Athlete Motion Tracking:

Design Document

Team Number: sdmay19-01

Client: Precision Performance Cycling (Nathan Johnson)

Adviser: Craig Rupp

Software Engineers: Monte Friestad and Nathan Mazarelo

Hardware Engineer: Ryan Hansen

Project Manager: Maddie Rogers

Email: sdmay19-01@iastate.edu
Website: http://sdmay19-01.sd.ece.iastate.edu/

Table of Contents

List of figures/tables/symbols/definitions

Terms

Bike Fitting: The way an athlete sits on a race bike to achieve optimal angles. The body position affects the athlete's center of mass which is to be positioned properly over the frame and handlebars. There are many different variables on a bike that can be changed. These include but are not limited to: the frame size, the seat height, the x, y, and z variables on the seat position, the seat width, the handlebar angle, the handle bar length , and the pedal size. All of these facets are related to how the athlete sits on the bike. The optimal position depends on the athlete. The ratio of legs, torso, and arms all contribute to the ideal position and angles.

Bike Fitting by Eye: Determining the angles such as hip angle or knee angle by eye. This means using rough estimates to create a recommendation. Sometimes a certain angle can be too heavily considered while others are not considered enough.

Lab: A bike is ridden on a set up that suspends the tires on rollers. This allows for the rider to stay stationary which means that training can occur indoors. The lab will allow for monitored training to improve position.

Real Life Training: Athletes train outdoors over varying distances. This allows them to change altitude or other riding conditions.

1 Introduction

1.1 Acknowledgement

The Athlete Tracking Team thanks Nathan Johnson, of Precision Performance Cycling, for his time, financial aid, technical experience, and equipment. As the sole proprietor and employee of Performance Cycling, the communication is directly through Nathan. He provides strong examples and clear direction for the project.

1.2 Project Statement

The purpose of this project is to create a system that decreases error compared to current available systems. This system will combine multiple electronic systems to remove any biases from the current method of bike fitting which is done by eye.

The design of this project is required to assist in digitally accessing the body position. The client is currently fitting the bike to the athlete by eyeballing it. This leaves room for error and biases while restraining the athlete to a stationary bike. There is not a current system on the market that encompasses both image processing and digital sensor analysis.

The project will encompass two different systems: a video system and a sensor hardware system. These two systems will need to operate both in an indoor environment as well as outdoors for training and races. The data will be saved for coaches and athletes to analyze at a later time - not live feed. The client requested this so the athlete will not change their performance during the race according to the data collected.

In addition, the different angles will be presented on the web application but not analyzed. The angle between the points will be reported, such as the hip or knee angle. The system will not determine whether that angle is ideal and if it needs to change, instead it will leave the analysis up to the coach.

This project will look to alleviate many of the different issues that current models are missing. This begins by leaving the analysis to the coach. Currently, the systems are telling the coaches how to change the body positions and bike fittings. This approach discredits each athlete's differences. Further, these systems operate outside of the allotted 2.5% error. This project will aim to operate within the error margin. The current systems also don't use multiple systems at once, many use just the sensors or just the image capturing.

The future outlook for this project will be to use this system for running as well. The expansion into running will also allow for the expansion of the company. This change will require minimal hardware changes with more software changes.

1.3 Operating Environment

The first system will be operating indoors with minimal externalities. The cameras and software interfacing will need to occupy a small enough space that being indoors is feasible. The lighting indoors will also need to be well lit in order for the cameras and software to capture the athlete.

The hardware system will be used outside during races and training. The locations for the sensors will be dynamic (such as the foot), needing them to be wireless. In addition, the sensors will need to be waterproof and durable for long races or training sessions in the rain. The other aspect is the moisture. The sensors will be attached to enclosed areas of the rider such as shoes or gloves. As the athlete continues to bike, the moisture will build up and be retained within the enclosure.

1.4 Intended Users and Intended Uses

The end consumer will be the athlete. The athlete will be hooked into the system and will have the data collection based on their performance. The customer will be the coach, in this case, Nathan. Therefore, we need to cater the uses to making it useful for the coach while also making sure the uses are friendly and compatible with the athlete.

The end consumer has a strong background in cycling and understands their comfort level. This system is used to collect data from a real life scenario and turn it into a visual representation that can be read and understood. Therefore, collecting the data must not interfere with the athlete.

