FoldAR: Gesture Analytics Using Apple Vision Framework Final Project Report

Ian Brown
Colorado State University,
Department of Electrical and Computer Engineering
Fort Collins, Colorado, United States
ian.brown@colostate.edu

Tani Cath
Colorado State University,
Department of Computer Science
Fort Collins, Colorado, United States
tani.cath@colostate.edu

Tom Cavey
Colorado State University,
Department of Computer Science
Fort Collins, Colorado, United States
tomcavey@colostate.edu

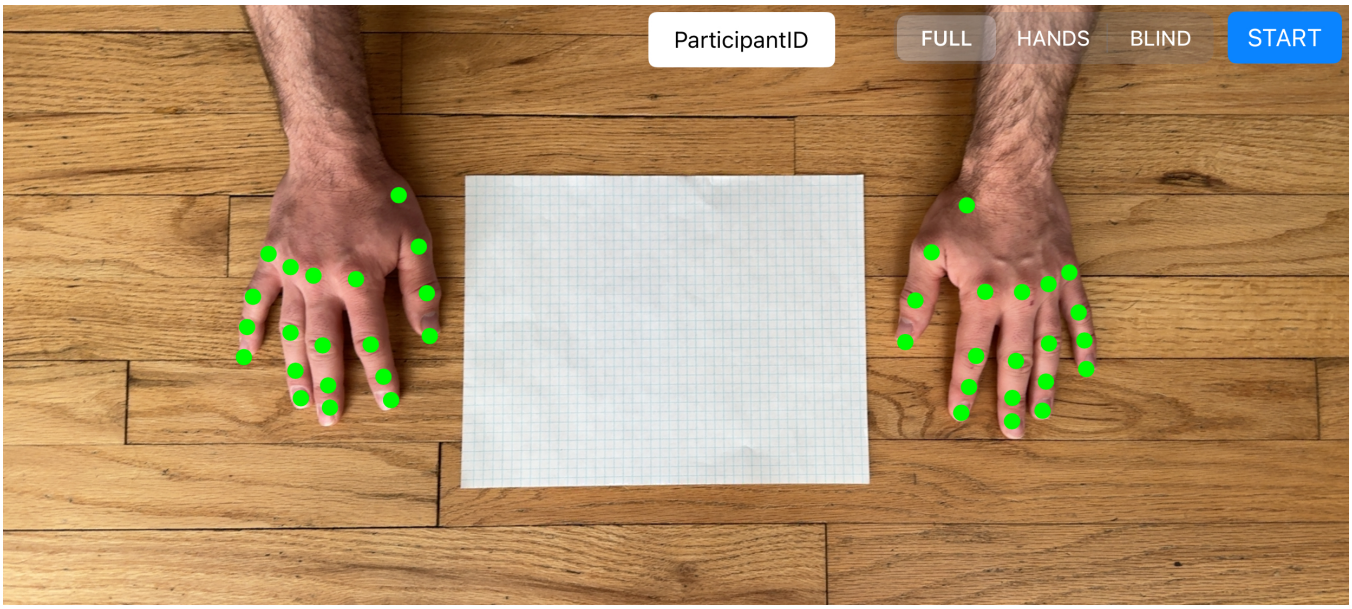

Figure 1: Hand Pose estimation application

ABSTRACT

In this experimental study, we investigate hand pose estimation and gesture analytics with paper folding exercises. Our methodology involves the utilization of an iOS application to collect hand point data during trials where subjects are instructed to fold paper in a standardized pattern. The user is instructed to perform the folding in 3 separate manners; eyes open, eyes not focused on paper, eyes closed. The study performs an evaluation involving diverse users, aiming to provide a robust method for tracking hand pose data. By evaluating individual fold patterns, a consistency or trend that can be identified across various trials and users may serve as a method to uncover unique hand patterns. Utilizing Apple's VisionOS Framework, the built-in machine learning models for human hand detection can track up to 40 distinct hand points in a frame in real time. While the key to our analysis is precise finger position data, data recording performance is important in determining trends in folds. Our study shows newer mobile devices have significant performance increases with data capture. We aim to provide insights for leveraging VisionOS based hand pose estimation, and the exploration of identifying unique hand gesture patterns for paper folding.

CS 567 – Intro to 3D UI, Fall '23, CSU, Fort Collins, CO

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/XXXXXXX.XXXXXXX

CCS CONCEPTS

• **Computing methodologies** → **Tracking**; • **Human-centered computing** → **Gestural input**; **Mixed / augmented reality**; *User interface design.*

KEYWORDS

Augmented Reality, Computer Vision, Hand Tracking, Gesture Detection, Hand Gestures, Hand Pose Estimation, VisionOS, Paper Folding

ACM Reference Format:
Ian Brown, Tani Cath, and Tom Cavey. 2023. FoldAR: Gesture Analytics Using Apple Vision Framework Final Project Report. In *CSU, Fall '23: CS 567 – Introduction to 3D User Interfaces, August 21–December 10, 2023, Fort Collins, CO.* ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Gesture recognition and hand pose estimation are important components of human-computer interaction, playing a crucial role in the user experience in various applications. Interactive computing using natural hand gestures involve sophisticated tracking devices using sensors, cameras, and software infrastructures built for hand tracking [16].

Gesture tracking is achieved with high accuracy with the use of machine learning and high resolution hardware [29]. The objective is to leverage the iOS platform and Apple's Vision Framework, to achieve real-time tracking and analysis of hand movements during the process of folding paper.

The motivation behind this study comes from the growing world of Human Computer Interaction systems and the need for continuously improving the accuracy and efficiency of hand pose estimation systems. As embedded sensors and cameras become capable of capturing more intricate details, requirements for an application that can accommodate complex and dynamic hand movements is increasing [5]. Current approaches often do not have the precision to capture the details of intricate tasks like paper folding, where smaller movements and subtle gestures are crucial for understanding the user's intended hand movements. Our research aims to address these challenges and contribute to the advancement of gesture recognition systems while focusing on applications involving hands-on activities, such as paper folding.

