Media Engineering and Technology Faculty
German University in Cairo

Speed as a Input Channel

Bachelor Thesis

Author: Abdelrahman Khaled
Supervisors: Dr. Wael Abouelsaadat
Dr. M A Moustafa Hassan

Submission Date: 19 May, 2024

Media Engineering and Technology Faculty
German University in Cairo

Speed as a Input Channel

Bachelor Thesis

Author: Abdelrahman Khaled
Supervisors: Dr. Wael Abouelsaadat
Dr. M A Moustafa Hassan

Submission Date: 19 May, 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

---

Abdelrahman Khaled
19 May, 2024

Acknowledgments

After praising and thanking God for his guidance and supervision over the project, I would want to thank Dr. Wael Abouelsaadat first. He provided me with all the necessary tools and information and helped me through the entire process.Second, I want to express my gratitude to my entire family for being there for me in my time of need and for their unwavering support.Thank you to Ahmed M. Ayman, Abdelrahman Moghazy, and Ibrahim El Seoud. especially for your ongoing assistance. Lastly, I would want to express my gratitude to all of the participants for trying the experiment and sending me the necessary results, as well as to my friends for their assistance and support.

Abstract

The study of how a person interacts with a computer system to accomplish a task is known as human-computer interaction, or HCI. Its objective is to improve the effectiveness and enjoyment of user-computer interaction.The field of Human-Computer Interaction is expanding and getting better all the time, and this includes mouse tracking and gesture input as user interface designers strive to comprehend how the human body interacts with computers. This thesis's primary goal is to determine whether the program can identify each user's speed by utilizing all of the user's necessary data. Ibrahim AboElSeoud's earlier ISpeed project has been modified for this thesis.

Contents

Chapter 1

Introduction

1.1 Motivation

The goal of many user interface designers is to meet the needs of the user. Additionally, as time passes, the user needs to change. Therefore, the goal of human-computer interaction is to handle every aspect of computer-related interaction that can please and impress the user. Since it is their only constraint when creating software or applications, the field of human-computer interaction can aid user interface designers in their understanding of human minds. The most appropriate interface can be created or improved by the user interface designers by utilizing both mental and physical abilities.

One of the things that should be taken into account while creating a user interface is speed. The speed at which a user interface is designed determines how quickly a computer can process input from users. Using a keyboard to type text or a mouse to click and drag is an example of user input. Seneler, Basoglu, and Daim (2009) conducted a study with 150 participants to determine which feature out of five features would be most preferred by a web user. The researchers were from Turkey and the United States. Adaptive behavior, content density, customization, minimal memory burden, and speed were the five features. 54.9 percent of those in attendance selected speed over all other elements, demonstrating the significance of quickness in user interface design.

The program may identify the user's speed and execute the action in accordance with it if the user can operate at a specific speed on purpose.Such a finding may result in new mouse applications for the program that Ibrahim Abo ElSeoud developed and later improved. since all computers come equipped with a trackpad or have the ability to connect an external mouse.

1.2 Aim of the project

The aim of the project is to know if a software can detect the user's speed produced by mouse cursor movement by giving the users data and can perform an action according to

that speed. If it is proved that the user's speed can be detected, Such an idea can lead to further enhancement to Human Computer Interaction field and can use speed as an input in the upcoming softwares.

Acronym With Citation [9, 5, 14, 15, 20, **?**, 4, 6, 13, 17, 10, 16, 12, 11] (AC2).

1.3 Structure of the Thesis

This Thesis includes a five Chapters.The Chapters are as follows:

• Chapter 1 is an Introduction to The Thesis and what The Thesis is try to accomplish.

• Chapter 2 is a background about project and discussing previous works related to this thesis.

• Chapter 3 presenting the previous work that has the same aim ,the modifications that are done to the previous works,and how the process done for collecting data.

• Chapter 4 is discussing and explaining the algorithm used in the project.

• Chapter 5 is the conclusion of the project and the work needed to be done in the future.

Chapter 2

Background

2.1 Overview

In this section, We are stating other previous research papers that have been made and are related to this thesis. These research papers cover Fitts' law , Acquisition of Expanding Targets , Facilitating Moving Targets Acquisition on Augmented Reality Devices , how long can you go with mouse resolution , LinearDragger technology and Leveraging Physical Human Actions in Large Interaction Spaces .

2.2 Fitts' law

Other prior research studies that have been done and are relevant to this thesis are listed in this section. These research papers address moving targets facilitation, expanding target acquisition, and Fitts' law is named by Paul Fitts [9], Leveraging Physical Human Actions in Large Interaction Spaces, Linear Dragger Technology, Acquisition on Augmented Reality Devices, How Long Can You Use Mouse Resolution? , and Fisheye Menus , here is Fitts' Law formula [14]

$$T = a + b \log_2\left(\frac{A}{W} + c\right)$$

Figure 2.1: Fitts' Law Formula [14]

Where T is the movement time a and b are empirically determined constants which depends on the device , c is a constant of 0,0.5,or 1. A is the distance (or amplitude)

of movement from start to target center and at last W is the width of the target , which corresponds to accuracy .