The end data must be presented in a manner that is decipherable. The coach and athlete will take the data and analyze it on their own, only needing the angles and pressure values reported.

The raw data will also need to be available. The main concern of the client is to view any further pieces of data if an anomaly were to occur.

1.5 Assumptions and Limitations

The first assumption is that the sensors and cameras will collect data fast enough to account for the biking and running rpms at 107 and 170, respectively. This is important to create the proper amount of points to avoid skewing the data. The next assumption is that the systems can be put on multiple different athletes and not have the analysis skewed.

The limitations vary. The first limitation is that the sensors are wireless. Improperly fitted sensors affect how the rider is positioned on bike making the data not representative of what is actually happening. The next limitation is the price of the camera. While the price of the camera was not specifically laid out, many cameras that are used for athlete body position are in the tens of thousands of dollars. In addition, the area needed for these cameras to operate are usually quite large and they need to be compatible with the physical area Nathan has available.

1.6 Project Deliverables

There are two deliverables that are expected: the image capturing system and the hardware sensor system. The project is split into two very intense sections based on these two needs.

The first deliverable encompasses the image processing system which includes choosing the camera, collecting the data, analyzing it, and then presenting it in a user friendly manner. The deliverable is a working system that includes cameras and a web app for the consumers to use. Further, there is the background software that is not noticed that is still included.

The second deliverable is the hardware system and the app that will relay the info from the system. The hardware will include the sensors and a transmitting device that will allow for the data to be sent from the wireless sensors.

1.7 Deliverable Goals

The main goal with the first deliverable is a fully integrated system from the cameras to the data presentation while operating within the 2.5% error. This goal will be met based on passing the testing implementations.

The goals associated with the second deliverable begin with acquiring the wireless sensors and assuring their communication with the transponding device. The next goal is analyzing the data. This goal will be accomplished when the data is received and the angles of the athlete are measured. The final goal will be getting the data to the aforementioned web application.

2 Specifications and Analysis

2.1 Functional Requirements

The functional requirements for this project are properly working cameras and image processing software. These are vital for any functionality in the system. Cameras and the software are needed to get the data from the athlete into a database. This will allow for any further processing to be done.

The second functional requirement is working sensors that interface with a transponder to get the data to a computer. These are needed in order to gain any data from the ride. Without this functionality no data can be pulled from sensors or analyzed. This means that any further additions or extensions of the project would have inadequate points to draw conclusions.

2.2 Non-Functional Requirements

The non-functional requirements for this project is the web application where both systems relay information to the display. This application and user interface are non-functional as the data will already be being exported to MatLab and Excel. The application just makes the data easier to view and understand.

2.3 Proposed Design/Method

The proposed design follows the block diagram outlined below.