Through extensive experiments we evaluate how different paper folding exercises are performed. The choice to use paper folding as the study is related to the inherent complexity and dexterity required to perform a combination of precise hand movements. By analyzing the hand pose data collected during the exercises, we seek to develop a method for identifying distinct folding stages. The results of the study can be used for various other applications ranging from interactive educational tools tot he development of more intuitive human-computer interfaces. Our methodology includes the utilization of Apple's Vision Framework which is equipped with many human body based machine learning models. Specifically we employ the use of The Hand Gesture model [6], which is capable of identifying up to 40 individual points for per hand.

2 RELATED WORKS

Our experiment in paper folding was inspired by a vast amount of prior work and this section aims to introduce methodologies, research, and documentation we find relatable to our experiment. Hand tracking can be performed with the use of a sophisticated hand tracking glove [17]. Dedicated sensors designed to capture hand gestures yielded high precision and accuracy, however is limited in flexibility compared to using a mobile device camera system. Apple's technology has made it possible for a mobile device to capture equivalent data. In [28] the challenges of tracking the intricate structure of the human hand are shown to be difficult due to the complex structure. Hand motion is highly articulate and can be difficult for computer vision based models, although the newest mobile phone cameras and processors are generations ahead of compute power and resolution. On the other hand, hand motion is also limited the range of motion in human joints and reduce the amount that each finger can articulate. Hand pose estimation can also be performed by segmenting individual fingers. In [19] a method is presented for determining gestures based on the number of fingers present in the frame. This reduces the functionality of gesture recognition but can be beneficial if only limited finger tip or finger count information is required.

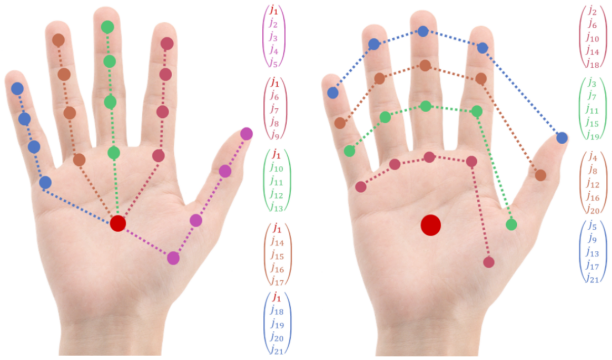

Figure 2: The nine tuples used to construct the Shape of Connected Joints (SoCJ) descriptors. Reproduced from [22]

In an effort to gain understanding in how Apple may have designed their human hand pose estimation model in [1] an approach is presented which demonstrates how a computer vision model using a convolutional neural network will detect a hand. The research also presents a few different methodologies to improve the performance and robustness of hand detection models.

More recent approaches to using hand detection using a self-attention model [21] achieve consistent and accurate detection while also managing to create a lightweight model optimized for low powered embedded or mobile devices. The dataset used to train the model included images in-the-wild, occluded hand images, and images of hands holding objects. Other examples of 2D models trained on hand pose datasets include [30], which boasts the use of an 18,000 stereo image dataset to train the model.

Our study needed works that specifically mapped hands along with their gestures so the recognition system from SHREC'17 was just one of many helpful sources [22]. While they focus on the development of a new 3D dynamic hand gesture dataset [22] and our focuses on using hand tracking to track the movement of people's hands, their mapping and modeling was a nice legacy work to compare to. Their models of hands, tracking skeletons, consideration of depth data, and especially the similarities in their skeleton to ours was reassuring, see Figure 2.

In *Towards Preventing Gaps in Health Care Systems through Smartphone Use: Analysis of ARKit for Accurate Measurement of Facial Distances in Different Angles* [15], they discuss the importance of accurate facial distance measurements in healthcare. The authors

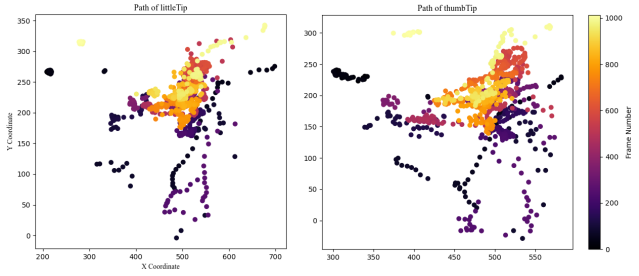

Figure 3: Temporal data for little finger and thumb tip trajectories

propose a new method for measuring facial distances using a smartphone and ARKit. They conducted an experiment to evaluate the accuracy of their method. The results of the experiment showed that their method is accurate and can be used to measure facial distances in different angles. The process by which they went about improving their model, using Apple devices and frameworks, proved helpful and similar enough despite changes in the tools apple provided since then.

A study from Chonnam National University [24] was relevant to our study because it discusses the potential of hand gesture recognition for human-computer interaction (HCI). The authors of the article propose a novel method for fingertip detection and hand gesture recognition in real-time using an RGB-D camera and a 3D convolution neural network (3DCNN) [24]. Their system achieved high accuracy and robustness in recognizing a variety of gestures, suggesting that hand gesture recognition is a promising approach for HCI.

Smartphones are increasingly being equipped with advanced censors and frameworks that access them differently, along with different implementations of models and data processing [26]. While they focused on using ultrasonic sensing to accurately capture hang gestures [26], they're applications architecture, information processing, and optimizing around a proprietary tools was quiet similar to various challenges we faced using Apple's Vision framework and SDK.

A study on how hand tracking enhances the immersive experience by placing the user in a first-person perspective, allowing them to fully perceive the position of their hands [25]. This study explores the impact of hand tracking on memory assessment in IVR systems. To this end, an application based on daily living activities was developed, where users must recall the location of various items. This was useful given we used a common activity and items (pieces of paper) and was important throughout to understand how our changes in constraints in folds were observed both in our tracking and the data.

Computer Vision is a vast field of study and has many endeavours in improving it's recognition abilities. However, a study into key points and hand bounding boxes proved useful [4]. FoldAR had various tools for developing a dynamic tracking system however before fully understanding and processing a dynamic movement a static recognition as necessary. Their insight into the effects of tracking accuracy at various points of the hand over others [4] influenced FoldAR's decision making when considering wrist tracking, finger joints, and etc.

