THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Efficient and Accurate Motion Tracking and its Applications within the Internet of Things Design Space

KEVIN GARDNER
SPRING 2021

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Computer Engineering
with honors in Computer Engineering

Reviewed and approved* by the following:

Mahanth Gowda
Assistant Professor in Electrical Engineering and Computer Engineering
Thesis Supervisor

John Sampson
Assistant Professor of Computer Science and Engineering
Honors Adviser

* Electronic approvals are on file.

ABSTRACT

Finger motion tracking from unobtrusive inertial measurement unit (IMU) methods, such as smart rings and smart watches, is becoming increasingly useful as adoption of wearables become ubiquitous. Its applications range from better evaluations of a patient's motor skills to real-time sign language translation. This paper explores two different methods for finger motion tracking—a hidden markov model (HMM) and a long short-term model (LSTM)—of the index finger metacarpophalangeal (MCP) and Interphalangeal Proximal (PIP) joints relative to the wrist using accelerometer data to predict relative position and discusses the tradeoffs of both methods. In order to overcome the challenge of limited IMU data, a synthetic IMU data preparation method was also explored.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to first thank my thesis supervisor, Mahanth Gowda, for his guidance throughout the research process. From the beginning he challenged me and provided invaluable technical help. I would also like to thank my honors advisor, John Sampson, who has helped me tremendously to stay on track throughout my time at Penn State as well as provide helpful tips toward successfully completing my thesis.

I would like to thank my parents, Catherine and Kurt Gardner, as well as my sisters, Cassidy and Caitlin Gardner, who have always supported me and are always looking out for me.

Chapter 1

Introduction

Motion tracking has many use cases in robotics for human-robot interaction (HRI) (Hua, et. al [1], Nagi, et. al [2], Gao, et. al [3], and Reddy, et. al [4]), virtual and augmented reality, visual effects, and the medical field. For example, there has been research into using motion tracking to create more quantitative measurements of a patient's motor capabilities, which normally would be a qualitative test based on a doctor's observation, as shown in the work of Kenny, et. al [5]. Finger tracking specifically, which is the focus of this paper, has even wider applications in sign language translation demonstrated by the work of Saleh, et. al [6] and Liu, et. al [7].

There are many different strategies for motion tracking such as wearables, multiple depth sensing cameras, and monocular RGB. This paper will explore the use of wearables, specifically, a smart watch and a smart ring equipped with an inertial measurement unit (IMU). Unlike pose estimation from a multi-camera setup or a single RGB frame setup, smart wearable devices are unaffected by factors such as low light, occlusion—whether by the user or another object or person—and fast intricate movements. When estimating human joint positions in 3D space using cameras or monocular RGB these factors can dramatically reduce accuracy. Additionally, they cannot be used ubiquitously as they require the user to be recorded with a camera. Smart wearables are becoming increasingly more commonplace with market penetration in the US at 22% as of 2019 [8], which makes motion tracking utilizing this technology increasingly useful.

Expanding on the work of Shen, et. al [9] and Liu, et. al [10], this paper proposes a method for building purely synthetic IMU datasets for 3D finger motion tracking. The proposed finger motion tracking solution can be split into two parts: constructing the synthetic IMU finger dataset and building the 3D finger motion tracking model. Constructing the synthetic IMU finger dataset involves extracting 3D finger and elbow position from monocular RGB videos, converting from pixels to centimeters, calculating the wrist orientation, transforming 3D finger position from the camera's coordinate system to the wrist's coordinate system (defined by the wrist's orientation), and then calculating the finger MCP joint orientation. This will provide the necessary components of IMU data: 3D acceleration and orientation of the finger MCP joint. This paper proposes two possible model architectures for 3D finger motion tracking, a LSTM and a novel solution similar to the work of Shen, et. al [9] applied to finger tracking, and discusses the tradeoffs of each architecture. I explored the aforementioned finger tracking solutions with just the index finger PIP joint relative to the wrist, but it could easily be applied to other or all PIP joints.

Chapter 2

Overview/Background

The wrist tracking solution presented by Shen, et. al [9] is a state of the art solution that requires minimal, unobtrusive sensors to predict wrist location relative to the shoulder and light-weight enough to run on a smartphone. Using the range of motion for the shoulder and elbow, they created a point cloud of possible wrist locations for each wrist orientation. Then a weighted particle filter—based on the frequencies of wrist locations for the given wrist orientation's point cloud—is applied and fed into an HMM along with wrist acceleration to predict wrist location. The search space is limited enough to produce great results due to the limited ranges of motion of the shoulder and elbow. The work presented by Shen, et. al [9] was able to achieve a median error of 9.2 cm for free-form motion, 8.3 cm for motion constrained to below the shoulder, and 5.7 cm for motion constrained to pre-defined gestures.

Given these results and the intuition that the index finger PIP joint is even more constrained relative to the wrist than the wrist is to the shoulder, I explored IMU-based finger tracking using a similar HMM-based model. Additionally, given the similarities between HMMs and LSTMs, I explored whether an LSTM could learn the relationships that are provided to the HMM in the form of the point cloud and particle filter. LSTMs often require a significant amount of data, which is limited for IMU-based tracking. The work of Liu, et. al [10] addressed limited finger IMU data for finger gesture recognition with a way to create synthetic IMU datasets using data collected from monocular RGB videos online. They used state-of-the art monocular RGB finger tracking models to generate 3D hand joint coordinates for a given frame.

Then using just those 3D points, finger acceleration and two degrees of finger orientation could be calculated. After some pre-processing steps to normalize the data, they demonstrated that the resulting synthetic acceleration and orientation data aligned well with the ground truth data. Additionally, they demonstrated that their finger gesture recognition model performed the same if not better when trained on the synthetic IMU data than the actual IMU data.