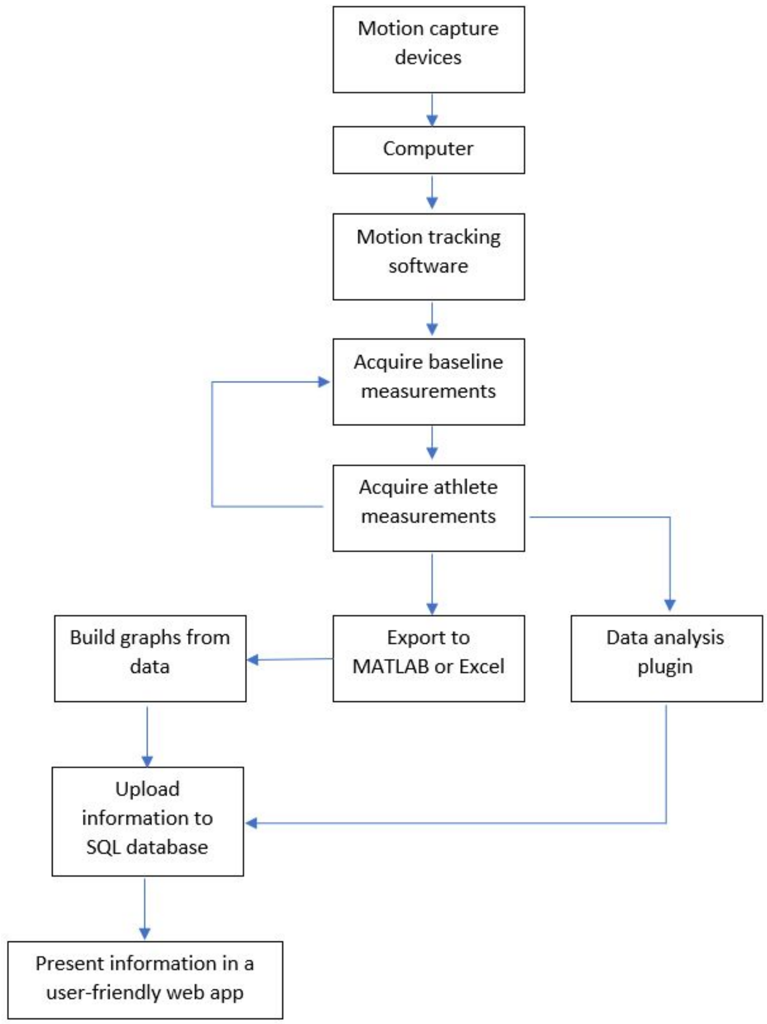

Figure 1. Project Block Diagram

2.3.1 Image Capturing

The two largest limitations of the camera are the budget and the frames per second. The final apparatus will have a minimum of three cameras set up. This means that the cameras alone cannot cost too much as these are just one system in the final project. The second large limitation is the frames per second. This is an issue due to high rpms that the athlete will achieve while pedalling. In order to capture a full cycle, the camera must function quick enough to pick up twelve points in a cycle.

The camera being used is a 1080p camera with 120 fps. The specific camera model is the circuit board and lens. This will keep the camera minimal and keep the costs down. Four cameras will be ordered. There will be two side cameras, a front camera, and one spare camera.

The cameras will be mounted on a tripod. This will allow for consistency and minimal calibration. The cameras mentioned above will be attached to the tripod and fixed to a certain height.

The camera is currently being tested against a stationary bike. This will allow us to determine whether it can pick up the required rpms of the biker. Further, it is also allowing us to see the built in capabilities. This is a step our team is taking very seriously as it is a functional requirement. In order for us to progress, the camera must meet all outlined standards as it is the only input for this phase of the data.

Figure 2. 120 fps Camera

2.3.2 Image Processing Software

The next step is the image processing software. There are many different software approaches to this step. Many times the packages will come together with a camera. This is an expensive option. The alternative to this is iPi. This is a software package that analyzes an image based on the camera input [1].This software was rather cheap compared to the other large name packages and had minimal renewal options, making this easier for the client.

This software also exports the data to both MatLab and Excel which was a high desire of the client. The iPi software also creates an appealing data dashboard that is outlined in Figure 3. This data is not in the final form that is needed for the web application. Instead the data will be extracted from the program and used for the web application with iPi serving as the raw data.

