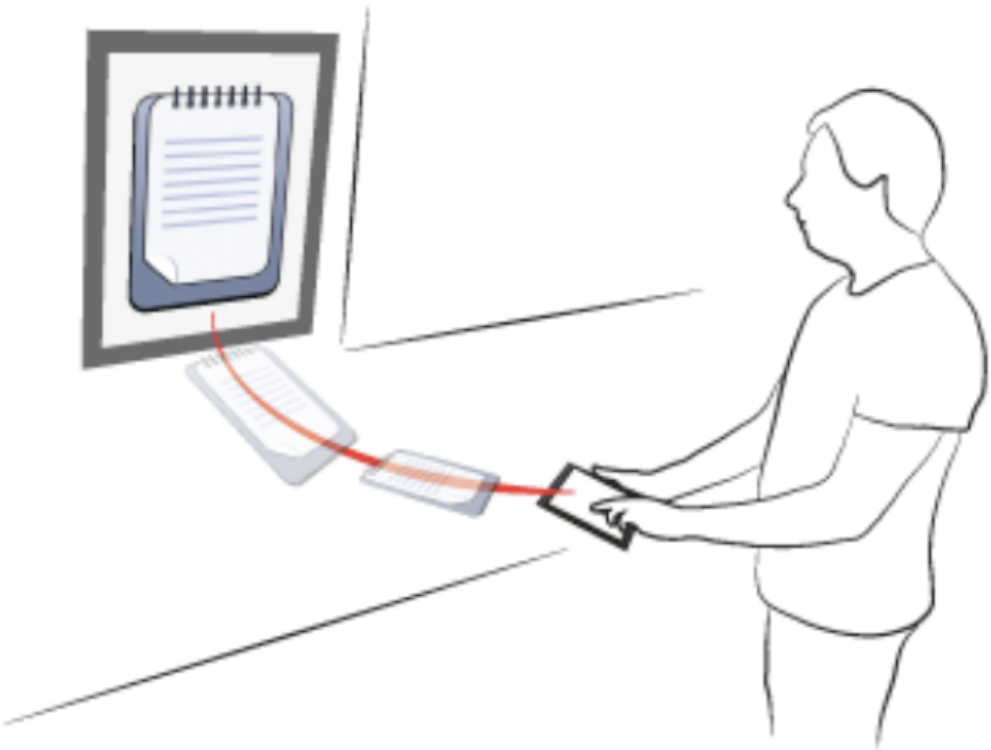

Figure 2.10: Figure 2.10: Eye Pull, Eye Push: users pull and push objects between remote screens and their personal devices with a combination of gaze and touch. In this scenario, the user selects a form on a public service terminal simply by looking at it, retrieves it to their touch device with a swipe, fills it in, and returns it with a swipe while looking up at the terminal.[6]

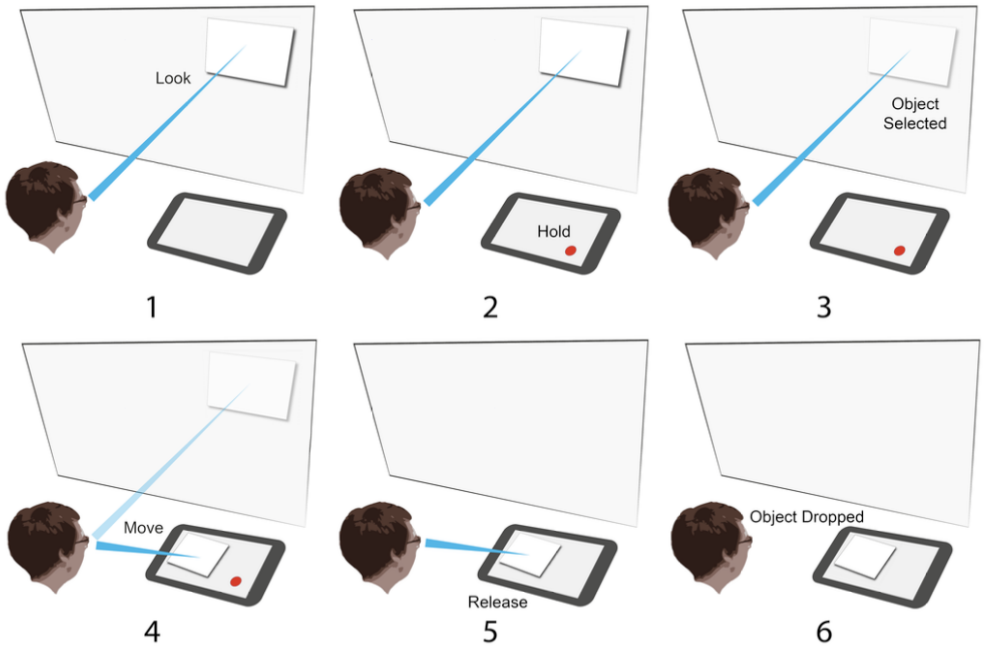

Figure 2.11: Figure 2.11: Eye Drag Drop: 1) Look at object, 2) Hold touch on tablet, 3) Object is selected and can be visibly moved, 4) Look at tablet, 5) Release touch from tablet, 6) Object is dropped.[6]