However, the work of Liu, et. al [10] was designed for finger gesture recognition not for finger tracking and cannot be directly applied to the state-of-the-art solution presented by Shen, et. al [9]. All three dimensions of wrist orientation are required to build the synthetic dataset for finger tracking, which is not addressed by Liu, et. al [10] because they utilized a coordinate system based on the user's shoulder instead of the wrist coordinate system used in this paper. This paper expands the work of Liu, et. al [10] to an adapted version of the work of Shen, et. al [9] and proposes novel solutions to 3D IMU-based finger motion tracking.

Wrist Coordinate System

The posture and motion of the fingers for a given hand are defined in the *wrist coordinate system* (WCS). The origin is at the wrist joint, the line defined by the forearm (elbow joint to wrist joint) is the x axis, the normal of the palm plane projected onto the plane with the x-axis as its normal is the z axis, and the y axis is in the direction orthogonal to the previously defined x and z axis.

Finger Joint Orientation and Location

Orientation of the finger, recorded by an IMU device such as a smart ring, records its acceleration vector in its own local coordinate system, which can be transformed into earth's coordinate system (ECS) via A^3 demonstrated in the work of Zhou, et. al [11]. Then the acceleration and orientation of the finger joint can be transformed to the wrist coordinate system, which is defined by the orientation of the wrist and can be found by applying A^3 to IMU data from a smart watch worn by the user.

Chapter 3

Synthetic IMU Dataset from Video

Instead of using real-world IMU data collected from a smart ring and a smart watch, this paper proposes creating synthetic IMU data from RGB videos. Compared to image-based finger tracking datasets, finger IMU datasets are very small. This is likely because collecting real-world IMU data is time consuming. It requires people to wear the appropriate sensors and perform hands movements. Since there are numerous large video datasets of people performing varying hand movements, if a synthetic IMU dataset could be created for finger motion tracking, the amount of data available would dramatically increase. In the work of Liu, et. al [10] a synthetic IMU dataset was created and successfully applied for hand gesture recognition. Liu, et. al [10] showed that the model trained on the synthetic dataset was able to achieve the same if not more accurate gesture classifications than the model trained on real-world IMU data.

3D Motion Tracking using Monocular RGB

The first step to building the synthetic data is collecting 3D hand joint and elbow positions from monocular RGB videos. This paper's proposed solution utilizes MediaPipe Hands created by Zhang, et. al [12] to collect 3D finger joint positions as well as MediaPipe Pose created by Bazarevsky, et. al [13] to calculate the wrist's orientation. MediaPipe Hands provides 21 3D hand-knuckle coordinates shown in figure 1 with x and y normalized to [0.0, 1.0] by the image's width and heights respectively and z representing the landmark depth with the depth at the wrist being the origin. MediaPipe Hands also predicts the handedness—left or right—of each hand that is found, which is crucial for finger joint acceleration calculations. MediaPipe Pose provides 33 3D pose landmarks shown in figure 2 with x and y normalized to [0.0, 1.0] by the image's width and height respectively and z representing the landmark depth with the depth at the midpoint of the hips as the origin.

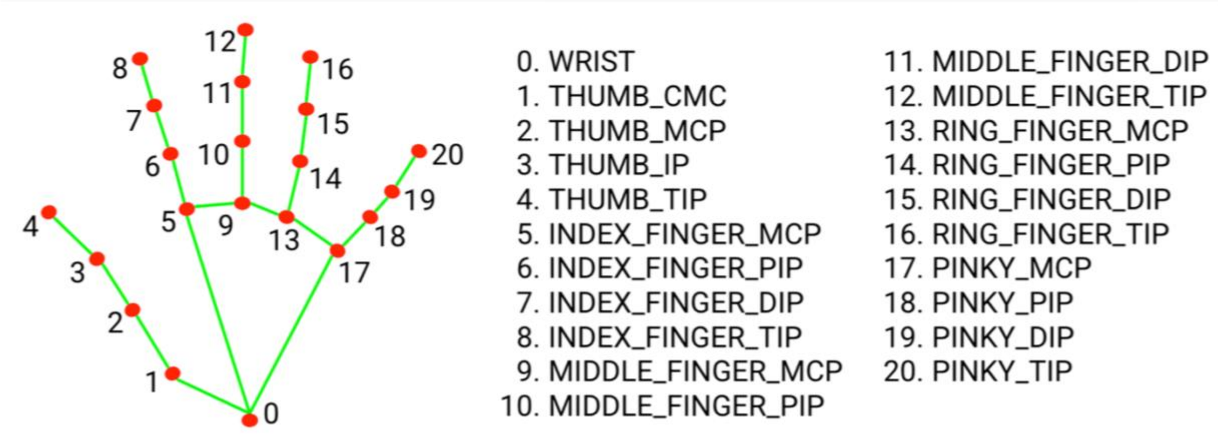

Figure 1: 21 Hand Landmarks from MediaPipe Hands [12]