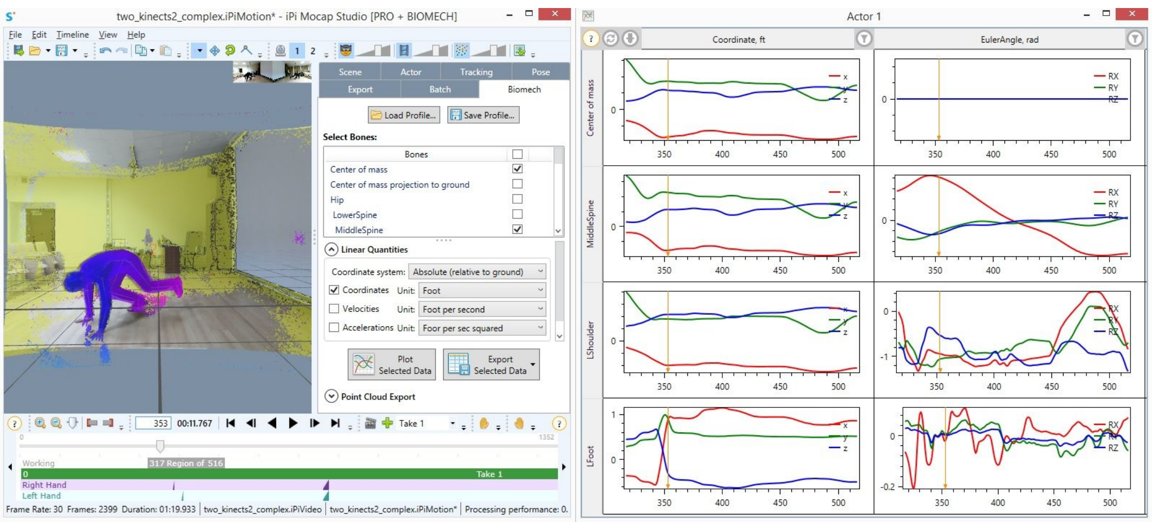

Figure 3. iPi Software

2.3.3 Analytics and Computation Software

There are many different options for a software to use. Our team chose Python for a multiple reasons. The software engineers have some experience in this language. However, the library capability that Python offers are vast for images and processing camera feeds. We are looking into libraries such as PIL, Pillow, and Scripty. The final reason we like Python is its interfacing ability with the GUI. Below is a picture that gives meaning to what the web application is tracking. In order to remain within the 2.5% error it is imperative that the equations are correct. If they are not correct, the coach will see incorrect issues and try to resolve them.

Figure 4. Important Body Points to Track

2.3.4 User Interface

The end goal of our project is to emulate a web app that displays the data after the ride. The first part of the web application comes with calculations from the video software. The software we are using gives important numbers, however we will need to do some calculations to get the values into a more compact and understandable form so that the coach or athlete can understand what happened during a session.

An issue that arose is data storage. Due to HIPAA laws, there is potential concern with storage of athlete data [2]. Therefore, when we use this application and store the data, we need to make sure that all of the computers that we are using have encrypted hard drives and that any data that we will be transmitting between our device and any other device to a remote database needs to be encrypted. Therefore, for this project, we will need to be careful.

The last part of the project is the GUI. This will be used for the athlete, coach or trainer to review session data during and after a session. The main goal of this project is to compare progress and see changes in an athlete over time. For this comparison, according to our client, the most important and useful comparison is to the side by side comparison of one session to the next. Therefore, the GUI will focus on giving side by side comparison by showing video of each of the sessions and using some calculations for changes in body position while training. While the goal is to use the calculations to analyze changes over time, the client also wanted there for the coach to view the values gathered directly in a spreadsheet. Therefore, this option will also be available.

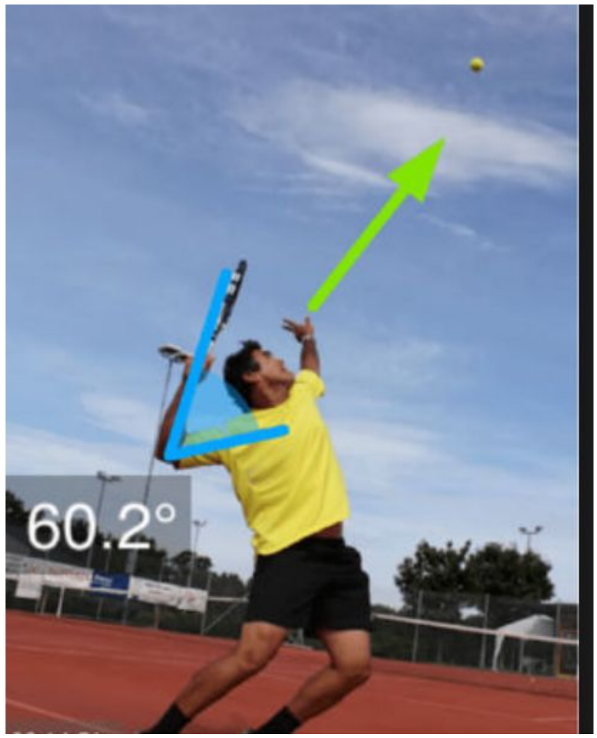

Figure 5. Dartfish GUI

2.3.4 Sensor Analysis

The sensor will be ergonomic for the user when active. This goes for both accelerometers and pressure sensors. Further, the sensors will both need to be wireless. In order for the athlete to be able to utilize the software, a distance of 50 yards will need to be achieved.

Both sensor types will need to be placed multiple times on the bike and athlete. Therefore, these need to be relatively inexpensive as multiples of each are needed. There are also environmental concerns that come with outdoor races that will need to be considered as well.

A sensor we are currently looking at is an Omega Force Sensor Resistor [3]. The ergonomic and price abilities can be seen below in the image. This is a functional requirement as there is a large network of sensors that will input the data for the final computations.