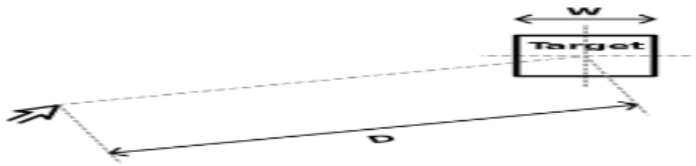

Figure 2.2: Fitts' Law

2.3 Machine Learning

One type of artificial intelligence (AI) is machine learning [5], which uses an algorithm derived from collected data to help a computer predict an outcome. In other words, the computer is trained to predict an event. Numerous algorithms are available for machine learning, and their use varies depending on the kind of data utilized for training. A wide range of applications use machine learning algorithms. Two examples are computer vision, which is used in robots or autopilot modes, and email filtering, which determines whether an email is spam or not. Decision trees and extra trees are two of the methods we employed in this experiment; these will be covered in Chapter 4.

Figure 2.3: Identifying Cars in an image

2.4 Acquisition of Expanding Targets

Some today's Softwares are now using fitts law in designing its targeting icon as in Figure 2.3 like Apple MacBook Softwares. The improvement in performance is dependent on

the final target size, not the initial one. That type of performance in expanding target is supported using a fitts law equation using a base values of data that contain no expansion at all. Michael McGuffin Ravin Balakrishnan [15] have investigated all the factors that affect the acquisition of expanding targets. They designed the experiment with three different expansion condition: Static, Which is the base case of fitts law to serve as a base case for comparison, Spatial expansion, which is expanding the target width to a certain width value over a certain period of time , and Fading-in expansion, Which is expanding the width over a certain period of time but visually fading over that period of time. They made different A-W combinations and divided the targets in two types , one increasing in Width over time and other decreasing in width.

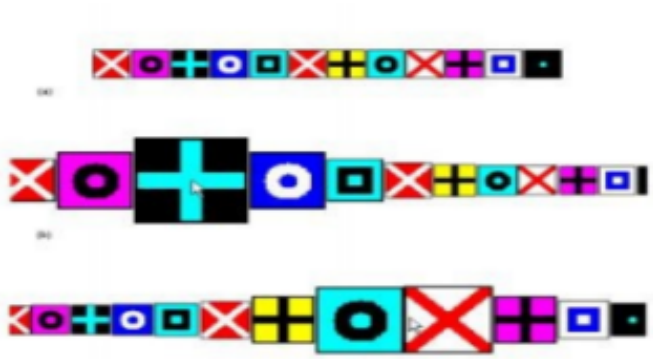

Figure 2.4: First prototype expanding widget design [15]

(a) When the pointer is far away, the buttons do not expand. (b) When the cursor is above a button, it expands entirely, and adjacent buttons are partially expanded and pushed sideways. (c) A user may attempt to travel to the right in order to choose the button with the light X on the dark backdrop if they are starting in the state depicted in (b). The button has shifted to the left and the user is now over a different button (one with a dark X on a light background) by the time the mouse reaches the intended button's location.

2.5 Facilitating Moving Targets Acquisition on Augmented Reality Devices

Fitts law is dependent on the width target, as previously mentioned. The shorter time required to choose or point on the target, the wider the target. However, we will find it difficult to choose the target if it is not static. According to research by Chuang-Wen You, Yung-Huan Hsieh, and Wen-Huang Cheng [20] there is a method that can let the user choose a moving target to increase error rate by at least 61.75 percentage while allowing the user to tolerate any acceptable distraction.

Figure 2.5: AttachedShock technique [20]

We refer to this method as AttachedShock. This method increases the target's width by creating a wave that propagates on the chosen target, making it easier to click or pick that target.

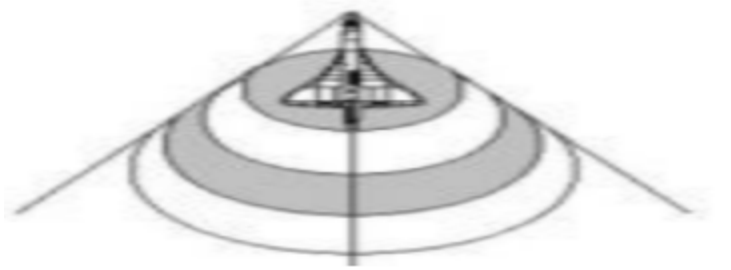

Figure 2.6: A real-world attached shock example [20]

2.6 How Low Can You Go

The resolution of computer mouse sensors is rising in the market these days. This implies that the smaller components the sensor can detect, the greater the resolution. It is unclear if the user will be able to manage these tiny motions, however research has been done on the subject. The smallest discernible change in input signal that results in an output signal change is known as the mouse sensor's resolution. Seldom is the length of such a movement detected. In an experiment to determine how high a resolution a user might achieve, Jonathan Aceituno1, Gry Casiez, and Nicolas Roussel [4] came up with 95 percentage as the cutoff for the acceptable creation of minor displacements and demonstrated that the dependable resolution is between 200 and 400 CPI. CPI, or counts per inch, is a unit of measurement used to describe mouse resolution.