In Banerjee's study [2] there were many complementary aspects and especially with the perspective of tracking of hands. Their idea of using hand "telerehabilitation" provided insights into the potential of using computer vision to improve remote monitoring of hand trauma via a common smartphone [2] and doing an easily accessible activity gave us an necessary perspective of how we could both visualize and constantly capture a consistent, seemingly mundane activity while closely monitoring accuracy of the tracking.

The Origami Guru project was one of few similar studies that involved gestures, paper folding, and using computer vision tracking to aid users in the task [27]. While FoldAR eventually moved away from Origami paper folding, their work, challenges, and attempted workarounds help FoldAR better narrow down tracking points, better define metrics of accuracy,

Deploying a hand detection model onto a mobile device means that data can be captured about hand poses in scenes that the model may not have been trained to do. This requires a robust model that has been trained to track hand movements in various environments and backgrounds [23]. The VisionOS hand pose estimation model is able to work in a variety of lighting and background scenarios. A state of the art detection model introduced in [20] shows how a real time tracking can be greatly improved by combining two algorithms: Particle Swarm Optimization [14] and Iterative Closest Point [18].

3 CONTRIBUTIONS

FoldAR presents a unique and tested implementation for tracking finger tip movement using Apple's Vision framework. FoldAR was developed using Xcode 15.1 (beta) and source viewed as public GitHub repository under *FA23-FoldAR-Step-by-step-Instructions-for-Folding-Paper-Models-in-AR*

4 EXPERIMENTAL METHODOLOGY

The hand point data is used to gain an understanding of the user's had positions while performing folds. The research question we hope to answer with this experiment is two parts, the first is whether or not we can identify the 3 individual folds based on the hand pose data. The second part is to determine if there are any trends with the data about test subjects, modes, or folding patterns. Participants were selected by the researchers, and all subjects must be able to complete folds both visually and by feel.

The first method we use to describe the data is to calculate the Euclidean distance between fingertip points. The distance between finger tips may be used to determine unique characteristics of folds, which stage of the fold a user has performed, or how many folds occurred in a trial. In combination with fingertip points and timestamp data, it is also possible to calculate velocity data, such as folding speed, time to finish, and finger acceleration. In order to evaluate the hand tracking and gesture recognition capabilities of Apple's Vision Framework, the application was developed and a series of experiments were conducted to generate data that would then be processed accordingly, see Section 4.2.

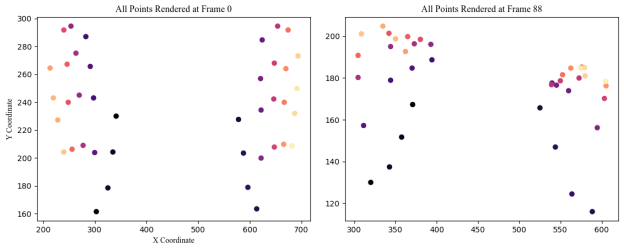

Figure 4: Points recorded during frames 0 and 88 of an experimental session

It may also be possible to determine which stage of the fold a user is in by the clustering of points. By tracking the movement key finger tip points such as the thumb, index, and little fingers over the entire trial the folds. The position of finger tips over the course of a folding trial reveals several distinct positions or areas within the frame. The clustering of this data shows that individual folds can be determined.

4.1 Data Collection Details

Researchers use the application to fill out userID, select the experimental mode, and begin the capture by pressing the start button. As discussed in Section 5, the application is built on the foundation of Apple's Vision Framework subsystem and utilizes a pre-trained hand pose detection model to identify individual joints of a user's hand(s) [6].

The Vision Framework hand detection model is capable of identifying four points per finger, three joints – metacarpophalangeal/carpometacarpal, proximal interphalangeal, and distal interphalangeal; MCP/CMC, PIP, and DIP, respectively and the fingertip, along with the wrist. These points can be detected for up to two hands simultaneously, the coordinate of each point is projected onto a two-dimensional plane relative to the software's input device (camera), for a total of up to 42 individual points or 84 values (X- and Y-coordinates), see Figure 4. These data points are processed using Python3 in a Jupyter Notebook which leverages the Pandas dataframe objects to perform operations on the data and MatPlotLib to plot and visualize the information.

Because the hand pose detection functionality comes pre-built into Apple's Vision Framework library (available in Xcode version 15.1 Beta and later), no additional machine learning model training was required. It's important that the research label the data correct, as there's no retroactive method to delete previously recorded data through the application.

4.2 Experimental Design

In order to collect useful data, a simple experiment was performed with fourteen individual participants as follows:

(1) The participant was asked to sit or stand in front of a flat work surface (table or counter) with a single sheet of 8.5x11 inch paper in front of them and their hands laying palms-down on either side of the paper.
(2) The iOS device running our application (one iPhone 8 and two iPhone 15 Pro Max's) was positioned facing the participant, approximately 36 inches away from them and 24 inches above the work surface and angled downwards so as to center the participants' hands and arms in the camera frame.
(3) The participant was asked to fold the sheet of paper in half three times using both hands using whatever technique they chose.
(4) As the participant performed the folding action, hand point data (as described in Section 4.1) was collected, see Section 4.3 for details.
(5) Once the third fold was completed, the participant placed the paper on the work surface in front of them and laid their hands, palms down, on the work surface on either side of the folded paper.

Steps 1-5 where repeated a total of three times with the following additional constraints placed on the participants:

- 1st repetition: No additional constraints.
- 2nd repetition: The user was instructed to look away so as not to look at their hands.
- 3rd repetition: The participant was instructed to close their eyes so as not to be able to see at all.

4.3 Dataset Collection and Details

As discussed in Section 4.1, Apple's Vision Framework hand pose detection library is capable of detecting up to 42 total points (21 points per hand). For our research we opted to ignore the wrist-point for a total of 40 points from two hands.

This data was recorded during the experimental process into a comma-separated values (CSV) file with each row containing a total of 85 columns, 80 for the finger point coordinates along with the participant's anonymous three-digit id (pid), the "mode" of the current experiment (full vision, hands not in vision, no vision, represented as a 0, 1, or 2), the frame count starting from the beginning of the session, and timestamp data including both date and time with millisecond resolution.