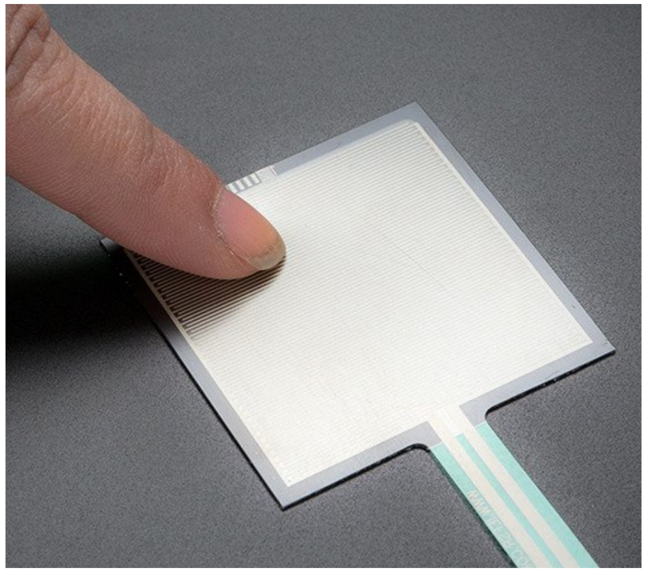

Figure 6. Omega Force Sensor Resistor

2.3.5 Transmission

Many athletes have phones on or near them when exercising. Therefore we are able to use the phone as a transmitter. The sensor has a few different options of transmission: Wi-Fi or Bluetooth. The phone receives the data and then exports it to a device that will compute the analysis for the data. This step is a functional requirement as in order to process the data it must be transmitted to the device doing the computing.

We have not begun the programming of this application at this time because of properly allocating our resources. We are working towards completing our Phase one software prior to working on our phase two software.

2.3.6 Sensor Processing Software

The data processing software will be in Python as well. This will allow keeping our files within the same library. However, this data is relatively easy to decode and turn into the final computed data since the input is analog values. Therefore the data will be imported to Python to compute the analytics.

We expect Phase Two algorithms to be easier than the phase one equations and more geared toward display.   There is little processing that needs to be done; therefore, this is a non-functional requirement since the data can be read manually from the analog values.

2.3.7 Sensor Graphic User Interface

The User Interface must again be focused on being user friendly with a simple backdoor for the client. The data displayed will come from Python and be displayed on a web app. The data will be presented in a similar way to that of the figure below.

Additionally, this is a non-functional component making this not one of our top concerns. In order for this to be of any use, our data and calculations must be correct.