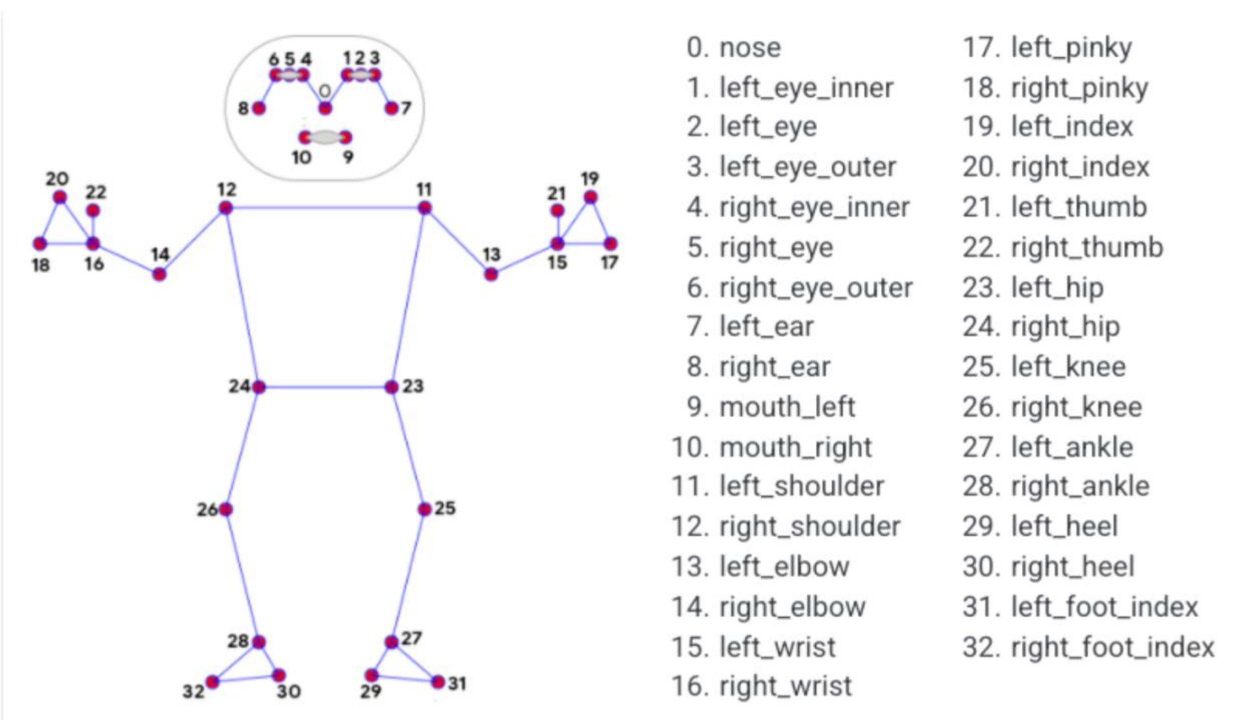

Figure 2: 33 Pose Landmarks from MediaPipe Pose [13]

Video Dataset

In working to build a synthetic dataset from RGB videos, I explored two different video datasets: IPN from the work of Benitez-Garcia, et. al [14] and the American Sign Language Large Video Dataset (ASLLVD) from the work of Athitsos, et. al [15]. Both of these video datasets were specifically created for continuous hand gesture recognition, meaning they contain long unedited videos of people performing hand movements. The IPN dataset provided a smaller subset of gestures, 13 static and dynamic, which could be an advantage if it could be used to build a motion tracking model optimized for a subset of hand gestures. However, the participant's elbow is frequently outside the camera's frame, which renders it impossible to calculate the wrist's orientation and therefore impossible to transform the finger joints' orientations and positions into WCS. However, IPN was still used to explore a synthetic dataset of only 3D acceleration vectors. ASLLVD provides numerous advantages over IPN: it contains thousands of different ASL signs, the participant's elbow is always in the camera's frame, a majority of participant's body is in the camera's frame—a factor in the accuracy of MediaPipe Pose—and each video is hosted on a server at the University of Texas at Arlington, which allows the videos to be read in without downloading the whole dataset. Figures 3 and 4 show the outputs of MediaPipe Hands on a video frame from IPN and ASLLVD.

Figure 3: MediaPipe Hands [12] on IPN dataset [14] video frame

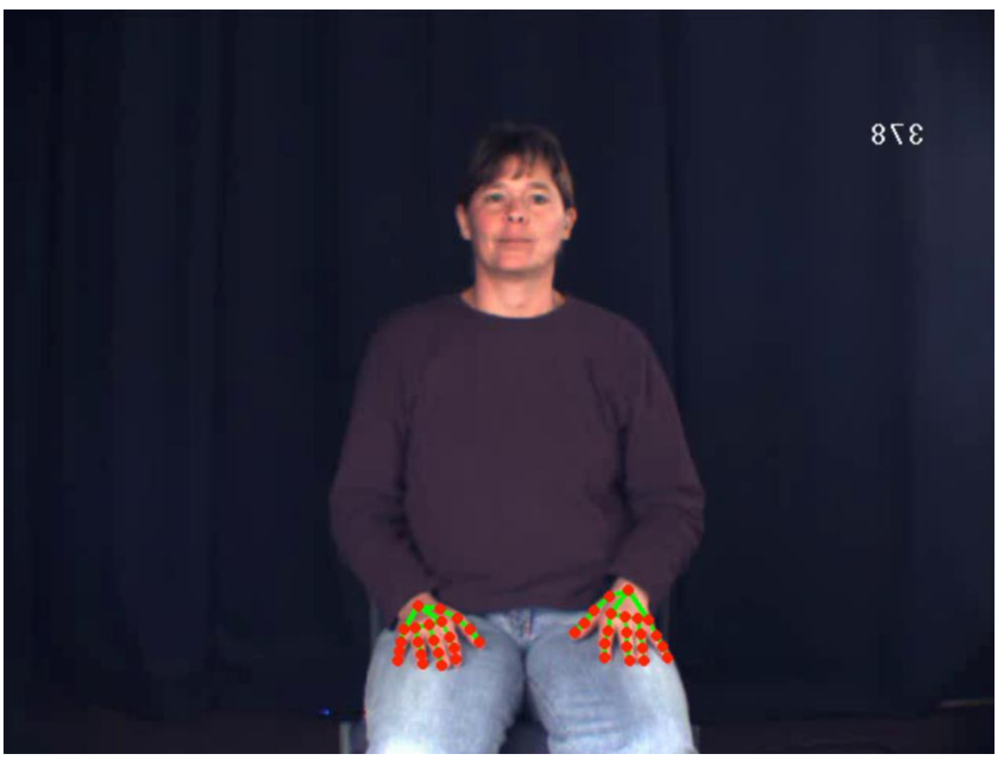

Figure 4: MediaPipe Hands [12] on ASLLVD dataset [15] video frame

Estimated Orientation

Once the 3D finger and elbows positions are collected—in the camera's coordinate system as described above—for each frame using MediaPipe Pose, the orientation of the wrist can be calculated as follows. The x component of wrist orientation is found by normalizing the wrist to elbow vector. Then the z component of wrist orientation is calculated by projecting the normal of the palm plane onto the plane with the x component of wrist orientation as its normal. Finally, the y component of wrist orientation is the cross product of the x and z component of wrist orientation. A visual representation of this relationship can be seen in figures 5-7 where the blue dots represent the index finger MCP, pinky finger MCP, wrist, and elbow joints, the purple vector represents the normal to the palm plane centered at the wrist, and the green vectors represent the x, y, and z orientation vectors of the wrist.