This data was recorded locally into the application's "sandboxed" documents directory on the iOS device, with an individual CSV file generated for each participant and mode/session, with a new row of data written to the file every frame. Once experiments were concluded, the CSV session data files were offloaded from the iPhones and loaded into a Python notebook for further data processing and analysis, see Section 6 for details.

5 APPLICATION DESIGN AND DEPLOYMENT

The design and development of our iOS application plays a critical role to our research, serving as the only tool for capturing and generating data logs during our experimental trials. This app is specifically developed for the paper folding experiment described in this study. The application reliably performs inference in real time with a high frames per second data recording rate, as described in the application performance section. This is mostly due to the specifically designed hardware and firmware contained within the Apple Neural Engine. In 2023 the Apple A17 Pro is capable of 35 trillion operations per second [8]. The advancement of Machine

Learning applications benefit from these technical advancements, and Apple provides a well documented API for developers. This section covers a detailed insight into the details of our application and focuses on the utilization of Apple Vision API's and the integration of of gesture analytics during paper folding exercises.

5.1 Swift and Vision

The core component of the application uses the UIViewController class [11] which controls the rendering of content on the screen. We leverage this class to show the live video feed sourced from the back-facing camera. The video frames from the camera feed are accessed with an AVCaptureSession [10] API, resulting in real-time capture of video frames that can be processed in the core logic of the app. The view controller, along with the AV capture session enable us to process and render an overlay to the live camera view. Additional user interface elements are added for researchers to record subject ID, document the experimentation mode, and use a start/stop button to control when hand points are recorded. The UI items are rendered independently of the AV capture session, and are not subject to the rendering of video frames.

5.2 HandPoseEstimation Model

When a video frame is captured we can make a hand detection request using the the Vision framework's built in model (VNDetectHumanHandPoseRequest) [12]. The frame is processed by the Vision request API and a the returned data object contains a prediction score, the x-and-y coordinates of detected finger joints, chirality among other useful information for hand pose estimation. For a single frame, the hand detection request can detect up to 2 hands. The documentation states only the largest hands in the scene will have the points be predicted. The user id, mode, and hand points are recorded and saved within the application's sandbox. No image data is preserved, and the resulting files of the experiment are small and allows research to be conducted in a variety of conditions. Data is offloaded from the device using the Xcode application on the Mac OS development system.

The framework provides a comprehensive list of points at the joint of each finger. The hand request returns a set of points that represent coordinates for the tip and joints' on all five fingers [13]. As discussed in Section 4.1, the points available on the thumb are: tip, interphalangeal, metacarpophalangeal, carpometacarpal; for the index, middle, ring and little fingers the points are: tip, distal interphalangeal, proximal interphalangeal, metacarpophalangeal. The result of this request can be used to access 2D coordinates of up to 40 specific finger joints within the frame. These points are displayed in real time to the iPhone's display.

6 RESULTS AND DISCUSSION

In the findings of our results we make positive observations that support our initial hypothesis, but also create new opportunities to expand on finding a deeper meaning in the human part of the data. In this section we provide quantitative data on how point distance calculations and other point tracking algorithms can be used to identify how many folds occur, when they occur, and the possibility of recognizing individuals by the way they fold. First to support our initial hypothesis, we find that there are a few indicators that can be used to detect distinct folding actions. Secondly, we inadvertently discovered that there may be trends in the recorded data points that can be used to distinguish subjects from one another. We find that each subject's finger trajectories follow distinct patterns that could be used as biometricly identifiable data to that subject.

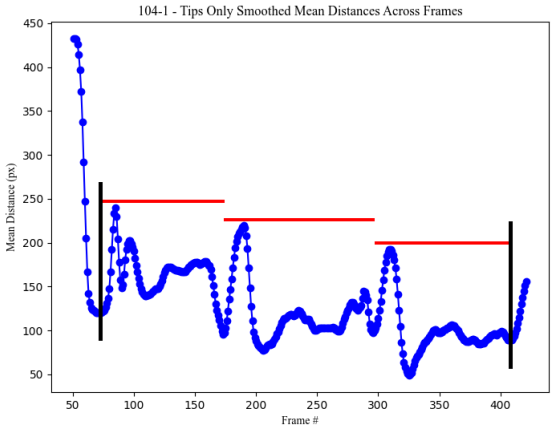

Figure 5: Mean distances of fingertips over all frames in a session. Black vertical bars represent assumed start and end points of folding sequence, red horizontal bars represent assumed distinct fold stages

6.1 Mean Euclidean Distances

The Euclidean distance formula is used to calculate the distance between two points on a 2-dimensional plane. This calculation is critical to our results in finding trends in folding behavior.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{1}$$

There are 3 different Euclidean distance calculations we use to help interpret the data. The first distance calculation performed is the mean of pairwise distance between finger points for a given frame. This provides a general measurement of how close the left and right hands are. The second distance calculation performed is the mean distance between finger tips on each individual hand. This effectively provides a general measurement for how close the fingertips are to each other, on each respective hand. The third distance calculation performed is the mean distance between all points on each individual hand. This measurement is generally an idea of how open or closed a hand might be. Other point calculations were experimented with but these three points are enough to show a positive correlation with fold stages. In most plots, there is a general trend of incrementally decreasing the mean distance between hands and fingertip points.

Also reference Table 1

6.2 Fold Stage Detection

The individual fold stages within a session can be shown by calculating the Euclidean distances between selective finger or joint points. First, in Figure 5, there are distinct change points roughly

Table 1: Distance between all finger tips on a single hand, averaged over five quantiles of datatset

		Mean Distance				
User Id	Trial #	1	2	3	4	5
102	0	83.73	50.42	37.31	32.96	43.98
102	1	133.81	97.82	59.74	37.73	60.69
102	2	65.27	45.99	42.00	37.74	38.07
103	0	107.68	54.29	55.44	52.82	71.01
103	1	83.87	62.72	52.35	54.66	81.61
103	2	86.11	50.09	60.19	51.85	65.25
104	0	98.34	61.93	54.62	55.48	69.07
104	1	104.65	62.83	50.40	52.47	70.21
104	2	83.45	56.48	60.96	58.04	63.32
			...			
302	0	50.10	36.65	41.64	39.21	46.50
302	1	47.70	28.79	26.92	26.72	28.79
302	2	60.74	31.35	28.24	22.51	38.78

at frame 80, 180, and 310 where relatively quick changes in finger tip locations are observable. An indication that the points temporal data may be the useful in identifying folds or other data. We find this trend to be accurate across all users and modes. Some examples are more difficult to interpret due to possibly noisy data or lower data capture speed. In the analysis of the mean distances data table [cite] we found that across all users as frame counts increase, the the mean distances become smaller. This gives some indication of the scale of paper folding becoming smaller with each half fold.