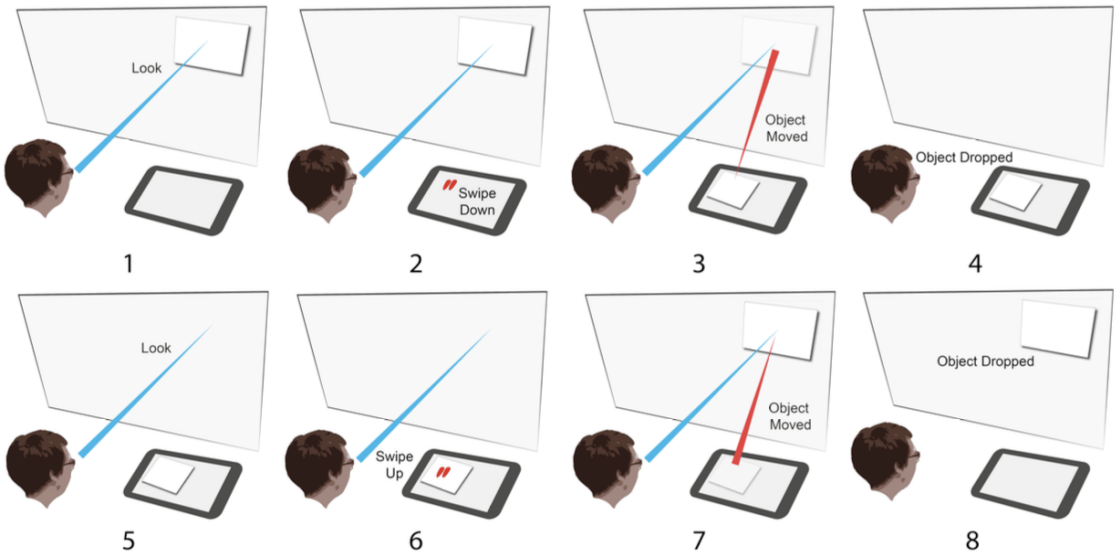

Figure 2.12: Figure 2.12: Eye Summon Cast. To summon: 1) Look at object, 2) Swipe down on tablet, 3) Object is moved to swipe location, 4) Object is dropped. To cast: 5) Look at destination, 6) Swipe up on object, 7) Object is moved to location of gaze, 8) Object is dropped.[6]

Evaluation Results

The Eye Pull, Eye Push system went through various user tests and evaluations to assess the usability and efficiency of its techniques. These tests indicated that 'Eye Summon and Cast' was the fastest out of the three interaction techniques, but it led to more errors on average due to its need for complex hand-eye coordination. On the other end of the spectrum, Eye Cut and Paste had the fewest reported errors as it gave constant feedback; however, due to its need for multiple taps, it was on average the slowest interaction technique. Eye Drag and Drop was the technique with the most positive feedback from test subjects as it gave continuous feedback and was relatively simple. In contrast, Eye Summon and Cast was the least universally liked due to the need for complex hand-eye coordination previously discussed.

2.2 Summary

In this chapter, we discussed related works, showing various new techniques in HCI. These works were grouped into categories like methods for reducing physical strain, gesture-based data transfer, improving accessibility, better desktop management, tangible interaction, and combining gaze and touch. The comparison showed the need to balance new ideas with practical use, focusing on user-centered design principles to make good interaction techniques.

Chapter 3

System Design and Implementation

3.1 Design Goals

The main objective of this system is to create a natural and easy-to-use alternative to the drag-and-drop system we all know. This is done through the use of natural hand gestures instead of a mouse and keyboard. These natural gestures include swiping, picking up, dropping, and touching. This makes the system feel very natural and easy to use for any new user.

Another goal is security and mobility, as the system uses a secure palm authentication system to log users in, ensuring your data is always safe and only you will be able to access it. This also means that cross-device drag-and-drop is a possibility, as you can log in to a device using your palm print, pick a file up, and place it in one of your phalanges. Then, go to another device and log in to said device, also using your palm, and drop the data you had just picked up from the first device onto the second device.

You may wonder, how is cross-device achieved? While it is understandable that one is able to log in to different devices, how exactly is data transferred between said devices? This is done through our integration with Google Drive services. When in network mode, when you place a file in any of your phalanges, this file also gets uploaded to the Google Drive cloud storage systems. This allows for easy retrieval of this file in order to place it on whichever device you are currently logged in on. When you drop a file from the cloud to a computer, this file is also deleted from the cloud, which ensures users' privacy as none of their files will ever permanently be on the cloud storage.

