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 (This should be the similar to the project plan)

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 we work closely with Nathan. He provides us great examples and a clear direction for our project.

1.2 Problem Statement

Elite cycling athletes depend strongly on a good fitting bike. The bike fit encompasses the saddle height, three separate angles of the saddle, handlebar stem length, and handlebar angle. Each variable translates into athlete body position. The fitting is currently done by eye with no angle analysis and only exists on a trainer.

The project will encompass two different systems: a video system and hardware system of sensors. These two systems will operate both in a lab and during real life 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 screen but not analyzed. Each athlete is different depending on their body. Thus, the data will be presented as calculated and not compared to a preferred ratio. Instead, the coach and athlete will make the next steps to better the performance independently.

The future outlook for this project will be to use this system for running as well. This expansion into running will also allow for the expansion of the company. The requirements will translate over and only require a few software implementations.

1.3 Operating Environment

The two systems will be held to two different locations while maintaining the common underlying fact that it cannot interfere with the athlete. The video analysis is to be done indoors. The stickers for the tracking must be easily visible for the camera and ergonomical for the athlete.

The hardware system is comprised of accelerometers and pressure sensors. The hardware system will be used outside during races and training. The locations for the sensors will be dynamic, needing them to be wireless. In addition, the sensors will need to be waterproof and durable for multiple hour races or training sessions in the rain over roads.

1.4 Intended Users and Intended Uses

The end consumer will be the athlete. The athlete 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 data that they can read and understand. Therefore, collecting the data must not interfere with the athlete. Then on the final product, the data must be presented in a way that is easy for both the athlete and coach to read. 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 looked at in an outside format. The client would like the data reported for the athlete and coach but also accessible to be pulled out for further analysis.

1.5 Assumptions and Limitations

The first assumption is that the sensors and camera 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 put on multiple different athletes and not have the analysis skewed.

The limitations vary. The first limitation comes from the rider comfort. The stickers for the camera and the sensors for the system need to not interfere with the athlete. The other limitation needs to be that the sensors are wireless. Sensors when not fitted properly will affect how the rider is positioned on bike making the data not representative of when the sensors or stickers are not on the rider. The next limitation is price on the camera. While the price of 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 used for these camera is quite large. The space specified for the system we are producing is not as large as these cameras require.

1.6 Expected End Product and Other 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 the two needs. The first system is due by the end of the first semester. The system encompasses everything from choosing the camera, collecting the data, analyzing it, and presenting the data in a user friendly manner. While keeping this on track, testing needs to be implemented along the way, different software and sources need to be used to get the data into a format that will be usable, and then presented into an interface for the coach and athlete. The deliverable is working system that includes a camera and a web app for the consumers to use.

The second portion of our project was listed to begin after the first part is complete. The hardware system is currently scheduled to be completed by the end of the second semester. The 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 transmitted from the wireless sensors.

2. Specifications and Analysis

2.1 Proposed Design

The proposed design follows the block diagram outlined below.

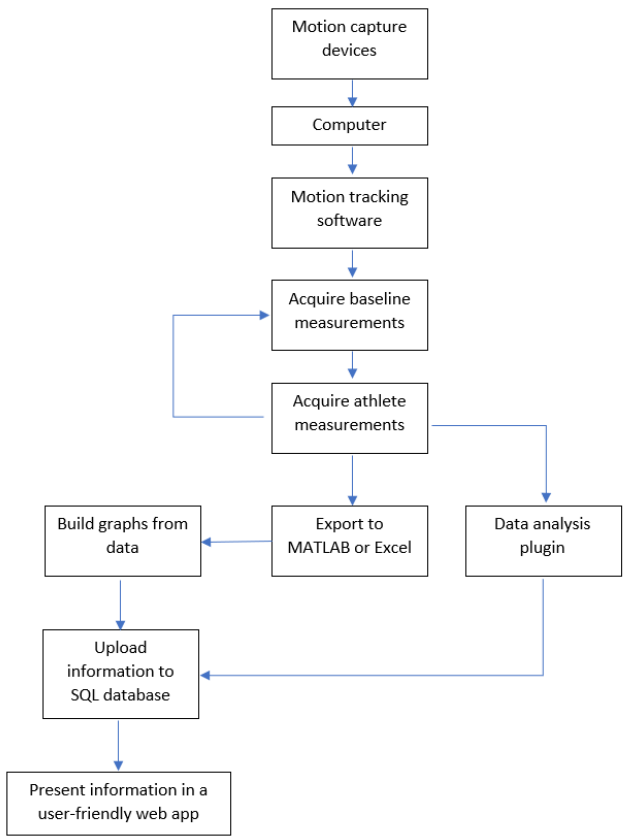

Figure 1. Project Block Diagram

2.1.1 *Image Capturing*

2.1.1.1 Camera

The two largest limitations of the camera are the budget and the frames per second. The final apparatus will have a minimum of 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 as the rpm of an athlete can be rather high especially when on hilly terrain. Further, the rpm alone cannot be the frames per second. In order to follow the path of the foot along with other movements affected by pedaling, multiple points are needed within one revolution to make a complete circle. The final obligation of the camera is that it must detect both color and depth.