6.3 User Identification

Looking at the traced paths of a single fingertip over the course of a folding session, as well as over the three experimental modes, and then comparing these plots between two individual participants, one can clearly see drastic differences such as in Figures 10 and 11 in Appendix B.

As these two figures clearly show, individual participants, when provided with a sheet of paper and the open-ended instruction to "fold the paper in half three times", each perform that task in vastly differing ways. This observation leads to the intuition that using only hand-point data during a given task, individuals could not only be distinguished from each other, but potentially could also be directly identified given a large enough dataset of how they user their hands in a variety of tasks, as discussed in Section 6.5.2.

The temporal data relationship between frames can help to further identify unique behaviors, from the finger trajectory path data. The trajectory is plotted using color coding that changes as the individual frame count increments. Certain participants perform patterns throughout each trial, as we can see in 3 the quick succession of vertical points in black show how a rapid and sweeping movement pattern occurs some time in the beginning of the fold trial. The data can be further analyzed to understand the time captures and the distances the finger tip points moved within a given amount time to determine spikes in finger acceleration.

When considering specific fingers in most cases we find that the index, middle, and ring finger tips generally follow the same dynamic movement paths see appendix, and the thumb tip and pinky tip have trajectories that can be particularly specific. For example, the indexTip path for user 102 during the mode 0 and mode 1 experiments have a significant amount of overlap versus mode 3. However, when comparing user 102 across all modes to user 302, they appear to be completely different patterns. 102 can be described as having an angular, stretched, or narrow arrangement where 302 has a sparse or rounded trajectory.

Table 2: Participant completions times by experimental mode

Compl. Time (s)	Full Vision	Hands Obscured	No Vision
Mean	22.395	22.788	29.190
Minimum	3.559	6.675	8.804
Maximum	72.903	57.437	63.200
Median	19.608	18.015	24.488
First Quartile	12.849	12.398	17.871
Third Quartile	24.094	31.388	41.288
IQR	11.244	18.990	23.417
Std. Deviation	16.276	13.915	15.738

The uniqueness that is observed in the individual finger paths also appears to remain consistent across trial modes. For example user 205 has distinguishable vertical lines with emphasis on the left hand side of the plots. This can be seen across all the trial modes and this consistency reinforces the idea that the patterns we observe are likely an indication of habitual or muscle memory characteristics when it comes to folding paper. However, this does not hold true for all subjects that were tested. For example User 107 on mode 2 (the eyes closed method) was left justified heavily, with only a few single points on the right hand side. There are much fewer points that overlap when compared to the prior mode 0 and 1 for that user.

Finally, we calculate an overlay of points in an attempt to understand how similar a user's folding patterns are to themselves. If the distance between any two points is below a threshold, we consider it an overlaying point. For our example we used the euclidean distance between two points with a threshold of 1.0. We find that generally most users are more similar than to others, with some exceptions. This can be expanded on to attempt to use the data to try identifying and masking noisy data or removing actions that do not contribute to uniqueness.

6.4 User Performance

One key metric that was analyzed from the experimental data was the time it took each participant to complete a fold sequence (see Section 4.2). As can be seen in Table 2 and Figure 6, while the average completion times were very similar, especially for the first and second experimental modes (full vision and hands obscured) with the third (no vision) averaging only slightly slower, with experimental mode 0 (full vision) having the tightest interquartile range.

While the statistical data shows a general pattern that progressively decreasing a user's vision capabilities directly correlates to decreased performance (longer completion times), the limited sample size of only fourteen participants most likely skews the data as one participant (108) showed almost no difference in completion time between partial vision and no vision, while three participants (106, 205, and, most noticeably, 300), roughly 21%, showed greater

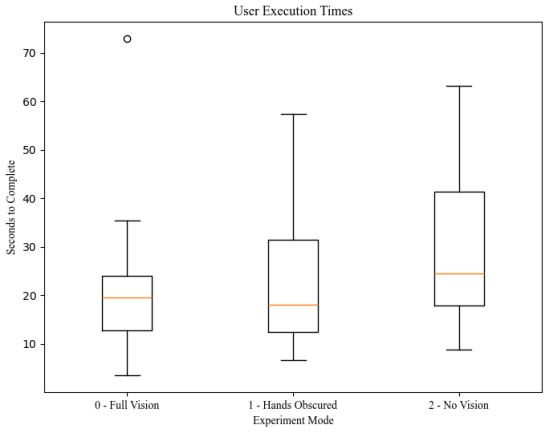

Figure 6: Participant completion time comparison by experimental mode

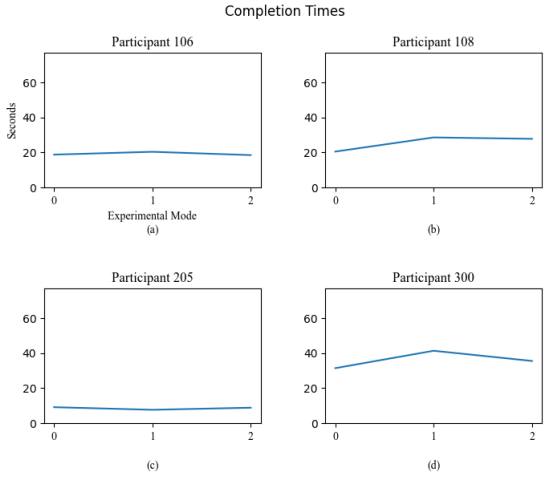

Figure 7: Participant completion times slower with partial vision than with no vision

performance (faster completion times) no vision at all compared to limited vision, often attempting to close their eyes during this experimental mode in order to improve their efficiency, as can be see in Figure 7. In addition, seven participants (50%) performed *slower* with full freedom of vision than with partial vision, and in some cases even slower than with no vision at all, such as participant 102 as seen in Figure 8.