The system is also designed to be extremely responsive as it interprets and recognizes gestures in real-time, which ensures an extremely smooth user experience. It also boasts robust error handling to manage potential issues that may arise in gesture recognition, cloud storage management, local storage management, or layer management.

Overall, the design goals of the system focus on creating an intuitive, easy-to-use, and natural drag-and-drop experience for the user through an authentication system, cloud storage, real-time gesture recognition, and seamless layer data management.

3.2 System Overview

This system acts as an advanced gesture recognition platform which integrates multiple components to create a seamless user experience. The system uses MediaPipe in order to track users' hands and accurately detect gestures. The gestures are detected through the use of both machine learning and real-time geometrical analysis. This enables the system to interpret user gestures with high precision.

Once the user starts the program up, the first step is user authentication. The program captures the user's palm in order to authenticate said user against their palm print previously saved in the system. This system ensures that all your data is safely stored and only you, the user, can have access to it.

After successful authentication, the system interface is shown to the user. This interface includes two boxes that act as files that the user can store in their phalanges, the computer webcam, and their phone's cameras. The user always sees the window with the boxes and the feed of whichever camera detects hands at any given moment. Users interact with the system by performing gestures in real-time. Using the webcam, users can point at the box they want to select. When the system detects pointing, it spawns a cursor that moves dynamically depending on where the user is pointing. They can pick up the box by doing the appropriate gesture. They can drop the box as well by doing the drop gesture. They can switch from local to network mode, and they can, by swiping one hand over the other, switch from one data layer to another.

The smartphone camera is intended for detecting phalange touches in real-time as its position allows the user to have their palms in a natural position and not directed towards the webcam in an awkward way. The phone camera is pointing at the user's lap. Since phalange touches are detected by using real-time geometrical analysis, the use of a phone's camera is highly beneficial since the higher resolution allows for much more accurate calculations which helps in decreasing the amount of false positives and negatives as well.

Data storage and management are handled on Google Drive infrastructure as the user connects their Google Drive to the system. On the first ever start-up, the system creates a folder which includes inside it 12 folders, one for each phalange. Then inside each phalange folder is a number of folders based on how many layers the system is running on. For example, there will be two folders inside each phalange folder if the system is running a two-layer system. When the user places data in a phalange, then the file is uploaded to the indicated phalange's folder. Then when said user decides to drop data from a phalange, the data in said phalange's folder is deleted, making room for new data that the user may need to drag and drop.

The system also makes use of a data layer management system, as when the system detects a swiping motion of one hand over the other, the system switches to the next layer. This hierarchical data management system is crucial for managing large amounts of data, which makes our drag-and-drop system very flexible.

To conclude, our system makes dragging and dropping very intuitive as it uses natural hand gestures as well as secure palm authentication to ensure data integrity and safety. The layering system paired with the Google Drive integration makes our system, in theory, infinitely scalable.

3.3 Explorative Study on Spatial Memory and Phalange Placement

3.3.1 Study Design and Objectives

The objective of this exploratory study was to see if users would be able to utilize spatial memory to place and recall where they have placed files using our drag-and-drop gesture interaction system. The aim of the study was to test the effectiveness of using phalanges as reference points for file placement and to test the real accuracy as well as the recall speed of test subjects using our system.

Participants and Procedure:

A group of participants was asked to place 12 items on their phalanges. The items represented were "Rock Music, Project Reports, Vacation Photos, Birthday Videos, Fiction Books, Action Movies, Sales Presentations, Dessert Recipes, Lecture notes, Travel journal entries, Workout routines, and Graphic design files" I have included items that are similar in nature and items that are unique in order tp fully test user's spatial memory in regards to where they place each item. The items were the same for all test subjects involved to ensure fairness and accuracy. Then they were given 15 minutes so that they would make use of their spatial memory and not their short-term memory. Before the study, users were encouraged to make use of their spatial memory, for example, placing the bigger files on the bigger phalanges or developing a pattern so that, for example, files of similar or related nature were to be placed next to each other or in an arrangement that would be easy to remember for the test subject.

Spatial memory is a cognitive process that enables a person to remember things based on objects in their environment. For example, if you park next to a tree and you want to find your car, you will be looking for the tree you parked under and not your car. Spatial memory plays a very important role when it comes to where we place items. That is why it is so important in your system that users make use of their spatial memory, as it helps them create mental maps of where they placed each file.

After placing items on the phalanges, users were given 15 minutes, then they were quizzed on where they placed each item. The goal of the 15-minute wait is to simulate a realistic time gap between the placement and the recall of items. The recall accuracy as well as recall time was recorded for each participant.

3.3.2 Results

The results of the study provided valuable insights into the effectiveness of using spatial memory and phalanges for item placement and recall.