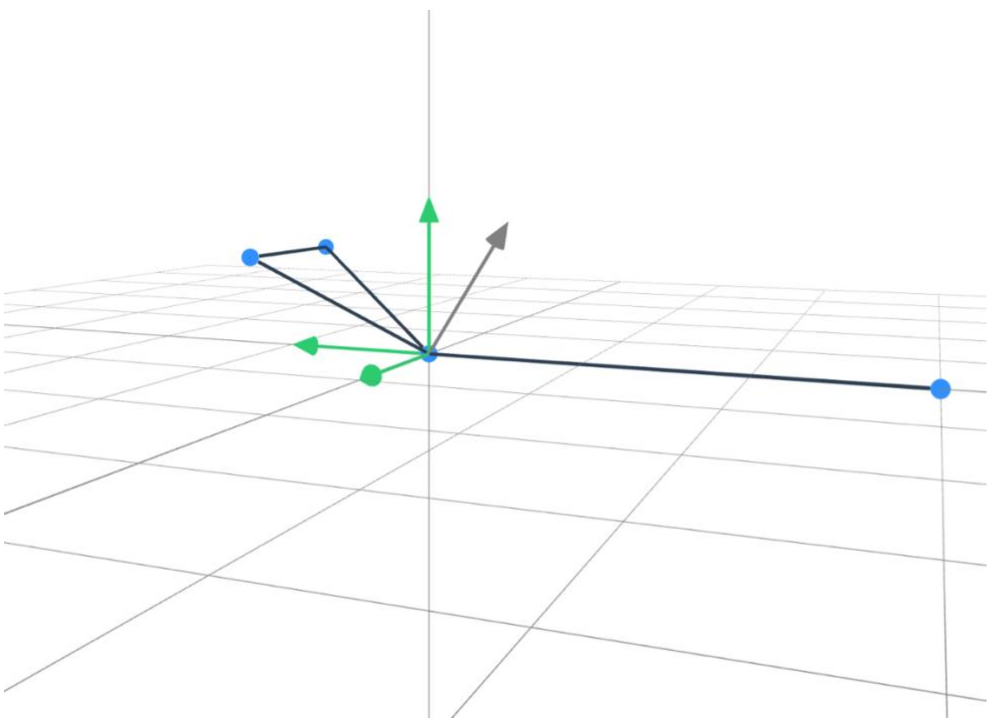

Figure 5: Simple WCS Example

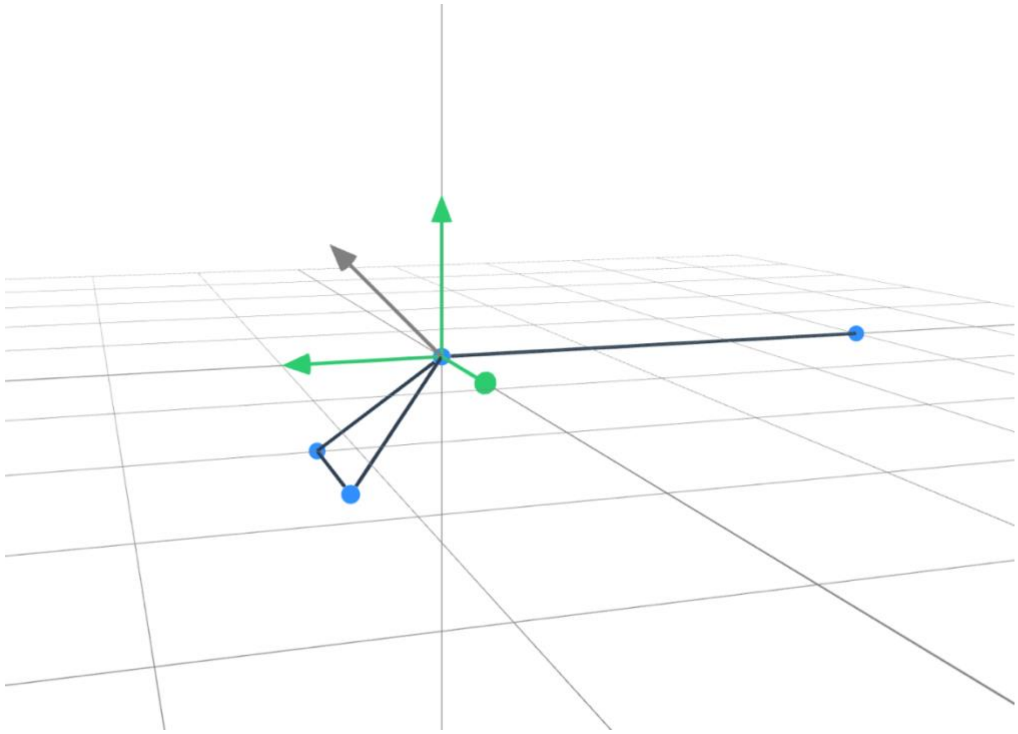

Figure 6: Another WCS Example

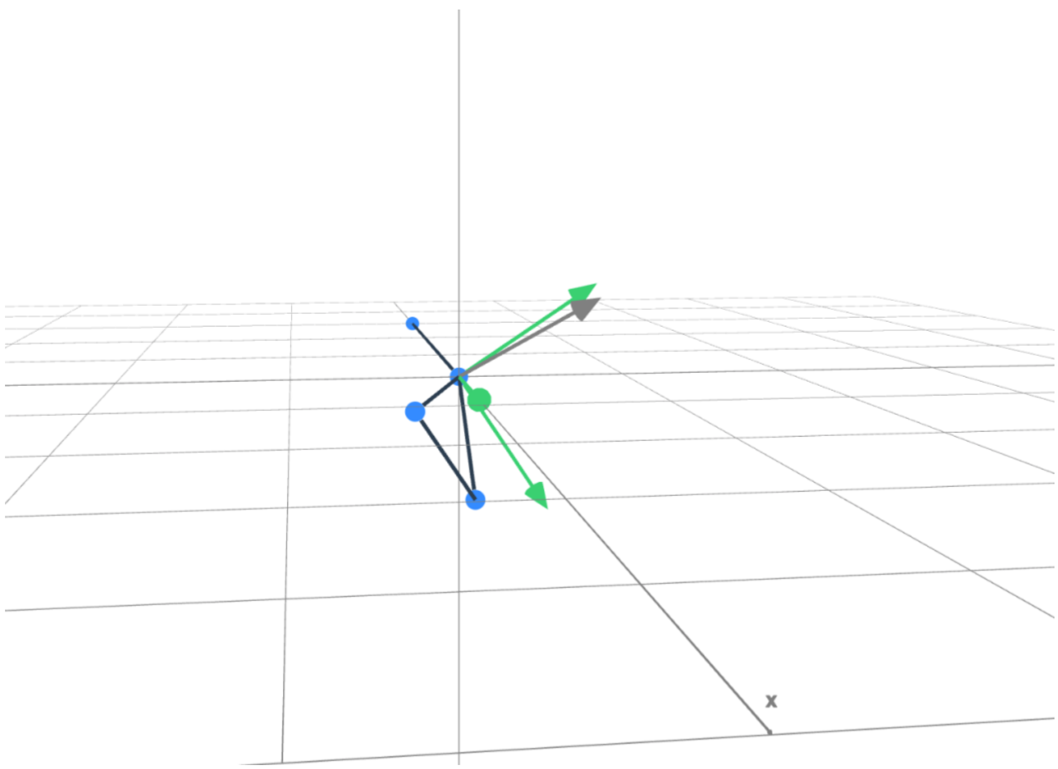

Figure 7: Complex WCS Example

The green wrist orientation vectors were calculated with the following equations. Equation (1) calculates the x component of wrist orientation from j_1, which is the elbow vector relative to the wrist.

$$\hat{x} = \frac{\overrightarrow{J_1}}{||\overrightarrow{J_1}||} \quad (1)$$

Equation (2) calculates the normal vector to the palm plane where j_3 is the pinky finger MCP joint relative to the wrist and j_4 is the index finger MCP joint relative to the wrist.

$$\widehat{N} = \overrightarrow{J_3} \times \overrightarrow{J_4} \quad (2)$$

Equation (3) calculates the z component of wrist orientation utilizing both (1) and (2).

$$\hat{z} = \begin{cases} \text{Undefined} & (\widehat{N} \cdot \hat{x}) = \text{-1 or 1} \\ \widehat{N} - (\widehat{N} \cdot \hat{x})\hat{x} & \text{otherwise} \end{cases} \quad (3)$$

Finally, equation (4) calculates the y component of wrist orientation utilizing both (1) and (3).

$$\hat{y} = \hat{x} \times \hat{z} \quad (4)$$

It is important to note that when the wrist is at 90 degrees flexion, the orientation of the wrist cannot be calculated. There is not enough information to calculate the z component of wrist orientation if the normal to the palm plane is parallel to the x component of wrist orientation. Fortunately, this is the only hand posture that leads to an undefined wrist orientation and it is likely not a common hand posture.

Estimated Acceleration

Once the orientation of the wrist is calculated for each frame as well as the 3D finger joint positions from MediaPipe Hands, the acceleration can be calculated for each frame. First the 3D finger joint positions are transformed from the camera's coordinate system to WCS. Then the 3D finger joint positions are scaled such that the palm is 3 cm, which should enable better generalizations to multiple body sizes and not having camera parameters as demonstrated by Liu, et. al [10]. Finally, the instantaneous acceleration is estimated using second-order central finite difference for frame 2 through N-1 where N is the number of frames in the video shown in (5).

$$a = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \quad (5)$$

Acceleration data collected through this technique is often very noisy, therefore, I apply a Butterworth low-pass filter over the acceleration measurement to smooth the data and correct some of the noise.

Chapter 4

Model Architecture

I explored both a LSTM approach as well as a Hidden Markov Model following a similar approach as Shen, et. al [9] but for tracking the PIP joint of the hand rather than the wrist.