6.5 Future Work

6.5.1 Time and Budget Constraints. Due to the time and budgetary limitations of this research, our team would greatly benefit from having more time to collect data from a larger pool of participants. In addition, performing the experiments in a more controlled environment would allow for greater consistency both in how the data is collected (more optimal lighting conditions and fixed angles and

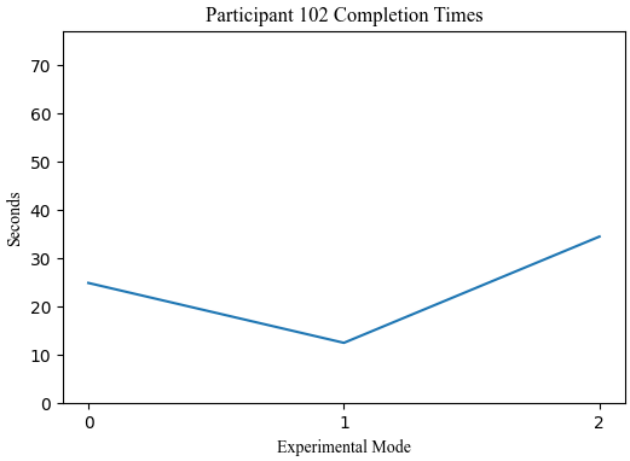

Figure 8: Participant 102's completion times

distances between the participants and the recording equipment), as well as being able to use uniform hardware for data collection to alleviate discrepancies between iPhone generations, as discussed in Appendix A.

6.5.2 Additional Research. As discussed in Section 6.3, one potential path of future research into identifying individuals based solely on hand-point data could be to prompt participants to perform series of different tasks that involve manipulating objects with their hands. In this context the folding can be used as a biometric identification technique. If the folding experiments were designed in a way to remove noise and lots of labeled data was captured, the collected data could be used to train a machine learning algorithm such as Random Forest to perform pattern recognition and identify individuals based on the way they use their hands. This is similar to keystroke recognition, which is used as a method of user identification and authentication [3].

The current data can provide insights into the finger tip speed or acceleration. Using the timestamp data, it is possible to calculate the relative speed of the finger tip points and determine spikes in finger acceleration. This could add a beneficial perspective to the data, and leverages the temporal dynamics of the experiment.

6.5.3 Lessons Learned. Future work: We should've asked them what primary hand do they use?

7 CONCLUSION

What were we expecting versus what did we get?

What were we expecting vs what did we get

How does the data indicate a trend

ACKNOWLEDGMENTS

We would like to thank Dr. Francisco Ortega, Richi Rodriguez, and all the individuals who participated in the experimental portions of our study.

REFERENCES

[1] Mahmoud Afifi. 2018. 11K Hands: Gender recognition and biometric identification using a large dataset of hand images. arXiv:1711.04322 [cs.CV]

[2] Tania Banerjee. 2023. *Computer Vision-Based Hand Tracking and 3D Reconstruction as a Human-Computer Input Modality with Clinical Application.* Master's thesis. University of Western Ontario. https://ir.lib.uwo.ca/etd/9173

[3] Nick Bartlow. 2009. *Keystroke Recognition.* Springer US, Boston, MA, 877–882. https://doi.org/10.1007/978-0-387-73003-5_205

[4] Tuan Linh Dang, Sy Dat Tran, Thuy Hang Nguyen, Suntae Kim, and Nicolas Monet. 2022. An improved hand gesture recognition system using keypoints and hand bounding boxes. *Array* 16 (2022), 100251. https://doi.org/10.1016/j.array.2022.100251

[5] Simon Fortin-Deschênes, Vincent Chapdelaine-Couture, Yan Côté, and Anthony Ghannoum. 2023. Patent US10838206B2 – Head-mounted display for virtual and mixed reality with inside-out positional, user body and environment tracking. https://patents.google.com/patent/US10838206B2/en

[6] Apple Inc. 2020. *Detect Body and Hand Pose with Vision.* Technical Report. Apple World Wide Developer Conference. https://developer.apple.com/videos/play/wwdc2020/10653/

[7] Apple Inc. 2022. *iPhone models compatible with iOS 16.* Technical Report. Apple Support Website. https://support.apple.com/guide/iphone/supported-models-iphe3fa5df43/16.0/ios/16.0

[8] Apple Inc. 2023. *iPhone 15 Pro and iPhone 15 Pro Max.* Technical Report. Apple Product Website. https://www.apple.com/iphone-15-pro/

[9] Apple Inc. 2023. *iPhone models compatible with iOS 17.* Technical Report. Apple Support Website. https://support.apple.com/guide/iphone/models-compatible-with-ios-17-iphe3fa5df43/ios

[10] Apple Inc. n.d.. *AVCaptureSession.* Technical Report. Apple Developer Documentation. https://developer.apple.com/documentation/avfoundation/avcapturesession

[11] Apple Inc. n.d.. *UIViewController.* Technical Report. Apple Developer Documentation. https://developer.apple.com/documentation/uikit/uiviewcontroller

[12] Apple Inc. n.d.. *VNDetectHumanHandPoseRequest.* Technical Report. Apple Developer Documentation. https://developer.apple.com/documentation/vision/vndetecthumanhandposerequest

[13] Apple Inc. n.d.. *VNHumanHandPoseObservation.JointName.* Technical Report. Apple Developer Documentation. https://developer.apple.com/documentation/vision/vnhumanhandposeobservation/jointname

[14] James Kennedy. 2010. *Particle Swarm Optimization.* Springer US, Boston, MA, 760–766. https://doi.org/10.1007/978-0-387-30164-8_630

[15] Klinker Kapsecker Leube Schneckenburger Jonas Nissen, Hübner. 2023. Towards Preventing Gaps in Health Care Systems through Smartphone Use: Analysis of ARKit for Accurate Measurement of Facial Distances in Different Angles. *MDPI* (2023). https://doi.org/10.3390/s23094486