The camera being used in a Microsoft Kinect. These cameras offer both distance and color data. This will be pivotal when calculating the positioning of the athlete as they fatigue. In addition, we are able to use three different cameras from the price of one expensive camera. Additionally, these cameras are also easy to find and are multifaceted as well as easy to manipulate.

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 that it comes with. 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. Microsoft Kinect

2.1.1.2 Image Processing Software

The next step in the processing the color and distance data that we receive, is via an image processing software. The difficulty with determining what software we will use is that typically the software and cameras come together. Therefore, we needed to find software that would allow a different camera. The other limitation in regards to software is the price. Many software applications require subscriptions and renewals. This was something we wanted to avoid for our client making it as user friendly as possible.

We came to the conclusion of using iPi. This software can take the image from a camera and turn it into multiple formats. The software can turn the image and corresponding data into different types of computable data. Additionally, it allows for easy transmitting of data between different sources, which is what we need in order to analyze.

We are currently waiting to acquire enough Microsoft Kinects for iPi to work. We are working to find them within the University so our client is not required to purchase anything, either hardware or software, prior to knowing it works cohesively. This point is a functional requirement. In order for the data to be transported, this software needs to interface with the camera as well as Python flawlessly.

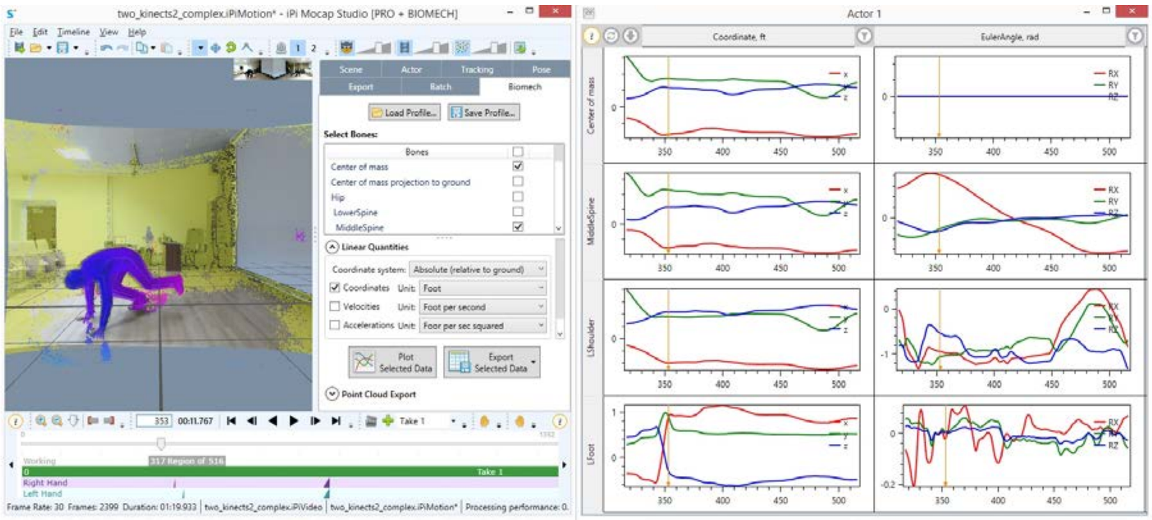

Figure 3. iPi Software

2.1.1.3 Analytics and Computation Software

In order to avoid the need for gates for development for our project, we have concluded that the data to be computed needs to be easily transferable. This means that we can expect the data to be delivered in a specific format into the geometric body analysis. This means that we could begin our computational software for the positioning sooner instead of needing to wait for it to be completed.

The software and programs we are in the middle or creating are beginning to piece together the data points and the math. The variables and equations behind calculating the angles are being scripted.

Thus, we have many different options for a software to use. Our team chose Python for a multiple reasons. We have some experience in this language. However, the library capability that the Python offer are largely available 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 we plan to use. Below is a picture that gives lighter meaning to what we are trying to track. This is a functional requirement. For the data to be within 2.5% of correct, the equations and data need to be correct. Therefore, it is imperative that the equations are correct. If they are not correct, the coach will see incorrect issue and try to resolve them, potentially making the issues worse.

Figure 4. Important Body Points to Track

2.1.1.4 User Interface

The end goal of our project is to emulate a web app that displays the data both live and afterward. The need for a web app is rather important to be able to view from a distance as well as live. Therefore, our group is using html especially with the ability to interface with Python. This will allow us to display that data that our previous system has created.