Hidden Markov Model

Shen, et. al [9] demonstrated a model to track the location of the wrist relative to the shoulder using acceleration and orientation from an IMU worn on the wrist. The model utilized the fact that in the case where the elbow stayed in one location, there was a one-to-one relationship between the orientation of the wrist and its location relative to the shoulder. Shen, et. al [9] showed that because of this fact, when allowing the elbow to move, the possible relative locations for the wrist for a given orientation were quite limited. This meant that the search space was small enough to enable a hidden markov model to be trained to predict the location based on previous orientations as well as acceleration. This same idea can be applied to tracking the finger PIP joint relative to the wrist where the finger MCP joint is comparable to the elbow joint in the work of Shen, et. al [9]. The finger PIP joint is a static shift in the x direction from the finger MCP joint. There is an additional constraint that reduces the search space for the finger tracking solution, however, since the palm cannot rotate independently from the wrist. This means there are only four degrees of freedom present for the solution presented in this paper.

Point Cloud Construction

Just as Shen, et. al [9] mapped each watch orientation to each possible elbow and wrist locations relative to the shoulder, I explored mapping each index finger PIP orientation to each possible index finger PIP and MCP locations relative to the wrist. Given that orientation and location are in WCS, the hand cannot rotate independently from the wrist, and the index finger cannot rotate independently from the palm, there are only four degrees of freedom: θ_1, θ_2, θ_3, and θ_4. Using the Denavit-Hartenberg transformations, the complete hand posture up to the PIP joint can be mathematically calculated. Equations (6) – (8) describe the Denavit-Hartenberg transformations.

$$T_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} (6)$$

$$T_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & a \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} (7)$$

$$T_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & a \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} (8)$$

These equations can be combined using the θ_1, θ_2, θ_3, and θ_4 to calculate transformation from wrist to MCP and PIP joints using (9) and (10).

$$T_{MCP} = T_z(\theta_2)\, T_y(\theta_1) (9)$$

$$T_{PIP} = T_z(\theta_2)\, T_y(\theta_1) T_z(\theta_4)\, T_y(\theta_5) (10)$$

The expanded equations are left out due to the size.