[16] Munir Oudah, Ali Al-Naji, and Javaan Chahl. 2020. Hand gesture recognition based on Computer Vision: A review of techniques. *Journal of Imaging* 6, 8 (2020), 73. https://doi.org/10.3390/jimaging6080073

[17] Timothy F. O'Connor, Matthew E. Fach, Rachel Miller, Samuel E. Root, Patrick P. Mercier, and Darren J. Lipomi. 2017. The Language of Glove: Wireless gesture decoder with low-power and stretchable hybrid electronics. *PLOS ONE* (2017). https://doi.org/10.1371/journal.pone.0179766

[18] Stefano Pellegrini, Konrad Schindler, and Daniele Nardi. 2008. A Generalisation of the ICP Algorithm for Articulated Bodies. In *British Machine Vision Conference.* https://api.semanticscholar.org/CorpusID:12104382

[19] M. Perimal, S.N. Basah, M.J.A. Safar, and H. Yazid. 2018. Hand-Gesture Recognition-Algorithm based on Finger Counting. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* 10, 1-13 (May 2018), 19–24. https://jtec.utem.edu.my/jtec/article/view/4115

[20] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. 2014. Realtime and Robust Hand Tracking from Depth. In *2014 IEEE Conference on Computer Vision and Pattern Recognition.* 1106–1113. https://doi.org/10.1109/CVPR.2014.145

[21] Nicholas Santavas, Ioannis Kansizoglou, Loukas Bampis, Evangelos Karakasis, and Antonios Gasteratos. 2020. Attention! A Lightweight 2D Hand Pose Estimation Approach. arXiv:2001.08047 [cs.CV]

[22] Quentin De Smedt, Hazem Wannous, Jean-Philippe Vandeborre, J. Guerry, B. Le Saux, and D. Filliat. 2017. 3D Hand Gesture Recognition Using a Depth and Skeletal Dataset. In *Eurographics Workshop on 3D Object Retrieval*, Ioannis Pratikakis, Florent Dupont, and Maks Ovsjanikov (Eds.). The Eurographics Association. https://doi.org/10.2312/3dor.20171049

[23] Ekaterini Stergiopoulou, Kyriakos Sgouropoulos, Nikos Nikolaou, Nikos Papamarkos, and Nikos Mitianoudis. 2014. Real time hand detection in a complex background. *Engineering Applications of Artificial Intelligence* 35 (2014), 54–70. https://doi.org/10.1016/j.engappai.2014.06.006

[24] Dinh-Son Tran, Ngoc-Huynh Ho, Hyung-Jeong Yang, Eu-Tteum Baek, Soox-Hyung Kim, and Gueesang Lee. 2020. Real-Time Hand Gesture Spotting and Recognition Using RGB-D Camera and 3D Convolutional Neural Network. *Applied Sciences* 10, 2 (Jan. 2020), 722. https://doi.org/10.3390/app10020722

Table 3: Frametime data by iOS device, averaged over all datasets

Frametime (ms)	iPhone 8	iPhone 15 Pro Max
Mean	245.652	34.039
Minimum	237.992	33.285
Maximum	255.900	35.843
Median	242.845	33.818
First Quartile	239.872	33.426
Third Quartile	250.605	34.270
IQR	10.733	0.844
Standard Deviation	6.704	0.715
Average Framerate (FPS)	4.071	29.378

[25] José Varela-Aldás, Jorge Buele, Irene López, and Guillermo Palacios-Navarro. 2023. Influence of Hand Tracking in Immersive Virtual Reality for Memory Assessment. *International Journal of Environmental Research and Public Health* 20, 5 (3 2023), 4609. https://doi.org/10.3390/ijerph20054609

[26] Zhengjie Wang, Yushan Hou, Kangkang Jiang, Wenwen Dou, Chengming Zhang, Zehua Huang, and Yinjing Guo. 2019. Hand Gesture Recognition Based on Active Ultrasonic Sensing of Smartphone: A Survey. *IEEE Access* 7 (2019), 111897–111922. https://doi.org/10.1109/ACCESS.2019.2933987

[27] Nuwee Wiwatwattana, Chayangkul Laphom, Sarocha Aggaitchaya, and Sudarat Chattanon. 2016. Origami Guru: An Augmented Reality Application to Assist Paper Folding. In *Information Technology: New Generations*, Shahram Latifi (Ed.). Springer International Publishing, Cham, 1101–1111. https://doi.org/10.1007/978-3-319-32467-8_95

[28] Ying Wu and Thomas S. Huan. 2001. Hand Modeling, Analysis, and Recognition for Vision-Based Human Computer Interaction. *IEEE Signal Processing Magazine* (2001).

[29] Yi Xiao, Tong Liu, Yu Han, Yue Liu, and Yongtian Wang. 2023. Realtime recognition of dynamic hand gestures in practical applications. *ACM Transactions on Multimedia Computing, Communications, and Applications* 20, 2 (2023), 1–17. https://doi.org/10.1145/3561822

[30] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. 2016. 3D Hand Pose Tracking and Estimation Using Stereo Matching. arXiv:1610.07214 [cs.CV]

[31] Thomas Zinner, Oliver Hohlfeld, Osama Abboud, and Tobias Hossfeld. 2010. Impact of frame rate and resolution on objective QoE metrics. In *2010 Second International Workshop on Quality of Multimedia Experience (QoMEX).* 29–34. https://doi.org/10.1109/QOMEX.2010.5518277

A APPLICATION PERFORMANCE

As was mentioned in Sections 4.2 and 5, experiments were conducted using two different Apple iPhone models, the iPhone 8 (2017), and the iPhone 15 Pro Max (2023). One significant performance discrepancy between the models used that was immediately noticeable in the data was the amount of time required to process a single frame of information. As can be seen in Table 3, while the iPhone 15 is able to maintain a very consistent average frametime of 34 milliseconds (just under 30 frames per second) with a standard deviation of only 0.715, the iPhone 8 struggles to maintain an average frametime of 245.6 ms (a slideshow-level four frames per second), with a significantly greater standard deviation of 6.7.