A software that is common to programs similar to our client is featured below. This is something that our client is looking to add to their feature and would like us to emulate. Therefore the picture below will resemble our final User Interface. The final interface will need to be both informative but basic. A coach will do the analysis based on the data we present. Therefore our assumption is the coach is a SME in biking and not web apps which is why this needs to be basic. Further the app needs to allow our client in to the backside to view the data if he so desires.

The developments we have made to our GUI is the layout. We have determined what data we are interested in presenting and the level of simplicity we plan to achieve. This is a non-functional requirement. The computed data is already determined in Python. Therefore, this GUI isn't required and could be replaced by reading the data manually.

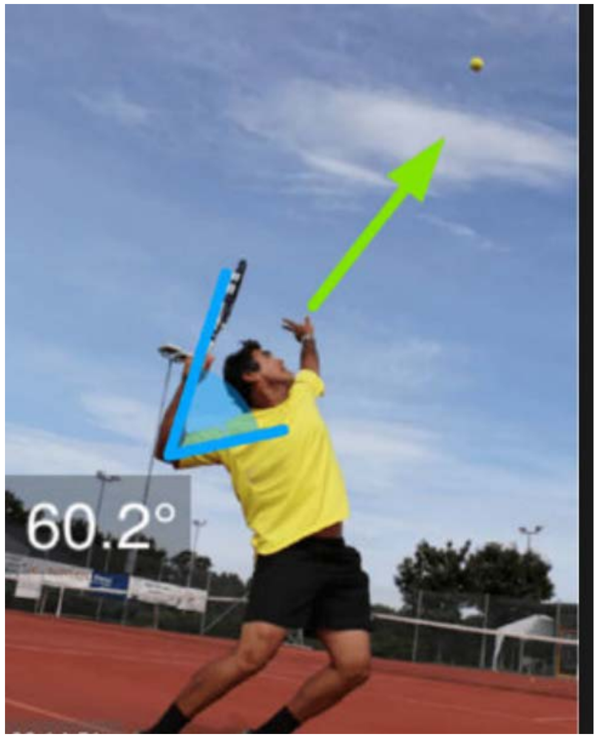

Figure 5. Dartfish GUI

2.1.2 Sensor Analysis

2.1.2.1 Sensor Selection

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 Omgea Force Sensor Resistor. The ergonomic and price abilities can be seen below in the image. This is a functional requirement as it is a large network of sensors that will input the only data for the final computations.

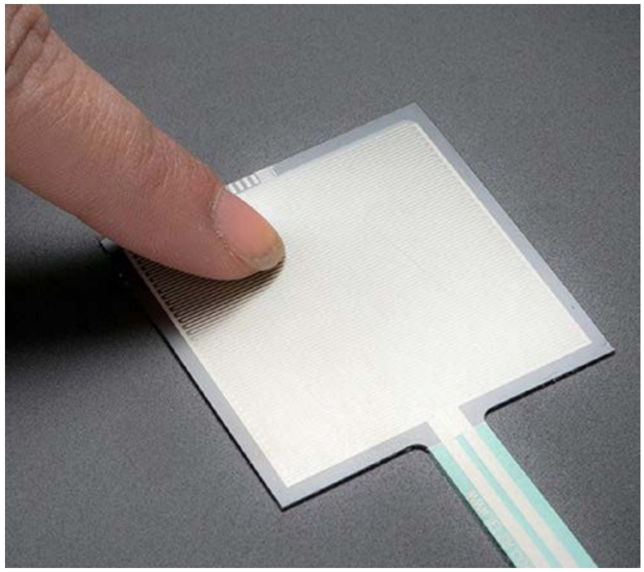

Figure 6. Omega Force Sensor Resistor

2.1.2.2 Transmission

Many athletes have phones on or near them when they are exercising. Therefore we are able to use the phone as a device that the sensors will transmit towards. 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 1 software prior to working on our phase two software.

2.1.2.3 Data Processing Software

The data processing software will be in Python as well. This will allow us to keep 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 we will be importing the data to Python and computing the analytics.

Here as well, we have not begun our algorithms for this again either due to resource allocation. However, we expect this 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.

2.1.2.4 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.

Little development has been made here. This will be added to our other GUI, making this an add-on after that step has been achieved. 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.