Figure 2.7: Mouse Physical layout [18]

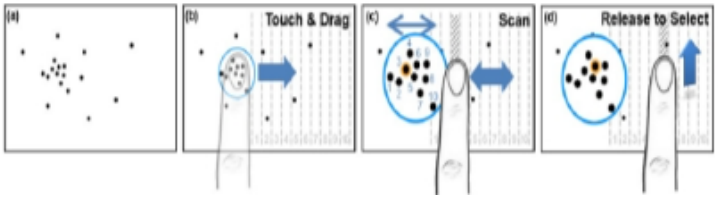

Figure 2.8: LinearDragger Process [6]

2.7 LinearDragger

These days, touch input is widely employed. Although it has a huge fingertip, it is not as exact as it may be, despite its overall efficiency. This study's experiment on LinearDragger, a novel and integrated one-finger target development approach for small and clustered targets, was conducted by Oscar Kin-Chung Au, Xiaojun Su, and Rynson W.H. Lau. When a user touches and drags on he touch screen, LinearDragger is activated, and the touched point establishes a contained user's desired goal. LinearDragger scans the chosen targets within the Region of interest one by one as long as the user drags his finger, with the starting drag direction dictating the scanning order. At last, the user lets go of his finger to choose the intended target. Such a method assisted in converting a touch-drag-release task from a 2D target selection problem to a 1D problem selection.

2.8 Leveraging Physical Human Actions In Large Interaction Spaces

To facilitate data exchange, an interface method that recognizes user gestures is created. Because it allows users to navigate and manipulate data at the same time, this technique is efficient. Can Liu [13] conducted an experiment to see how useful wall-sized screens are for navigation. This experiment showed that using users' natural movements as a guide is highly helpful for navigation. For instance, the user can simultaneously manipulate

data and zoom in and out by moving their head, whereas on a desktop, this can only be done one at a time due to the limited number of input devices. PoPle Prototype is an interaction technique designed to assess the efficacy of data sharing between two or more users of the gestures made by users in the same area. To sum up, these ideas are ideal in two key respects. First, the study demonstrates that wall-sized, high-resolution displays are superior to desktop screens. Second, efforts focused on creating prototypes will facilitate interactions between people that are connected to the ideas explored in the study.

Figure 2.9: Wall-Sized Display that are used in the experiment [13]

2.9 Fisheye Menus

Fisheye Menus," written by Benjamin B. Bederson [5]0introduces an innovative approach to interface design using fisheye graphical visualization techniques applied to linear menus. This concept enhances the functionality of menus, particularly useful for long lists increasingly common in e-commerce and other data-intensive applications. Fisheye menus adaptively change the size of menu items to focus around the user's mouse pointer, allowing for the entire menu to be displayed on a single screen without the need for traditional navigational aids like scroll bars, buttons, or hierarchical organization.

The paper details a pilot study involving ten users who compared fisheye menus with traditional scrolling and hierarchical menus. The study found that users generally preferred fisheye menus for browsing tasks due to their dynamic scalability and ease of locating items, while hierarchical menus were favored for goal-directed tasks where users needed to find specific items quickly.

Key to the fisheye menu design is its ability to make long lists manageable by dynamically adjusting item visibility and size based on the item's proximity to the mouse cursor. This not only improves user experience by reducing the need for excessive scrolling but also enhances the efficiency of menu navigation. The design issues discussed include the balance between focus length and font size, the utility of an alphabetic index, and adaptations like the "focus lock" mode that stabilizes the focus to facilitate easier selection of specific items.

Overall, "Fisheye Menus" contributes to the field of human-computer interaction by offering a novel solution to the challenge of navigating long menus, suggesting a blend of traditional and innovative techniques to improve user interface design.

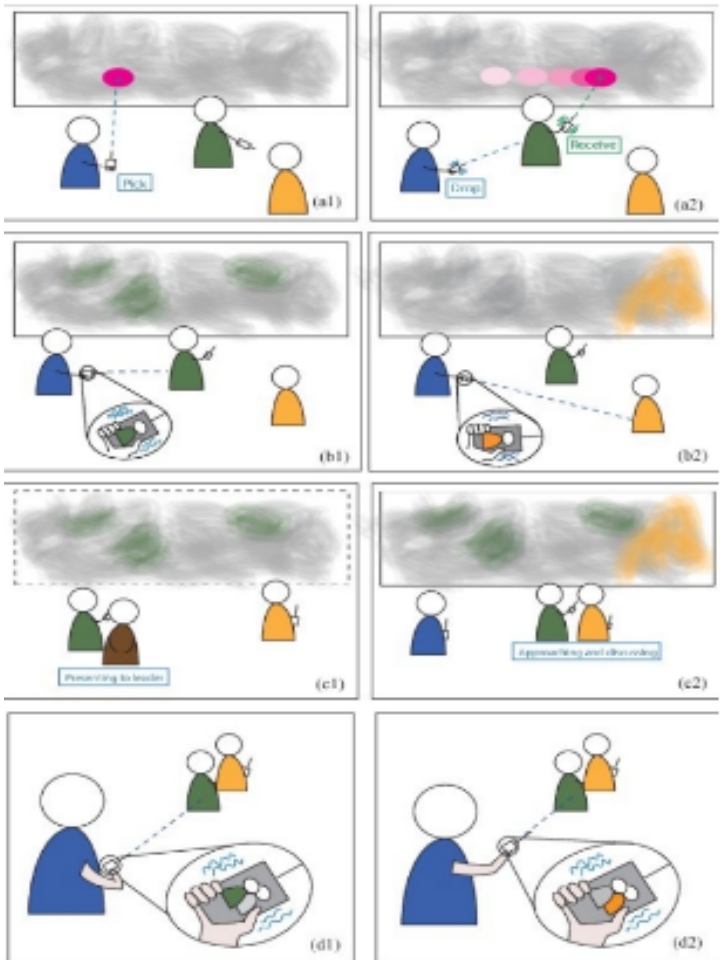

Figure 2.10: PoPle Prototype Example [13]