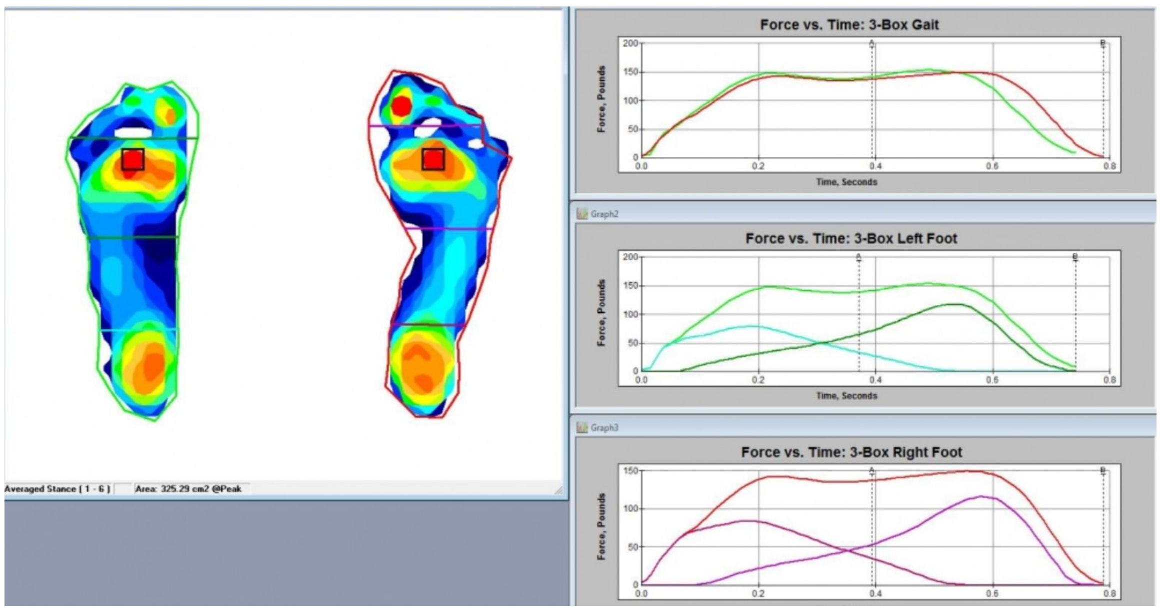

Figure 7. Sensor Data GUI

2.4 Design Analysis

Our design analysis for this document will mainly focus on the image processing system. Our sensor analysis has been minimally looked into as to allocate more time and resources for the image processing, as preferred by our client.

2.4.1 Camera Selection and Testing

The camera selection has been determined. It is currently being tested on different colored objects as well as different moving objects (linear, circular). The best colors for the cameras to track, in regards to quality of data output, is also being tested.

2.4.2 Software Selection

The software that currently being worked with are Python, HTML, JavaScript, and a few other useful GUI programs. The goal of the software is to ensure the data can be presented in a meaningful and efficient way. Python was chosen because there are many common libraries for data analytics and presentation. Additionally, Python is lightweight and easy to learn. HTML and JavaScript were chosen to help develop the web application because of their accessibility to tutorials and documentation. They also offer integration with Python code which is what will be used for the data analytics. Easy to operate GUI programs like ipi will be used to help extract data from the cameras to be used for further analysis.

2.4.3 Strengths and Weaknesses

The strength behind the project model is the combination of the two systems. This is something the market doesn't currently offer that the client will appreciate. The other strength is the error margin of 2.5% or less. The strengths on the current set up is the price. The cameras are able to be on the cheaper end as they did not come with a housing, yet still met all the specs.

The weaknesses of the project are the multiple sections. The system is made of a few different pieces which increases the opportunity for breaking. Also, multiple computers will need to be used, as each camera requires its own usb controller to interface with the image capturing software.

2.4.4 Threats and Opportunities

The emerging threat is the different systems that are already on the market. They have the components individually which requires many different systems.

The opportunities of this system are with additional extensions. An example for this would be an application for running.

3 Testing and Implementation

The areas to test are: camera precision, sensor precision, GUI friendliness, and calculation accuracy.

The camera will be tested by using a linear programmable actuator. This will give the exact distance, acceleration, and position. We can compare the distance displayed by the camera to that of the actual programmable actuator. This will determine the accuracy of the camera.

The next test is the sensors. We will use a programmable force inducer. This will provide a determined force. The force and the surface area will both be known and can be used to determine the pressure. This will give the accuracy of the sensor since the data is just being displayed and not final computation is necessary.

The GUI friendliness will be determined by asking our client. The client will determine if the GUI performs to a level of simplicity.

The final test run is the calculation accuracy. The protractor creates an angle exemplified by an apparatus that will be similar length to the proportions of a potential rider. This will allow for an exact measurement to be determined and carried out. If the GUI does not reflect this information an error occurred within the calculations.

Camera testing is currently being implemented. Before any other tests can be carried out, it must be determined that the camera is within allowable operating error. However, due to the nature of the use of the camera, it is anticipated that the results will be within the error as well as the frame per second, making this a decent subject. Since the testing is just beginning, there are not any affirmative results. If any issues arise, it will be expected to change cameras to something such as a GoPro or web camera.

The final results and needed changes will be documented as we progress in the project.

3.1 Interface Specifications

The majority of interfacing will be done through our web application. All the data received from the cameras and sensors will be sent to this application where the user will be able to view it. The user will be viewing the application from a desktop or a mobile device due to the fact that the user may not be stationary. This provides issues as the data needs to be readable from the desktop and and mobile perspective as well as all the applicable features. We will be using an application called Bootstrap which will help with testing that our application can be read in a mobile and desktop format [4]. It has different view functions that will allow us to see the application in both formats. This will mitigate any issues like data being cut off the screen or certain features not being able to be accessed. This is very important as the user will be analyzing the data themselves so the whatever is shown via the application must be readable and correct.