On average, each participant recalled about 7 of 12 items correctly, which indicates a recall accuracy of almost 58.3 percent. While this shows moderate effectiveness, it also highlights the need for potential improvements in the system design and user training. Bearing in mind the average was skewed as some users scored extremely low as they admitted to trying to use their short-term memory rather than their spatial memory.

As for recall time, the average user recall time for each item was 0.8 seconds, which indicates that participants could efficiently use their spatial memory to locate the items, despite the moderate accuracy.

To enhance the usability and effectiveness of the system, several improvements can be considered. They include but are not limited to enhanced training wherein we should train users to be able to better utilize their spatial memory, which in theory should lead to a major increase in recall accuracy for our system, as well as the inclusion of visual aids such as visual cues where users can peek at what each phalange holds by showing their palm to the camera could almost eliminate any false recalls in theory.

3.3.3 Discussion and Conclusions

The study's findings suggest that while participants were able to use spatial memory to recall item placements with moderate accuracy, there is significant room for improvement. The use of spatial memory strategies appears to be beneficial, but not all participants employed these strategies effectively. This indicates a need for better user training and possibly enhancements to the system to support more intuitive in regards to spatial memory usage.

Understanding Spatial Memory in Context:

Spatial memory allows users to create mental maps to easily recall where they have placed items. In this study, participants used their ability to create mental maps to map out where they have placed each item on their phalanges. The variability in recall accuracy indicates that individual differences in the ability to use spatial memory have a big effect on the results.

Future Research:

Further research is needed to explore the spatial memory strategies that are most efficient for the largest group of people, as well as incorporating sensory cues such as haptic feedback or sensory cues.

Conclusion:

The explorative study proves that while spatial memory can be used effectively for file placement and recall on phalanges in our gesture-based drag-and-drop system, there are

several improvements that can be incorporated, such as user training and sensory cues. By understanding and addressing the factors that influence spatial memory effectiveness, the system can be made much more effective, ultimately improving the user experience and accuracy of file recall.

3.4 Interaction Design

The system design uses fingers and natural hand movements for different tasks. Fingers have three parts: tip, middle, and base. This helps the system know where you are pointing. The system has some gestures set up for different actions. For example, the point gesture selects a box on the screen. The system shows a cursor that moves where you point. Select gestures (Select 1 and Select 2) are used to pick up a box. Select 1 starts picking up, and Select 2 completes it, like grabbing something. Drop gestures (Drop 1 and Drop 2) are used to release a box. Drop 1 starts the drop, and Drop 2 completes it, like letting go of something.

Touching phalanges means using one hand to touch certain parts of the fingers on the other hand. The system detects this with geometric analysis. This gesture lets users place or pick up data at specific finger parts, making interactions more precise and useful. Also, there is a swiping gesture to switch between layers. By swiping one hand over the other, users can move through different data layers easily, improving the system's use and experience.

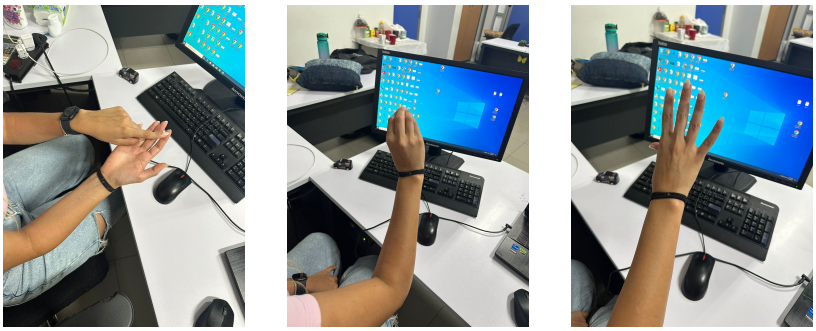

Figure 3.1: Process of selecting a file from one of phalanges and dropping it on to the desktop environment (Left to Right)

3.4.1 Interaction Techniques

The system's interaction methods use machine learning for recognizing gestures and geometric analysis for precise touch detection. The point gesture, which selects a box, involves detecting 21 hand landmarks with Mediapipe. It normalizes their coordinates based on the hand's bounding box and predicts the gesture with a pre-trained machine

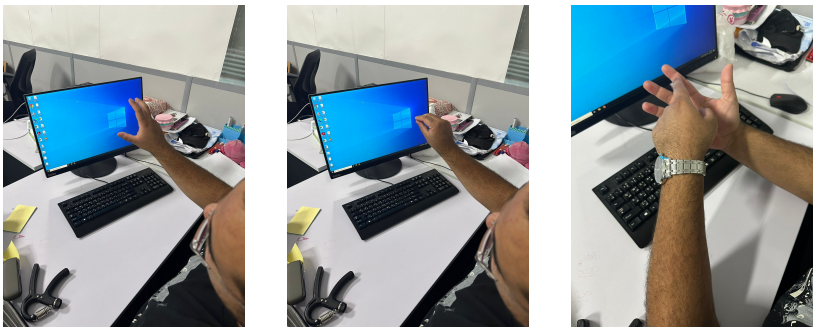

Figure 3.2: Process of selecting item from desktop environment and dropping it one of the phalanges (Left to Right)

learning model. If the gesture is recognized as a point, the system updates the cursor position to match the average position of the detected landmarks.