Long Short-Term Model

I explored two different LSTM models, one trained on just index finger PIP acceleration and another trained on index finger PIP acceleration and orientation, and I implemented the former. For the LSTM model that I implemented based solely on acceleration, I explored many different LSTM models varying the number of LSTM layers from 1 to 5, the number of neurons in each layer from 16 to 1024, the learning rate from 0.1 to 0.0001, the loss function between mean squared error (MSE) and logcosh, batch size from 1 to 100, and activation functions such as relu, sigmoid, and tanh. None of the LSTM models trained solely using acceleration to predict location performed very well as demonstrated in chapter 5.

Tradeoffs

HMMs are much simpler than LSTMs. Both LSTMs and HMMs capture sequential information in data, however, HMMs are linear models while LSTMs apply a point-wise non-linearity to the output. HMMs work under the Markovian assumption that the current state depends only on the previous state. While this assumption is likely not completely accurate, by including a few prior acceleration and orientation measurements into a given state, this has shown to provide accurate motion tracking for wrist tracking by Shen, et. al [9]. LSTMs, however, are non-linear and not bound by this Markovian assumption. They can learn long, distant dependencies. LSTMs therefore are more powerful than HMMs because they can learn hidden relationships from data, which are then factored into the final predication. For example, an LSTM trained on finger motion data of a person waving could first learn the repetitive pattern of a wave, but then could learn than that the likelihood of the pattern breaking increases the longer the person waves. This added complexity, however, makes LSTMs significantly harder to train. They often require a lot of data as well as an arduous hyper-parameter search to find the right combinations that enable the model to learn the appropriate relations and perform well on unseen data. For example, an HMM significantly outperformed an LSTM for action classification in the work of Alp, et. al [16], when the amount of training data was limited. Additionally, an LSTM trained for rainfall-runoff modeling from the work of Boulmaiz, et. al [17] required at least nine years of historical database measurements with its performance increasing up to 12 years of historical database measurements. In many studies, the amount of IMU training data is very small compared to vision-based finger motion tracking. For example, in the work of Mummadi, et. al [18] there were 22,000 samples and in the work of Kim, et. al [19] there were a total of 1,000 total hand gestures performed. Vision-based finger motion tracking datasets, however, have up to five million samples such as in the work of Malik, et. al [20]. The synthetic dataset preparation method proposed in this paper toward the application of finger motion tracking hopes to allow for significantly larger IMU datasets, which could lead to better results using an LSTM.

Chapter 5

Results

I explored many different LSTMs trained on just synthetic acceleration data—without orientation—and found none of the models performed well, which could be due to not enough data, but most likely because acceleration alone simply is not enough to make a prediction about current position. It performs better than taking the double integral over a series of accelerations, which has an error that quickly grows without a bound. The LSTM trained on acceleration alone at least has bounded error, but does not predict the finger's location well. Figure 8 shows the predicted output versus the ground truth for the test data, which can be seen to perform very poorly. In the future we hope to utilize the synthetic data preparation method described in this paper to build significantly larger datasets than what was used to train this basic LSTM. Additionally, we hope to implement the above calculations for wrist orientation in order to add this information to the training dataset, which should allow for better predictions from the LSTM and allow us to implement the HMM described above.

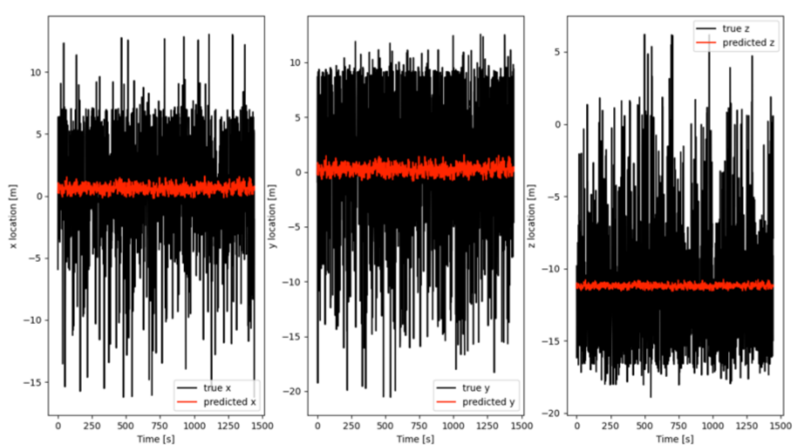

Figure 8: LSTM Output

Chapter 6

Related Work

As previously mentioned, motion tracking has been explored using many different approaches in addition to IMU-based tracking such as monocular RGB and depth sensing cameras (RGB-D). Both methods are used to track finger motion similar to IMU-based tracking, but utilize image input data instead of acceleration and orientation data in order to make the prediction. Monocular RGB based tracking was used in this paper in order to build synthetic IMU data. However, predicting depth from monocular RGB data can be challenging, which RGB-D based tracking addresses by including depth information as an input for the prediction.

Monocular RGB

Monocular RGB-based motion tracking uses linear statistical models and machine learning models to predict 3D poses from a single image as shown in the works of Zhang, et. al [12], An, et. al [21], Sharma, et. al [22], and He, et. al [23]. This is often applied to each frame of a video, similar to the process described earlier, in order to gain 3D joint positions from a video.

Depth Sensing Cameras

Motion tracking utilizing depth sensing cameras—demonstrated in the work of Zhang, et. al [24]—builds on the work on monocular RGB-based models, but looks to achieve higher depth accuracy by measuring depth directly. Depth-based hand tracking methods could be explored as an alternative to monocular RGB in the solution presented within this paper, however, there are considerably less RGB-D videos available as compared to RGB videos.