Leveraging Physical Human Actions In Large In-teraction Spaces

Background

Figure 2.11: Fisheye Menus Example [19]

Chapter 3

Data Collection

3.1 Previous Approach

In this chapter , We are going to discuss the previous works and approaches that were done by the previous students related to this thesis.The first experiment was made by Ibrahim El Seoud and Abdelrahman Moghazy and Ahmed M. Ayman, and the second experiment was made by Salman K. ElDash

3.1.1 ISpeed Project

A previous approach was developed by Ibrahim El Seoud. The experiment used three speeds , Slow , Medium , and Fast. Each of these speeds is categorized with numbers 0, 1 ,and 2 respectively. Each user is running an application written by C++ programming language using the OpenGL , Open Graphics Library. The application will open a command prompt window asking for users ID to be entered. And will begin to make a folder with a name of the user's ID. After entering the user's ID, a window opens that contains a green line and red line appears randomly in four opposite directions, for example : if the green line appears on top , the red line appears below it, with the name of the speed appears on the top of the window as it appears in figure 3.2.

will have a total of a 90 times trial. And for the three speeds for each direction , the user will have a 3*90 = 270 times trial, and for four different directions 270*4 = 1080 times trial in total. Between each 30 trials in each direction is 60 seconds break and between each direction is a 300 seconds break. A folders within the users ID folder is created with ID of each direction, the directions is categorized with number from 0 to 3 respectively, and with each direction folder , a folder with the speeds ID .The data should appear finally as a CSV file for the collected data for each trial inside the speeds folder as it appears in figure 3.3. Then, the speed should be calculated using an algorithm written in python by Abdelrahman Moghazy and the data will be trained by the Extra Trees algorithm as discussed in Chapter 4 Section 4.5, Although the Ibrahim El Seoud

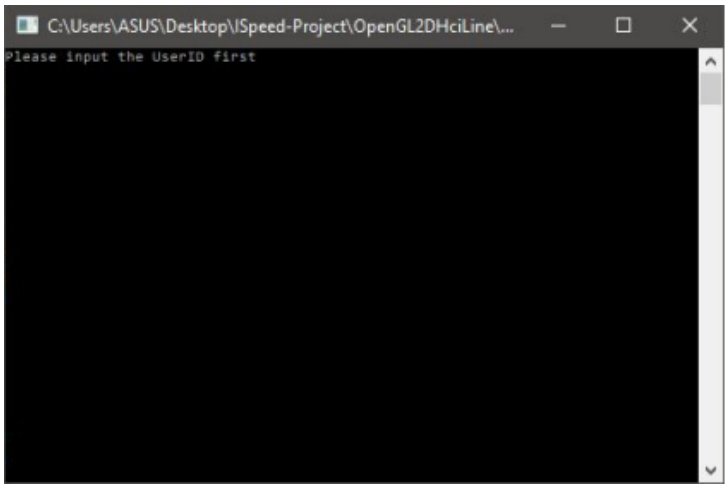

Figure 3.1: Command Prompt Entering User's ID [2]

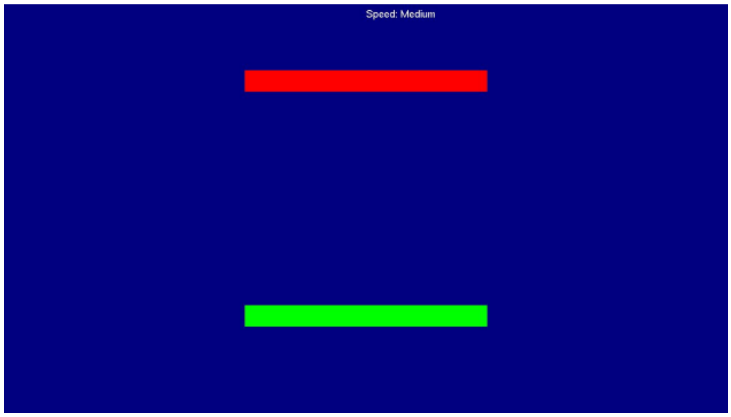

Figure 3.2: Green line and Red line window with speed at the top of the window [2]

experiment worked, Ahmed M.Ayman needed to get more accurate results. So, He added another speed to get a more accurate one. In the last ISpeed Project, The total speed the user will try on will be a total of four speeds, Very Slow, Slow, Fast, and Very Fast.Each one will be categorized by numbers from 0 to 3 respectively.So the total number of trials is 1440 trial as the number of trials for each speed in one direction is 90 trials and now we have 4 speeds and still 4 directions, so 90*4*4 = 1440 trials.And the same process goes on as stated in SubSection 3.1.1.