The select gestures (Select 1 and Select 2) are performed sequentially to pick up a box, simulating the physical act of grasping an object. The machine learning model predicts these gestures based on the normalized landmark coordinates. When Select 1 is followed by Select 2, the system identifies this sequence as a command to pick up a box, visually highlighting the selected box.

Touching phalanges uses geometric analysis. The system detects both hands and their landmarks with Mediapipe and measures the distance between the left hand's index fingertip and the right hand's phalanges. If this distance is below a set threshold, it means a touch. The system then determines the gesture, like placing or picking up data at that specific phalange.

The swipe gesture for switching layers involves detecting both hands and calculating their movement. If one hand moves over the other in a specific way, the system recognizes it as a swipe gesture and changes to the next layer.

3.4.2 Implementation

The Human Hand and Phalanges

The human hand can do many complex movements. Each hand has 14 finger bones, called phalanges. Most fingers have three bones (proximal, intermediate, and distal), but the thumb has two bones. This structure allows for many gestures, making it good for gesture-based interactions.

Our system uses the 12 finger bones (not counting the thumb) to store and manage data. Each bone can hold many files, using the hand's natural ability for layered and precise movements. This idea gives a new, simple, and efficient way to interact with digital content. The special structure of the hand allows for many gestures that can be known and used for different commands on a computer.

Layered Data Storage and Retrieval

Layered data storage means being able to store multiple items within a single gesture or action, organized in layers. This technique is very important for making data manipulation more efficient in a gesture-based system. In our project, each phalange can hold more than one file, allowing for a hierarchical storage structure.

The challenge with layered data storage is making sure users can remember and efficiently access the layers. We are doing experiments to find the best number of layers that users can remember and manage effectively. This involves testing different setups and checking user performance and recall abilities. Understanding the limits of human memory in this context is very important for designing an effective and user-friendly system.

Machine Learning

Machine learning plays a huge role in our system when it comes to accurately detecting single-handed gestures. I found an existing machine learning code online [4], which I then trained with a dataset of 100 pictures for each gesture. Mediapipe extracted the 21 features per hand and fed them to the machine learning algorithm. The model used Random Forest Classifier. The result was a highly capable model, very much able to detect any of the four single-handed gestures consistently.

Technical Processes

The system uses algorithms and technical processes in order to detect gestures and interactions. It heavily utilizes Mediapipe's hands solution for detecting and tracking hands in real-time. The solution easily detects and handles single-handed gestures as well as gestures with both hands. It captures and converts each video frame into RGB format for the most accurate processing. All hand landmarks are then drawn on the detected hands, which heavily aids in debugging and usability for the end user or developer.

The hand detection algorithm is highly accurate and precise, as it uses a detection confidence coefficient of 0.75 and a tracking confidence coefficient of 0.5. These confidence levels aid in ensuring that real-time gesture recognition is always reliable. This allows the system to easily track and understand hand movements, even in the most dynamic environments.

Machine learning is used to detect gestures, and geometric analysis is used for dynamic touch detection. The machine learning model is trained on labeled gestures. It detects 21 landmarks per hand, including joints, palm, and wrist.

Training the model involves collecting a large dataset of labeled hand gestures, normalizing and flattening the landmark coordinates, and training a machine learning model to classify the gestures. The model's performance is evaluated and fine-tuned to improve

accuracy, ensuring reliable interaction. A well-trained model can recognize gestures with high accuracy, providing a robust and intuitive user experience.

Geometric analysis is used to detect phalange touches and layer switches. Since Mediapipe is able to detect landmarks of the hand, we can continuously calculate the distance, using the x, y, and z coordinates, between the left fingertip and all phalange landmarks of the right hand. So, whenever the distance between the left fingertip and any of the phalanges is less than a predetermined threshold, a touch is registered, and an action takes place, whether it be placing an item in said phalange or picking up an item. A layer switch happens whenever a swipe is detected, so the program is always on the lookout for a lateral movement of one hand over the other, and once detected, the layer switch occurs.