Chapter 7

Future Work

In addition to HMMs and LSTMs there are numerous other models that could perform well on a sequential data, such as simpler linear regressions (ARIMA) or other complex deep neural network architectures (Convolutional Neural Networks). There have been studies showing that adding CNN layers to LSTM models for time sequential data tasks can improve accuracy as shown in the work of Mehtab, et. al [25]. Additionally, more work should be done on evaluating the quality of the training data gathered from individual videos in order to automatically discard videos that lead to particularly noisy or poor acceleration/orientation data.

Chapter 8

Conclusion

This paper explored using an HMM and an LSTM for finger motion tracking to predict 3D finger position relative to the wrist from IMU data of a smart ring and a smart watch as well as discussed tradeoffs between both methods. Additionally, a synthetic IMU data preparation method was proposed to substantially increase the training data available for IMU-based finger motion tracking. I hope that these methods in the future can be fully implemented and tested to validate the intuition and mathematics presented in this paper. Then I hope it is applied to critical areas such as sign language translation or improved evaluations of patients' motor skills.

Appendix A

Keywords

ASL: American Sign Language

ASLLVD: American Sign Language Large Video Dataset

CNN: Convolutional Neural Network

ECS: Earth's Coordinate System

HRI: Human Robot Interaction

IMU: Inertial Measurement Unit

LSTM: Long Short-Term Memory RNN

MCP: Metacarpophalangeal Joint of the finger

Monocular RGB: A color image containing red, green, and blue channels, but unlike RGB-D, has no depth component

PIP: Interphalangeal Proximal Joint of the finger

RGB-D: A color image containing red, green, and blue channels as well as a fourth channel with depth information

RNN: Recurrent Neural Network

WCS: Wrist Coordinate System

BIBLIOGRAPHY

[1] Hua, Wang, Huang Xiaodiao, and He Jin. “Finger Tracking Control of Underwater Dexterous Hand Based on Fuzzy CMAC.” 2007 Chinese Control Conference, 2006, 121–24. https://doi.org/10.1109/chicc.2006.4347044.

[2] Nagi, Jawad, Alessandro Giusti, Gianni A. Di Caro, and Luca M. Gambardella. “Human Control of UAVs Using Face Pose Estimates and Hand Gestures.” Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction, 2014, 1–2. https://doi.org/10.1145/2559636.2559833.

[3] Gao, Qing, Jinguo Liu, Zhaojie Ju, and Xin Zhang. “Dual-Hand Detection for Human–Robot Interaction by a Parallel Network Based on Hand Detection and Body Pose Estimation.” IEEE Transactions on Industrial Electronics 66, no. 12 (2019): 9663–72. https://doi.org/10.1109/tie.2019.2898624.

[4] Reddy, Vantukala VishnuTeja, Thumma Dhyanchand, Galla Vamsi Krishna, and Satish Maheshwaram. “Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition.” 2020 IEEE-HYDCON, 2020. https://doi.org/10.1109/hydcon48903.2020.9242677.

[5] Chen, Kenny, Paolo Gabriel, Abdulwahab Alasfour, Chenghao Gong, Werner K. Doyle, Orrin Devinsky, Daniel Friedman, et al. “Patient-Specific Pose Estimation in Clinical Environments.” IEEE Journal of Translational Engineering in Health and Medicine 6 (2018): 1–11. https://doi.org/10.1109/jtehm.2018.2875464.

[6] Saleh, Nour ElHoda, Bilal Debs, Elianne Moussallem, Ahmad Rifai Sarraj, and Soha Saleh. “Home-Based Therapy of Chronic Stroke Survivors' Upper Limb Assisted by a Finger Movement Tracking Device.” 2015 International Conference on Advances in Biomedical Engineering (ICABME), 2015. https://doi.org/10.1109/icabme.2015.7323264.

[7] Liu, Yilin, Fengyang Jiang, and Mahanth Gowda. “Finger Gesture Tracking for Interactive Applications.” Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 4, no. 3 (2020): 1–21. https://doi.org/10.1145/3414117.

[8] Wurmser, Yoram. “Wearables 2019.” Insider Intelligence. Insider Intelligence, January 3, 2019. https://www.emarketer.com/content/wearables-2019#page-report.

[9] Shen, Sheng, He Wang, and Romit Roy Choudhury. “I Am a Smartwatch and I Can Track My User's Arm.” Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services, 2016. https://doi.org/10.1145/2906388.2906407.

[10] When Video meets Inertial Sensors: Zero-shot Domain Adaptation for Finger Motion Analytics with Inertial Sensors. Yilin Liu, Shijia Zhang, Mahanth Gowda. IoTDI, 2021.

[11] Zhou, Pengfei, Mo Li, and Guobin Shen. “Use It Free: Instantly Knowing Your Phone Attitude.” Proceedings of the 20th annual international conference on Mobile computing and networking, 2014. https://doi.org/10.1145/2639108.2639110.

[12] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. "MediaPipe Hands: On-device Real-time Hand Tracking." 2020.

[13] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. "BlazePose: On-device Real-time Body Pose tracking." 2020.

[14] Benitez-Garcia, Gibran, Jesus, Olivares-Mercado, Gabriel, Sanchez-Perez, and Keiji, Yanai. "IPN Hand: A Video Dataset and Benchmark for Real-Time Continuous Hand Gesture Recognition." In 25th International Conference on Pattern Recognition, ICPR 2020, Milan, Italy, Jan 10–15, 2021 (pp. 4340–4347). 2021.

[15] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, Q. Yuan and A. Thangali, The ASL Lexicon Video Dataset, CVPR 2008 Workshop on Human Communicative Behaviour Analysis (CVPR4HB'08)

[16] Alp, Elit Cenk, and Hacer Yalim Keles. “A Comparative Study of HMMs and LSTMs on Action Classification with Limited Training Data.” Advances in Intelligent Systems and Computing, 2018, 1102–15. https://doi.org/10.1007/978-3-030-01054-6_76.