Figure 9 visualizes this discrepancy, and shows just how much processing power is required to run Apple's Vision Framework and how, despite the iPhone 8 being *technically* capable of running an augmented reality application built on the Framework, the user experience is drastically degraded due to the extremely low framerates (high frametimes) which have been found to have a significant negative impact on user quality of experience (QoE), especially at values lower than 15 FPS [31].

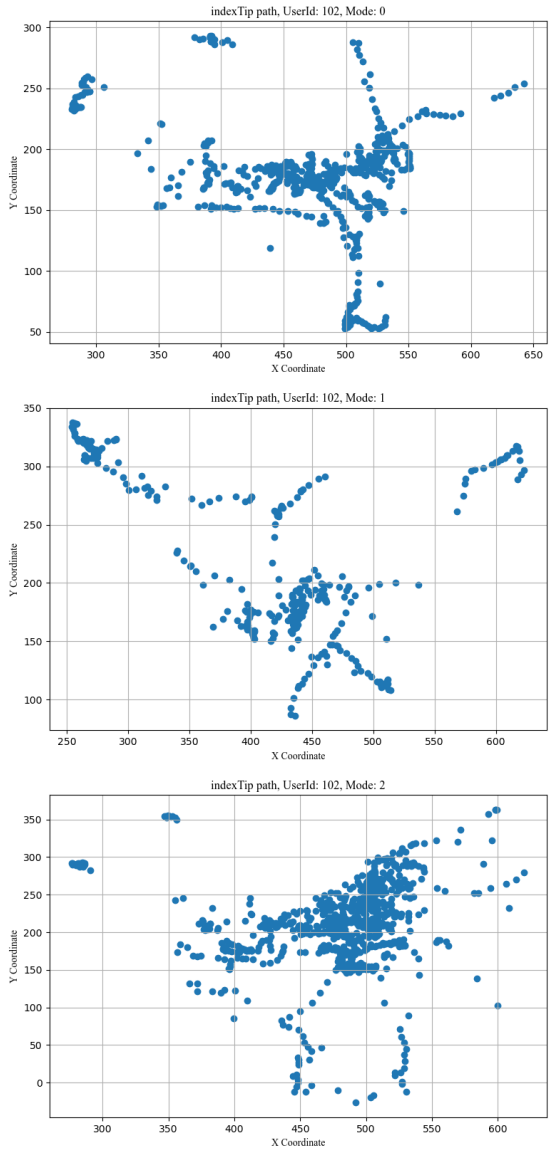

Figure 10: Path of Participant 102's index fingertips during all three experimental modes

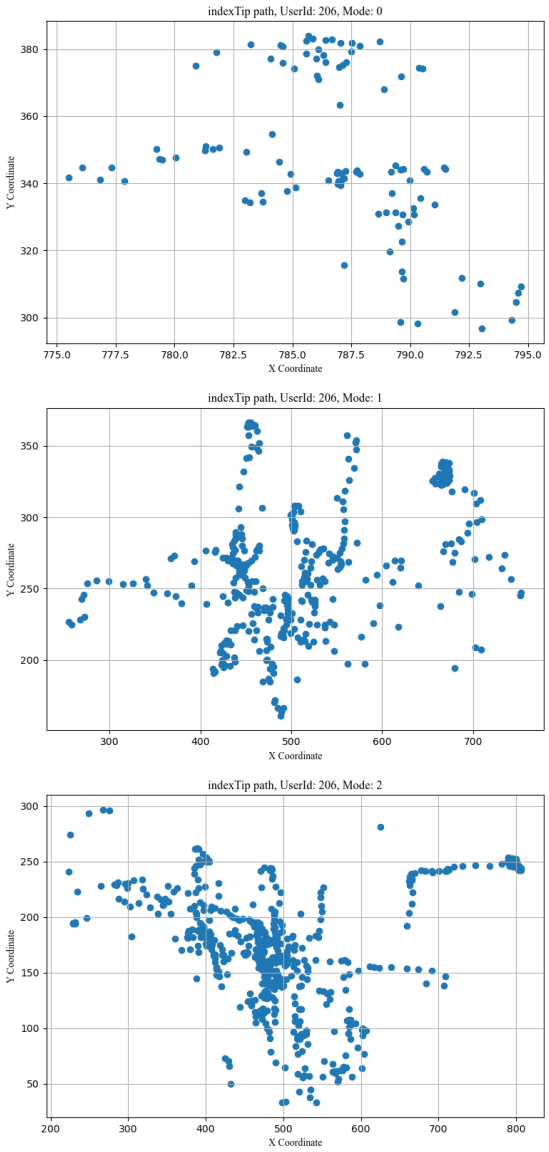

Figure 11: Path of Participant 206's index fingertips during all three experimental modes

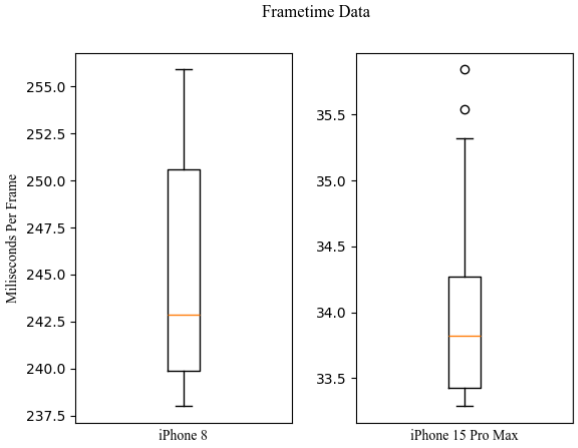

Figure 9: Frametime data comparison between iPhone 8 and iPhone 15 Pro Max

While the performance difference between the iPhone models is unsurprising considering that Apple officially dropped the iPhone 8's support for the most recent Apple mobile operating system (iOS 17) [7, 9], the data collected on the older hardware is still usable in our analysis, though roughly 7.2 times more sparse. Similar to the limitations imposed due to a small sample size of participants discussed in Section 6.4, these performance differences due to hardware variations are due to the scope of the study as a full-scale research project would have also allocated funds to provide all researchers with identical hardware.

B SUPPLEMENTARY TABLES AND FIGURES

Received 9 December 2023