3.5 Flowchart Analysis and Detailed Interaction Process

3.5.1 Advanced Interaction Flow: Two-Handed Phalange Touch Detection and File Management

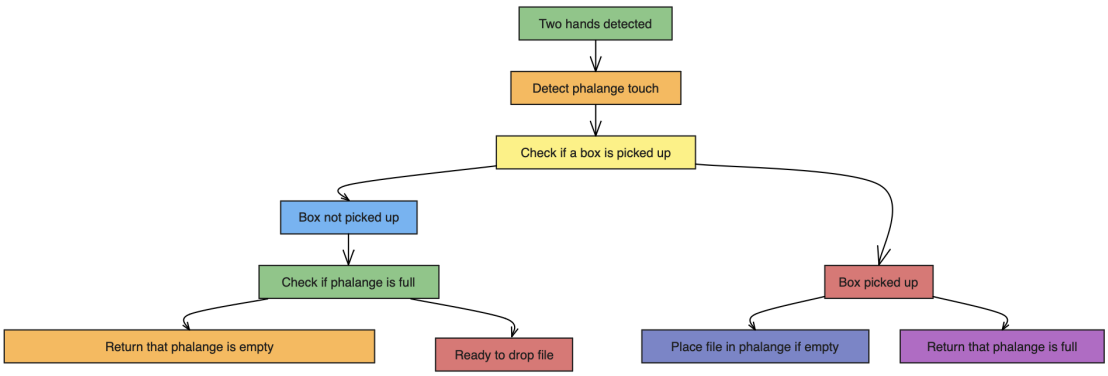

Figure 3.3: Flowchart for Two-Handed Phalange Touch Detection and File Management

The provided flowchart provides a breakdown of the interaction states of our drag-and-drop gesture-based system when two hands are detected by the camera feed. The flowchart outlines all the processes, including but not limited to phalange touches and placing and retrieving files.

The process begins once the camera feed detects two hands simultaneously, made possible through the use of MediaPipe's Hands solution, which provides and tracks 21 landmarks per hand. This hand detection protocol can detect and track hands continuously in real time, ensuring constant reliable identification of hand landmarks. Additionally, the high detection threshold in the source code ensures that only crystal-clear detections are processed, decreasing the number of false positive detections.

Once the system detects two hands is starts to attempt the detection of phalange touches it does so by continuously in every frame calculating the Euclidean distance between the index fingertip of the left hand and the phalange landmarks on the right hand the system keeps calculating said distance infinitely and once the distance is less than a certain predetermined threshold the system registers a touch. this precise detection is crucial in giving the end user a seamless and smooth experience.

Once the system detects a phalange touch, it starts checking whether or not a box has been picked up by the user. The system always maintains a state that shows whether or not an item is picked up and ready to be placed in a phalange. The system uses actions like 'select 1' followed by 'select 2' to indicate that an object has been picked up. This system ensures that the user always has a smooth, error-free experience.

If no box is currently picked up then the system checks for the status of the touched phalange. If said phalange is full, meaning it already contains a file, then the system indicates that the data in this phalange is now ready to be dropped onto the operating system. This check is made by checking the 'finger storage' array if we are in local mode or checking said phalange's folder in the cloud if we are in network mode. If the phalange is empty, then the system returns to the user an error message indicating that the selected phalange is empty, notifying the user that maybe he has touched a phalange other than the one he was supposed to touch.

If the system detects that a box has been picked up, then it goes to check the status of the selected phalange. If the phalange is full, then the system returns an error message to the user, indicating that the chosen phalange is full and unable to accept new data. However, if the phalange is empty, then the system updates its state to indicate that now there is no box picked up, and it places the data onto the selected phalange. In the process, it updates the Google Drive folder of the phalange with the new data if the user is in network mode, or updates local storage if the user is in local mode.

The system's implementation involves initializing the hand detection model, setting appropriate thresholds for detection confidence to ensure reliable hand landmark tracking. The phalange touch detection logic uses geometric analysis to calculate the distances between specific hand landmarks continuously. State management is crucial, where the system switches states based on detected gestures. The interaction logic ensures that placing and retrieving files is as smooth as can be for the end user.

3.5.2 Single-Handed Interaction and Network Mode Operations

This flowchart dives into the single-handed gesture workflow when it comes to picking up or dropping files. It goes through the steps from Google Drive and local file upload to Google Drive and local file deletion.

The process starts by taking the hand using MediaPipe's hand tracking solution, which detects 21 landmarks in the user's hand, making gesture tracking in real time a possibility. This system is then configured to handle real-time gesture tracking for a smooth experience.