Column Description in the CSV Data file

- trialIdx is for the current trial's number.
- NumberInBlock is the number of the trial in the speed and direction of the trial.
- direction is the ID of the direction of the mouse cursor from green line to red line where 0 stands for the left to right , 1 stands for right to left , 2 stands for up to down , and 3 stands for down to up.
- speed is the speed's ID.

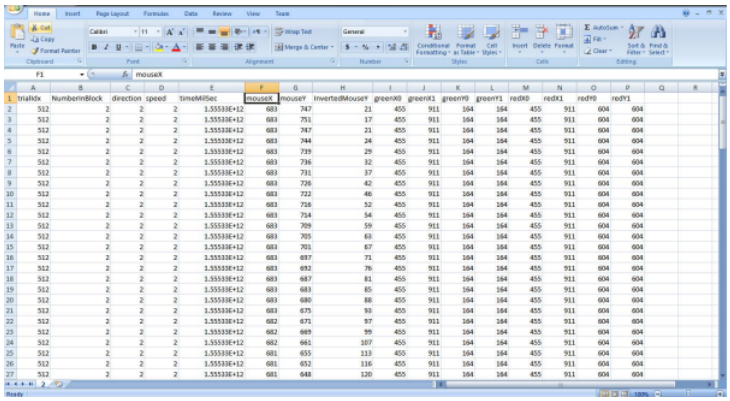

Figure 3.3: CSV datafile [2]

- timeMilSec is the time taken for each trial in milliseconds.
- mouseX is the position of the cursor of the mouse on the x axis.
- mouseY is the position of the cursor of mouse on y axis,windows origin is top left while openGL origin is bottom left.
- InvertedMouseY is the real y coordinates of the mouse which is consistent to the OpenGL lines.
- greenX0 is the start of the green line on the x axis without the thickness, and greenX1 is the end of the green line on the x axis without the thickness.The green line appears thick to be visible to the user.
- greenY0 is the start of the green line on y axis without the thickness, and greenY1 is the end of the green line on y axis without the thickness.
- redX0 is the start of the red line on the x axis without the thickness,and redX1 is the end of the red line on the x axis without the thickness. The red line appears thick to be visible to the user.
- redY0 is the start of the green line on y axis without the thickness, and redY1 is the end of the green line on y axis without the thickness.

3.1.2 Enhanced Directory Navigation

This experiment ,designed by Salman K. ElDash, was made to check whether people can differentiate between their own speed. Its whole idea is about measuring the speed of the mouse crossing the red line from the green line. Such a movement is done using six different orientations as in figure 3.4 (right to left, left to right, up to down, down to up, up left to down right, down right to up left). Another essential factor is the distance. The experiment measures the speed in three different distances, short (2.5cm), medium (8cm), and large (12.5cm). The participants in the experiment are divided into 4 groups. Group 1 has to test 2 speeds, group 2 has to test 3 speeds, group 3 has to test 4 speeds, and group 5 has to test 5 speeds. Once a participant selects a certain speed, he or she is required to test it in all orientations and distances, and a few seconds break will be made after each trial is finished.

3.2 Previous Approach

Figure 3.4: The six orientation [2]

Implemented Class

Classes that are implemented are:

• Experiment: The class where the experiment is simulated

• PartInfo: the class where the data of the participant is stored in

• Trial: the class that deals with configuration files

• StartTrial: the class that calls a new Frame (Experiment static variable) when a trial is finished

• Frame: the class that contains two components (Red and green) with their respective mouse listener events in a declared frame and handles the speed calculation.

• Stopwatch: a timer class used to calculate time taken for the mouse to propagate from the green to the red line

• EndExp: A thank you message left at the end

And Configuration file :

• Config.properties: The file that contains all the required trials a participant must finish

Strategy

The experiment was tested on 48 participants, divided into 4 groups with 12 participants each. Each participant had to test only on one speed group category .The Speed Groups are in figure 3.5 and 3.6 . To get more precise results, all the external factors had to be eliminated by turning off all applications and displaying the trail frame in full screen. Once a participant finished testing, data was recorded in text files recording detection. After the data was recorded, it was divided according to the group category separately and analyzed.In conclusion to this experiment,Users can differentiate between up to 5 speed variables.