3.2 Hardware and Software

For are hardware, we plan to use a linear programmable actuator to test the camera precision. The actuator is able to determine the precise distance, acceleration, and position of the object it is tracking. We can compare the distance and position of an object tracked by the camera to that of the actuator to determine the accuracy and precision of the camera. We also plan to use a programmable force inducer to test the pressure sensors. The weight and the surface area of an object will be known and be used with the inducer to get the correct force. This force will then be compared to that of the sensor to test its precision and accuracy. Finally, a protractor will be used to measure the correctness of our program that computes the angles of the athletes body based on data retrieved from the camera. The protractor will be able to give us the exact angle of the position we are measuring and this can be compared to the output of our program to determine if our calculations were correct.

From the software side of things, we plan to use the common debugging tool found in many compilers. This will allow us to watch the code and even insert determined data for variables in essence of testing. This debugging tool will make it easier to find the errors and their source easily. An application called Bootstrap will also be used to test that all the data and features can be seen through the mobile view and desktop view of the web application.

3.3 Functional Testing

Unit testing in the Eclipse IDE will be used to verify the behavior of our angle analysis program. Two vectors and their predetermined angle will be given as input to the unit test. The test will then compare the output of the program based on the given input to verify that the calculations are correct. If the program output differs from the unit test, modifications to the program will be made.

Integration testing using Bootstrap will be used to verify that the web application is working in both the desktop and mobile format. We want to user to use the web application on both their computers and mobile devices. After completing the web app for the desktop site we will use Bootstrap to integrate into a mobile site and then verify that the integration has worked using the various features Bootstrap has like the view from desktop and mobile. Also, after unit testing our angle analysis program we want to put it onto our web app. This integration of the program onto the web app will need to be verified through comparing program results before integration and after.

3.4 Non-Functional Testing

The non-functional testing will be anything beyond the computing where analog values can be read. This would mainly stem from the GUI aspect as well as the 2.5% accuracy, and the environment circumstances of the sensors.

These will be achieved through a trial run. The environment test will place the sensors on a bike and have test run through less than ideal conditions. Positive results would require that the system, start to finish would work the same.

The GUI will be rather obvious whether it is correctly working. The only aspect that is ambiguous is the user friendliness. This can be tested by asking biking non-subject matter experts opinions on the GUI and their ability to figure the data out.

The final part is the 2.5% accuracy. This is not a functional requirement since the data is still collected and computed.

3.5 Process

As previously indicated, the tests have been outlined in previous Section 3 and Section 2. We understand our client, not only what they are looking for but what they need. Thus, our testing is just beginning on a very sound and well thought out platform which is highlighted in Section 3.

3.6 Results

In order to confidently stand behind the data, further results will need to be run. Different scenarios such as different colors, different backdrops, and even different lighting are looking to be run in the future.

There have been a few things that are able to be confidently reported on. The color orange is easily picked up from surrounding areas. There are some bright colors that we want to stray away from such as bright yellow. This tends to be similar to skin tone at times as well as potential backdrops.

The other standard we will have to draw for our client is clothing. Many athletes wear black shorts, however, their jerseys tend to be exciting colors for road safety. Determining the bright color sticker is significantly harder on a bright color jersey. We will also need to require the subjects to wear something form fitting to their torso and legs to not skew with the data. Their shoes will need to be uncovered as well.

Figure 8. Typical Athlete Jersey

Figure 9. Typical Fit of Bike Attire

We are in daily contact with the ETG and Computer Science department to acquire enough cameras to get the iPi to work. Once we have the full resources we will begin further testing.

3.7 Modeling and Simulation

We are using a library in Python called Matplotlib to model the athletes body and movements via plots. After receiving the data points from the cameras we can insert them into our Matplotlib program which then plots the points in a 3D plane. The plot simulates the athlete's motion being recorded by the cameras. Using this plot can we calculate the angles of the athletes body and display them back to the user via the web application. An example of this can be seen in Figure 10 below.