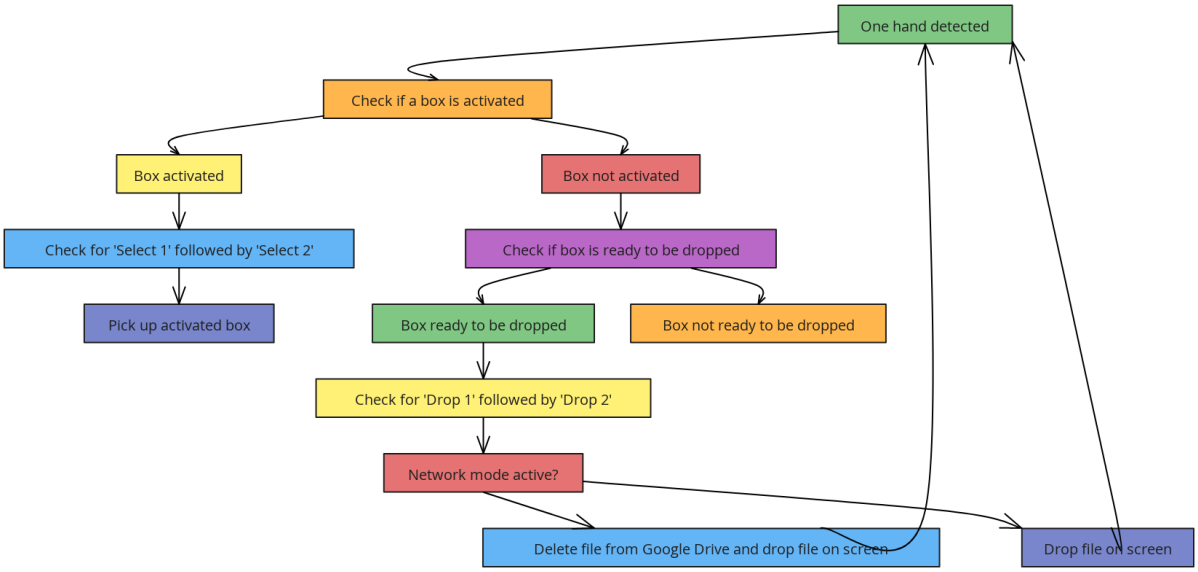

Figure 3.4: Flowchart for Single-Handed Interaction and Network Mode Operations

Once a hand is detected, the program checks to see whether a box is ready to drop or not. If a box is ready to drop, then the system is awaiting 'drop 1' followed by 'drop 2' in order to drop the data to the operating system and delete said data from local storage if in local mode, or Google Drive cloud storage if we are in network mode. However, if a box is activated, then the system will be awaiting 'select 1' followed by 'select 2' in order to pick up the box. When the box is picked up, it is then placed in any of the available phalanges using two-handed gestures.

The system initializes the hand detection and tracking model with specific confidence thresholds for accurate gesture recognition. The gesture recognition process involves using a pre-trained machine learning model to classify hand gestures and update the system state accordingly. The network mode operations leverage the Google Drive API for file uploads and deletions, ensuring secure and efficient file management.

3.6 Conclusion

The palm authentication and gesture recognition system use advanced computer vision and machine learning for an enhanced user experience. With natural hand gestures and precise touch detection, users drag and drop objects smoothly. They can select, move, and drop data with simple hand movements. The system's algorithms make sure gestures are recognized correctly for smooth interactions. The combination of machine learning and geometry provides precise and reliable gesture detection, making the system very reliable. Its strong design ensures accuracy and ease of use, giving users a natural drag and drop experience.

Chapter 4

Conclusion

4.1 Summary

This thesis tackled the concept of creating a brand new drag-and-drop-based interaction system using hand gestures, eliminating the need for a keyboard and mouse. The research problem centered around improving user interaction with digital files through the use of spatial memory and phalanges for file placement and recall. The proposed solution made use of advanced machine learning techniques as well as real-time geometric analysis built on Mediapipe's hands solution. The system was designed to be secure, incorporating palm authentication and cross-device functionality through Google Drive integration.

During development, several gestures were implemented such as pointing, selecting, dropping, swiping, and placing. The idea of leveraging layers and spatial memory was also experimented with. An experiment was conducted to test users' ability to use spatial memory to recall where they have placed files in reference to their phalanges. The study found that test subjects were able to recall file placements with moderate speed and accuracy, which shows potential but also highlights the need for enhanced user training and the introduction of special hardware to the interaction technique for improved performance.

4.2 Limitations

Albeit the very promising potential of the interaction, there are several limitations identified. First of which is the use of standard consumer webcams, as they provide a less than ideal experience in terms of smoothness mainly due to their hardware limitations as they lack depth sensors. Said sensors would greatly enhance the user experience due to their enhanced gesture recognition accuracy. Additionally, some users were unable to utilize their spatial memory, which led to less than stellar recall accuracy. These issues can be attributed to interaction constraints as there is a lack of visual cues that aim to help the

user remember where he placed the file, as well as an individuality issue since users who were able to utilize their spatial memory performed relatively well when it came to file recall.

While current research provides some solutions, such as depth cameras for improved accuracy, there are still open research problems in optimizing gesture recognition systems for diverse user interactions and environments. The integration of more sophisticated hardware and advanced machine learning models could potentially address these limitations.