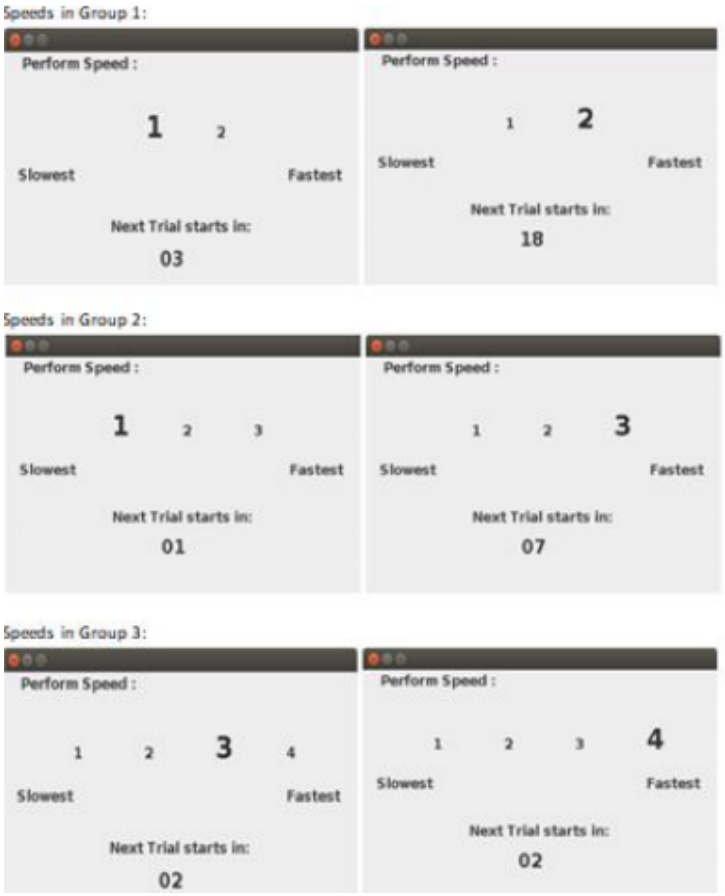

Figure 3.5: Speed Group 1,2,3 [2]

3.3 Modifications

3.3.1 Ispeed C/C++ Project

Although Ibrahim El Seoud and Abdelrahman Moghazy and Ahmed M. Ayman experiment worked, We needed to get more accurate results. So, We added another speed to get a more accurate one. In the updated ISpeed Project, The total speed the user will try on a total of five speeds, 1/5, 2/5, 3/5, 4/5 ,and 5/5 .Each one will be categorized by numbers from 0 to 4 respectively.So the total number of trials is 2400 trial as the number of trials for each speed in one direction is 120 trials since we added a new block to be 4 blocks each direction and now we have 5 speeds and still 4 directions ,therefore we will get 120 Trials *5 Speed *4 Blocks = 2400 trials , and decreasing the number of breaks given between each trial and between the end of the whole direction (1 directions / 4 directions) trial as some of the participants get bored of the whole 2 hours and 30 minutes. we made the seconds of breaks between each trial to be 30 seconds instead of 60 seconds and the number of seconds between the end of the whole direction to be 3 minutes instead of 5 minutes which give us from 1 hour to 1 hour 15 minutes to finish the

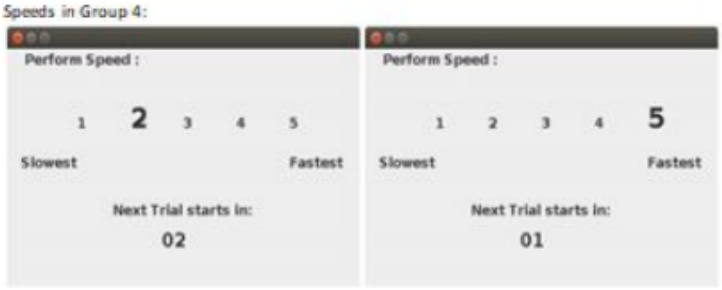

Figure 3.6: Speed Group 4 [2]

experiment ,and we added at the end of column files for each trial "user" which output in every line in final ".csv" files the user id to get better recognition of the data.

Although experiment Ibrahim El Seoud and Abdelrahman Moghazy ,Ahmed M. Ayman ,and Salman K. ElDash , was using simple experimnet interaction so we will develop a demo speed-based interaction by adding Support speed zooming interaction ,Support speed scrolling interaction ,and Support speed selection interaction

3.3.2 Salman ISpeed Java Project

Although experiment ,designed by Salman K. ElDash was used to test 4 different speeds for 4 different participant groups ,We added a new 2 speeds (6 and 7) so we can get Group 1 has to test 2 speeds, group 2 has to test 3 speeds, group 3 has to test 4 speeds, and group 5 has to test 5 speeds, group 6 has to test 6 speeds and group 7 has to test 7 speeds. Once a participant selects a certain speed, he or she is required to test it in all orientations and distances, and a few seconds break will be made after each trial is finished.

3.4 Performance

We have made the experiment on 7 university students, 7 males and 5 females.The User's age was between 19 and 23 years old.The setting of the experiment is a silent room in the university ,So nothing can interrupt the participants. All the participants were aware of the experiment and no one found difficulty in any of the trials. Each trial of the whole experiments took between One hour to One hours and 30 minutes .The User was using A Razer DeathAdder Elite ,like the one in figure 3.7,mouse as it have a better resolution rather any mouse in computer shops and was made of 5G sensor Razer that can handle accelerations up to 50 G while maintaining accurate tracking at 450 inches per second that can make the mouse movement much easier to the participants.All the data of each participant will be collected in a CSV as stated in SubSection 3.1.1.

Figure 3.7: Razer DeathAdder Elite [18]

Chapter 4

Machine Learning

4.1 Decision Tree

A type of supervised machine learning known as a decision tree requires explanation of the inputs provided by the data, and the corresponding output is found in the training data. The data is continuously split into a tree-like graph that answers questions found in the training data in order to predict potential outcomes. The decision is made once all the characteristics have been computed, with each branch signifying the test result and each leaf node representing a class label. The pathways from the root reflect the classification rule. A decision rule produces each leaf node. A decision rule, as shown in figure 4.1, is a conditional statement that poses a yes-or-no inquiry or An algorithm can classify the nodes to produce a tree-like model based on the response provided by the data in a question that can categorize the data into query scenarios [10], such as figure 4.1.