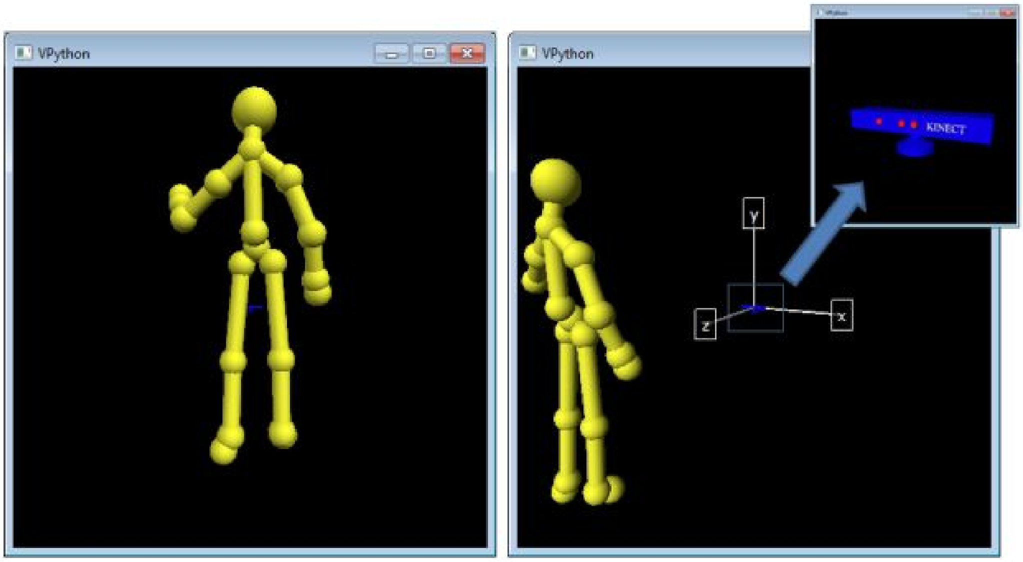

Figure 10: Model of Athlete Motion

3.8 Implementation Challenges and Issues

Overall the group did run into some issues and challenges during the implementation stages. One issue we ran into was the color of the tracking sticker. Since riders usually wear a bright colored outfit for road safety , tracking them using a bright colored sticker was significantly harder since the sticker could not easily be distinguished from their outfit. The second challenge we ran into was the quality of the cameras. The first batch of cameras we used recorded at 480p 30fps. After multiple test with the cameras, it was clear some of the athlete's movements were not being properly recorded resulting in a loss of data. Better cameras were needed to capture every change in the athlete's motion.

4 Closing Material

4.1 Conclusion

We have currently researched our phase one, image capturing beginning to end. This allows us to ask our client to only purchase the items needed once. Thorough investigation will create a path for the entire project to follow and mitigate any potential useless purchases.

Our goal is to create an all-encompassing bike position analytic system. This will incorporate image capturing for in lab environments as well as sensors for on road training.

Our plan of action is to finalize the camera selection and testing. Then finding additionally Kinects and incorporating them into iPi. This will allow us to begin the final stages of coding for our phase one platform. Then we being our debugging sprint where we find the errors to keep within the required accuracy.

4.2 References

[1] ipioft.com. 2008. About ipi Software. Retrieved November 11 2018 from http://ipisoft.com/store/ipi-biomech

[2] HHS Office of the Secretary,Office for Civil Rights, and OCR. 2013. Summary of the HIPAA Privacy Rule. Retrieved November 29 2018 from https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/index.html

[3] omega.com. 2018. Pressure. Retrieved November 29 2018 from https://www.omega.com/pressure/

[4] Mark Otto. 2018. Bootstrap. Retrieved November 29 2018 from https://getbootstrap.com/

Figure 1. Project Block Diagram

"sdmay10-01Block Diagram." *Matlab - Documents*, Ryan Hansen.

Figure 2. 120 fps Camera

"120 fps Camera." *Amazon.com*

Figure 3. iPi Software

"IPI - Member Network." *IPI - Member Network*, www.ipiphoto.com/.

Figure 4. Important Body Points to Track

"Body Position Matters - Bikers' Guide." *Biker's Guide - Online*.

Figure 5. Dartfish GUI

"Dartfish User Profile." Dartfish. General Membership, www.dartfish.com/.

Figure 6. Omega Force Sensor Resistor

"Force Sensor." Omega Catalogs and Specs. Omega Online, omega.com.

Figure 7. Sensor Data GUI

"Pressure Sensor Biometrics." Google Image Search.

Figure 8. Typical Athlete Jersey

"Outfitted - Biker's Guide." *Biker's Guide - Online.*

Figure 9. Typical Fit of Bike Attire

"Outfitted - Biker's Guide." *Biker's Guide -Online.*

4.3 Appendices

Our Microsoft Kinects are owned by Iowa State University.