[17] Boulmaiz, T., M. Guermoui, and H. Boutaghane. “Impact of Training Data Size on the LSTM Performances for Rainfall–Runoff Modeling.” Modeling Earth Systems and Environment 6, no. 4 (2020): 2153–64. https://doi.org/10.1007/s40808-020-00830-w.

[18] Mummadi, Chaithanya, Frederic Leo, Keshav Verma, Shivaji Kasireddy, Philipp Scholl, Jochen Kempfle, and Kristof Laerhoven. “Real-Time and Embedded Detection of Hand Gestures with an IMU-Based Glove.” Informatics 5, no. 2 (2018): 28. https://doi.org/10.3390/informatics5020028.

[19] Kim, Minwoo, Jaechan Cho, Seongjoo Lee, and Yunho Jung. “IMU Sensor-Based Hand Gesture Recognition for Human-Machine Interfaces.” Sensors 19, no. 18 (2019): 3827. https://doi.org/10.3390/s19183827.

[20] Malik, Jameel, Ahmed Elhayek, Fabrizio Nunnari, Kiran Varanasi, Kiarash Tamaddon, Alexis Heloir, and Didier Stricker. “DeepHPS: End-to-End Estimation of 3D Hand Pose and Shape by Learning from Synthetic Depth.” 2018 International Conference on 3D Vision (3DV), 2018. https://doi.org/10.1109/3dv.2018.00023.

[21] Shan An, Xiajie Zhang, Dong Wei, Haogang Zhu, Jianyu Yang, and Konstantinos A. Tsintotas. "FastHand: Fast Hand Pose Estimation From A Monocular Camera." 2021.

[22] Sharma, Sanjeev, and Shaoli Huang. “An End-to-End Framework for Unconstrained Monocular 3D Hand Pose Estimation.” Pattern Recognition 115 (2021): 107892. https://doi.org/10.1016/j.patcog.2021.107892.

[23] Yiming He, Wei Hu, Siyuan Yang, Xiaochao Qu, Pengfei Wan, and Zongming Guo. "3D Hand Pose Estimation in the Wild via Graph Refinement under Adversarial Learning." 2020.

[24] Zhaohui Zhang, Shipeng Xie, Mingxiu Chen, and Haichao Zhu. "HandAugment: A Simple Data Augmentation Method for Depth-Based 3D Hand Pose Estimation." 2020.

[25] Mehtab, Sidra, Jaydip Sen, and Subhasis Dasgupta. “Robust Analysis of Stock Price Time Series Using CNN and LSTM-Based Deep Learning Models.” 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2020. https://doi.org/10.1109/iceca49313.2020.9297652.

ACADEMIC VITA

Kevin A. Gardner

kgardner2112@gmail.com | Philadelphia, PA

Education

The Pennsylvania State University | Schreyer Honors College — **University Park, PA**
College of Engineering | Bachelor of Science in Computer Engineering — *Class of May 2021*

Professional Experience

Apple Inc. — **Wallingford, Pa (Virtual)**
Incoming Software Engineer — *May 2020 – Aug 2020*

The Boeing Company — **Seattle, WA**
Emergent Technology Intern | Boeing Global Services — *May 2019 – Aug 2019*

- Developed Boeing's first iOS Swift Package for standard use in many production iOS applications, saving a month of development time per app
- Supported a project with Sales and Marketing that is projected to save the company $2 million per year

Safe Catch — **Sausalito, CA**
Sales/Marketing Intern (2018) | Marketing Intern (2016) — *Jun 2016 – Aug 2018*

- Quantified the interest of celebrities using an Excel spreadsheet, ultimately advising the company to switch their strategy for attaining celebrities and saving the company $5,000
- Promoted and sold Safe Catch's new flavored tuna pouches, canned ahi, and canned salmon to local retail stores throughout Pennsylvania, New Jersey, and Delaware, closing the sale in 5 stores

Leadership and Activities

THON AR — **University Park, PA**
iOS Development Team Lead — *Nov 2018 – Present*

- Utilized Apple's ARKit image detection and CloudKit to detect images of THON directors, bringing these images to life throughout the hallways of the Penn State Hershey Medical Center
- Managed development team of 5 students building an augmented reality iOS app in order to bring the magic of the Panhellenic Dance MaraTHON to children hospitalized THON weekend

Penn State Unmanned Aerial Systems — **University Park, PA**
Grounds Control Team Member — *Aug 2018 – Present*

- Designed an algorithm to find the GPS coordinates of a target on the ground from its location in an image and the GPS coordinates, altitude, and heading of a drone
- Assisted in the development of a website to receive images from a drone, classify them, and submit them to a judge's server

Penn State IFC/Panhellenic Dance Marathon — **University Park, PA**
Dancer Relations Committee Member | Family Relations Liaison — *Sep 2018 – Feb 2019*

- Ensured the safety and well-being of the dancers by supporting them both physically and emotionally throughout THON's 46-hour no sitting no sleeping dance marathon
- Instructed the committee on the guidelines for interacting with THON families while also acting as primary contact to maintain a relationship with a past THON family

Penn State Advanced Vehicle Team Eco Car Competition — **University Park, PA**
Controls Team Member — *Aug 2017 – May 2018*

- Remodeled regenerative braking algorithm for a converted hybrid electric Camaro in order to allow the algorithm to support multiple gears in a manual car
- Analyzed algorithms under Dyno mode conditions to ensure the car would function properly in competition

Alpha Kappa Psi Professional Co-ed Business Fraternity — **University Park, PA**
Webmaster (2019) | Fundraising Chair (2018) — *Feb 2018 – Present*

- Conducted educational Python coding workshop for an attendance of 29 members from the brotherhood to educate members about the benefits of learning a programming language and teach coding fundamentals

Relevant Coursework

Object Oriented Programming with Web-Based Applications (Java)	Circuits and Devices
Introduction to Computer Architecture	Digital Design: Theory and Practice
Data Structures and Algorithms	Computer Organization and Design

Skills

Skills: Advanced: Python, Java, Swift, Data Structures & Algorithms, Beginner: HTML, CSS, Verilog, 3D Modeling (Blender)