4.3 Future Work

With time, better hardware, and advancements in software, several improvements could be made to the system. Firstly, integrating depth cameras instead of using webcams would greatly enhance the average user experience since, due to their ability to see in a three-dimensional space, they can recognize complex gestures much more efficiently than consumer webcams.

Advancements in machine learning technology would allow the system to dynamically adapt and learn each user's individual movements, behaviors, and quirks, making the system much more personalized. Additionally, enhanced user training, where the users are trained on utilizing their spatial memory to its full potential, would greatly enhance the system's average recall accuracy.

Integrating sensory cues such as haptic, visual, or auditory cues could also further enhance the usability and accessibility of the system, as their inclusion would help users better recall where they placed their files, reducing the reliance on users' memory.

This chapter provided a comprehensive conclusion of the thesis, summarizing the research problem, proposed solution, and the results achieved. It discussed the limitations of the current system, particularly the challenges posed by hardware constraints and variability in user performance. Finally, it outlined potential future work, suggesting improvements in hardware, machine learning models, and user training to enhance the system's effectiveness and user experience.

Appendix

List of Figures

2.1 Older adult playing a tactile puzzle on a tablet. This figure demonstrates the practical application of touch-based interactions which can be extended to large displays. [1] 4

2.2 Intuitive information transfer techniques with Toss-It. (a) from a PDA to another PDA (b) from a PDA to a printer (c) from a PDA to multiple PDAs. [8] 5

2.3 SPARSH – Touch to Copy, Touch to Paste.[5] 6

2.4 Moving a video link from a mobile screen to a larger digital display.[5] . 7

2.5 Identifying user by using a unique gestural sign.[5] 7

2.6 Figure 2.6: Rolling windows to reveal an overlapped one.[3] 10

2.7 Background recognizability task (alpha-50 condition). A source image with overlaid palettes is shown at the top left, and three candidate images—near copies of the source image—are shown in the other three quadrants.[2] . . 12

2.8 Slurp extracting a digital object from a sculpture.[9] 13

2.9 Slurp injecting a digital object onto a screen.[9] 14

2.10 Figure 2.10: Eye Pull, Eye Push: users pull and push objects between remote screens and their personal devices with a combination of gaze and touch. In this scenario, the user selects a form on a public service terminal simply by looking at it, retrieves it to their touch device with a swipe, fills it in, and returns it with a swipe while looking up at the terminal.[6] . . 15

2.11 Figure 2.11: Eye Drag Drop: 1) Look at object, 2) Hold touch on tablet, 3) Object is selected and can be visibly moved, 4) Look at tablet, 5) Release touch from tablet, 6) Object is dropped.[6] 16

2.12 Figure 2.12: Eye Summon Cast. To summon: 1) Look at object, 2) Swipe down on tablet, 3) Object is moved to swipe location, 4) Object is dropped. To cast: 5) Look at destination, 6) Swipe up on object, 7) Object is moved to location of gaze, 8) Object is dropped.[6] 16

3.1 Process of selecting a file from one of phalanges and dropping it on to the desktop environment (Left to Right) 23

3.2 Process of selecting item from desktop environment and dropping it one of the phalanges (Left to Right) 24

3.3 Flowchart for Two-Handed Phalange Touch Detection and File Management 26

3.4 Flowchart for Single-Handed Interaction and Network Mode Operations . 28

Bibliography

[1] P. Baudisch et al. Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-Operated Systems. *International Conference on Human-Computer Interaction INTERACT*, 2003.

[2] P. Baudisch et al. Multibending: displaying overlapping windows simultaneously without the drawbacks of alpha bending. *Conference on Human Factors in Computing Systems (CHI)*, 2004.

[3] O. Chapuis et al. Copy-and-Paste Between Overlapping Windows. *Conference on Human Factors in Computing Systems (CHI)*, 2007.

[4] computervisioneng. sign-language-detector-python. https://github.com/computervisioneng/sign-language-detector-python, 2023.

[5] P. Mistry et al. SPARSH: Passing Data using the Body as a Medium. *CSCW: Computer Supported Cooperative Work*, 2011.

[6] J. Turner et al. Eye Pull, Eye Push: Moving Objects between Large Screens and Personal Devices with Gaze and Touch. *14th International Conference on Human-Computer Interaction INTERACT*, 2013.

[7] F. Winberg et al. Designing Accessible Auditory Drag and Drop. *Conference on Universal Usability, CUU '03*, 2003.

[8] K. Yatani et al. Toss-It: intuitive information Transfer Techniques for Mobile Devices. *Conference on Human Factors in Computing Systems (CHI)*, 2005.

[9] J. Zigelbaum et al. Slurp: Tagibility, Spatiality, and an Eyedropper. *Conference on Human Factors in Computing Systems (CHI)*, 2008.