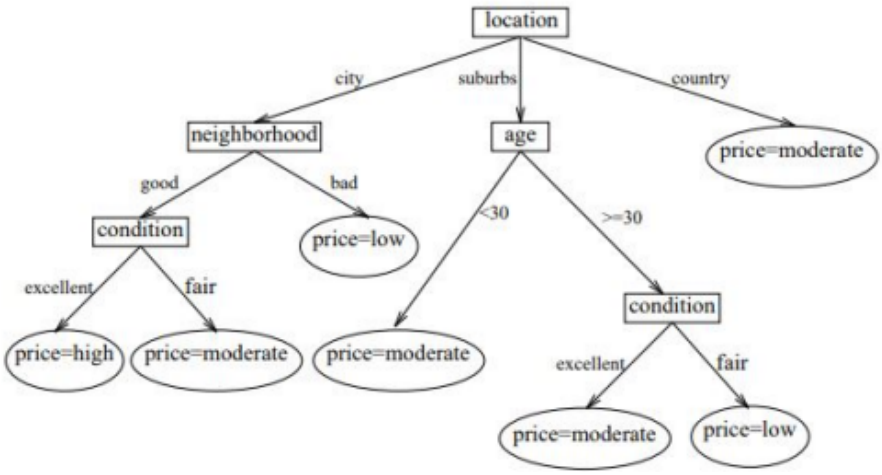

Figure 4.1: Decision Tree Example 1 [8]

Figure 4.2: Decision Tree Example 2 [8]

4.1.1 Entropy

Decision trees employ entropy to determine how similar a set of data is. Entropy influences the boundary-drawing process of Decision Trees and regulates their splitting. Entropy Formula [10] is as follows:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Figure 4.3: Entropy Formula [10]

The total number of classified groups in the original data is shown in Figure 3.1,c. The probability of the categorized group's number over the original data is represented by the symbol pi. Half of the original data has an entropy of 1, and the original data's entropy is equal to 0. A dataset with a high entropy indicates that we are unable to draw conclusions from its information.

4.1.2 Information Gain

Information Gain is one of the primary building blocks for Decision Tree construction. It is a gauge for the amount of knowledge that a class can impart [16]. The Information Gain formula is as follows:

The parent node, or node that is split up into child nodes, has an entropy of E(S). The total number of divided nodes is k. The divided node's entropy is expressed as E(S,Qi). Since it has more information, the high Information Gain attribute will be split first.

$$G(S,Q) = E(S) - \sum_{i=1}^{k} p_i E(S,Q_i)$$

Figure 4.4: Information Gain Formula

4.2 Why Decision Tree

There are numerous benefits to decision trees. One of them is that they are easy to read and comprehend, as evidenced by the fact that one need only glance at the tree chart and a quick explanation[12]. Even with little complicated data, it still has significance since specialists may provide crucial context by elaborating on the situation's alternative, costs, and probability. It can calculate every possible scenario, including the best and worst ones, and predict values for various situations. Finally, because of its simplicity and speed in providing findings and information, it can be used in conjunction with other decision-making processes.

4.3 Extra Tree

additional trees, Abbreviated as "extremely randomized trees," this approach is a sort of ensemble learning that builds numerous Decision Trees during the training phase, utilizing a multiple learning algorithm to improve predictive accuracy [11]. Instead of using the optimal decision tree, it uses a random one. With the exception of a bootstrap replica, it divides nodes by arbitrarily choosing a cut-point and grows the tree using all of the learning samples.

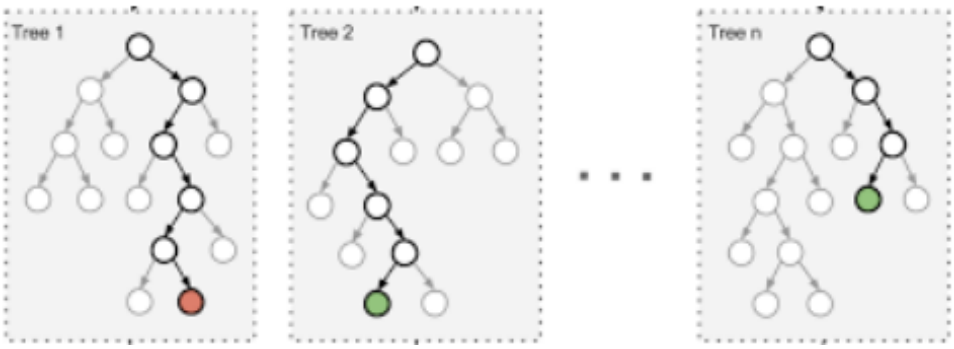

Figure 4.5: Extra Tree Diagram [7]

4.4 Why Extra Tree

Extra Tree minimizes the danger of overfitting, a circumstance where the details and noise of the training data are learned so well that it might adversely affect the performance of the real model. Extra Tree uses numerous Decision Trees. It performs well on massive amounts of data, such databases, and provides a high degree of prediction accuracy compared to Decision Trees because it combines several Decision Trees to reach a decision. Lastly, it will keep working and be able to estimate missing data from any Decision Tree.

4.5 Algorithm

There are four categories for speed: 0 represents very slow speed, 1 represents medium speed, 2 represents quick speed, and 3 represents very fast speed. The dataset that was gathered from every user will be used to train the algorithm. The speed will be divided into two datasets by the algorithm. One blank dataset for the variable y, and one for the features x. The dataset x will then be divided into two different sets by the algorithm: x train and x test. Likewise, the dataset will be divided into two sets by the algorithm: y train and y test. The dataset will be divided using the most popular data science technique, an 80-20 split.

```
from sklearn.model_selection import train_test_split
x = data.drop(['speed'], axis = 1)
y = data['speed'].astype("int32")
print( y.head() )

x_train, x_test, y_train, y_test = train_test_split( x.values, y.values, test_size=0.2, random_state=42 )
```

Figure 4.6: Splitting the data into training and test sets

After splitting, we ran the Extra Trees algorithm on the x train data and y train data.

```
clf = ExtraTreesClassifier(n_estimators=950, random_state=0)

clf.fit(x_train, y_train)
```

Figure 4.7: Extra Tree algorithm [7]

Chapter 5

Conclusion and Future Work

5.1 Conclusion

We collected information about every participant so the software could identify each of their five speeds, even though we are unable to distinguish between the five speeds. The algorithm created in Chapter 4 in section 4.5 has identified each participant's speed as the eight individuals have worked on the experiments by training on 80 percent of the data and testing on the remaining 20 percent. The test sets' accuracy is 97.78 percent, which is excellent since it indicates that we can identify each participant's five speeds. Our algorithm was able to predict the speed as accurately as possible with 97.78 percent prediction accuracy.

5.2 Future Work

In order to have a wider range of speeds, it would be advantageous to add one more speed, making a total of six. This project, which was completed by earlier researchers working with Dr. Wael Abouelsaadat, can assist in combining the speeds [3] in order to get more accurate findings.Reducing the amount of breaks between each direction trial and the next trial after the first trial ends, since some participants get weary of the entire hour or more. To make the experiment more effective, 12 more volunteers are welcome.The following features must be present on the device used for the same experiment:

• We can experiment with various hand-operated motions by utilizing the jump motion feature and the finger index touching functionality.

•able to be attached to an air mouse. It is now possible to create an application that can handle the user's various pace variations by crossing or moving the mouse as this previous PHD project tackle. [1]

Appendix

Appendix A

Lists

AC2 Acronym With Citation [9, 5, 14, 15, 20, **?**, 4, 6, 13, 17, 10, 16, 12, 11]

List of Figures

Bibliography

[1] Wael Abouelsaadat. *Mid-air Scrolling on Large Screens*. PhD thesis, GUC, 2018.

[2] Wael Abouelsaadat. *Speed as a Input Channel*. PhD thesis, GUC, 2019.

[3] Wael Abouelsaadat. *Machine Learning to Enhance DragDrop*. PhD thesis, GUC, 2020.

[4] Jonathan Aceituno, Géry Casiez, and Nicolas Roussel. How low can you go?: Human limits in small unidirectional mouse movements. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1383–1386. ACM, 2013.

[5] Ethem Alpaydin. *Introduction to Machine Learning*. MIT Press, 2009.

[6] Oscar Kin-Chung Au, Xiaojun Su, and Rynson W.H. Lau. Lineardragger: A linear selector for one-finger target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2607–2616. ACM, 2014.

[7] Abdelkader Berrouachedi, Rakia Jaziri, and Gilles Bernard. Deep extremely randomized trees. In *Advances in Intelligent Systems and Computing*, volume 1037, pages 487–497. Springer, 2019.

[8] Hendrik Blockeel, Laurens Devos, Benoît Frénay, Géraldin Nanfack, and Siegfried Nijssen. Decision trees: From efficient prediction to responsible ai. *Frontiers in Artificial Intelligence*, 6:1124553, 2023.

[9] Paul M Fitts and Michael I Posner. Human performance. 1967.

[10] Jerome H. Friedman, Ron Kohavi, and Yeogirl Yun. Lazy decision trees. In *AAAI/IAAI*, volume 1, pages 717–724, 1996.

[11] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

[12] Carl Kingsford and Steven L. Salzberg. What are decision trees? *Nature Biotechnology*, 26(9):1011, 2008.

[13] Can Liu. Leveraging physical human actions in large interaction spaces. In *Proceedings of the Adjunct Publication of the 27th Annual ACM Symposium on User Interface Software and Technology*, pages 9–12. ACM, 2014.

[14] I. Scott MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7(1):91–139, 1992.

[15] Michael McGuffin and Ravin Balakrishnan. Acquisition of expanding targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 57–64. ACM, 2002.

[16] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[17] J. Ross Quinlan. Simplifying decision trees. *International Journal of Man-Machine Studies*, 27(3):221–234, 1987.

[18] The best mouse in market - spring 2024.

[19] Xinyong Zhang Wenxin Feng, Hongbin Zha. Fisheye menus. *Benjamin B.Bederson*, 2000.

[20] Chuang-Wen You, Yung-Huan Hsieh, and Wen-Huang Cheng. Attachedshock: Facilitating moving targets acquisition on augmented reality devices using goal-crossing actions. In *Proceedings of the 20th ACM International Conference on Multimedia*, pages 1141–1144. ACM, 2012.