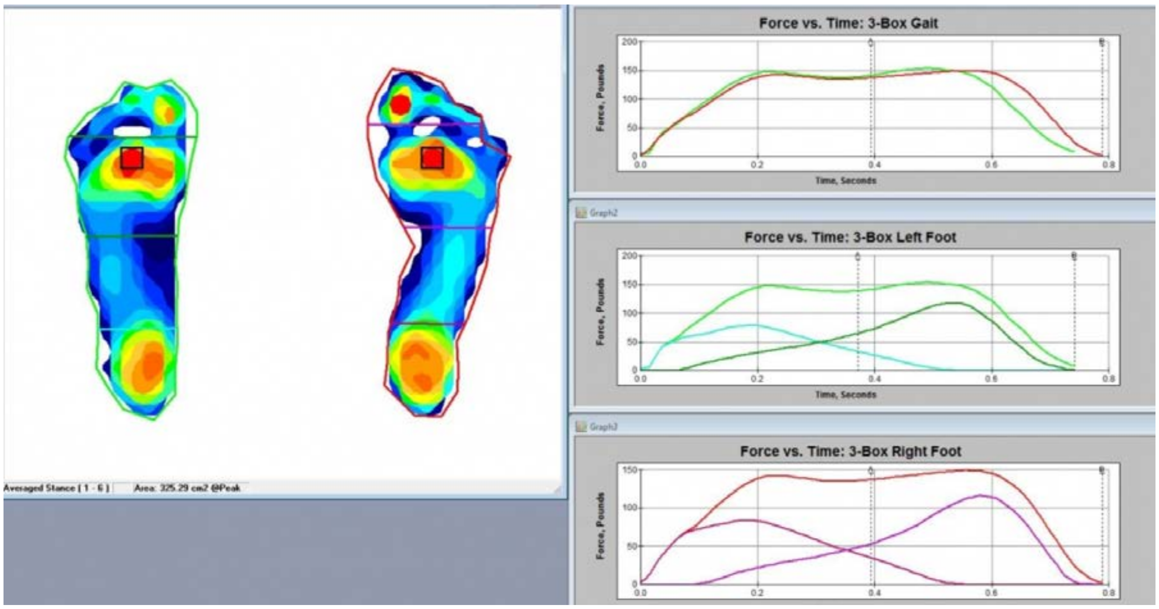

Figure 7. Sensor Data GUI

2.2 Design Analysis

Our design analysis for this document will encompass only the image processing. Our sensor analysis has been minimally looked into in order to allocate the time and resources for the image processing, as preferred by our client.

2.2.1.1 Camera Selection and Testing

The camera selection has been determined. We have chosen the Kinect (for further reasons see previous section). We are currently testing it on the different colored as well as moving objects. Something we plan to look into is best colors for the points to be tracked. When near skin tone we are expecting a few things to change, which is a further test we plan to try on with the Kinect.

2.2.1.2 Software Stuff

The software that we are currently working with is both html, Python, C++, and a few other useful GUI programs. This has allowed for us to determine different ways to parse the data as well as present it. Something that worked well for us is Python. It is both versatile and easy to learn. It has some great image libraries that we plan to take advantage of. Something that did not work well was C++. This is cumbersome and the visual capabilities were not of what we found Python to be.

3 Testing and Implementation

The needed types of tests are as follows: camera accuracy, sensor accuracy, GUI friendliness, and calculation accuracy.

The camera accuracy will be tested by using a linear programmable actuator. This will give us the exact distance acceleration, and position that we are looking for. 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 portion is the sensor accuracy. 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 us 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 is expected to be easily interpreted by his clients.

The final test run is the calculation accuracy. We will use a protractor to create 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.

We are currently working on the camera test. 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 Kinect we are anticipating that the results will be within the error as well as FPM, making this a decent subject. Since we are just beginning to test, and have not created enough samples, we do not have any affirmative results. If any issues arise, we expect 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

We are currently testing our camera and the visual display software we are using. This will allow us to determine the feasibility of these camera. After determining that it will work, we will need three different Kinects to use iPi. Once this happens we can begin to test the fully integrated hardware to software leap in phase one.

After these tests are complete, further tests will be created and completed. Additionally, we may determine a finite number of tests. However, as we proceed more tests may need to be added if we find things that are unexpected or results that are contradictory.

3.2 Hardware and software

We plan to incorporate all of our hardware and software previously listed in this document for testing. At this point in time, we have designed the tests outlined in Section 3. Further many aspects that our project will allow us to incorporate is the answer being obvious. Either the transmitting app is working or isn't, or the GUI is working correctly. The hardware may be more difficult to debug as we progress in the second phase.

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 our own data for variables in essence of testing. This debugging tool will make it easier to find the errors and their source easily.

3.3 Functional Testing

The functional tests will mainly stem from the integration of the hardware. Then the second portion will be derived from the use accuracy of the equations. This will be things such as the mentioned protractor example and the applied force on the pressure sensor example. These will clearly depict whether the full hardware and function requirement is integrated properly via testing.

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-SME's opinions on the GUI and their ability to figure it out.

The final part is the 2.5% accuracy. This is not a functional requirement since the data is still collected and computed. The accuracy now has the ability to be refined through equations to narrow in on the 2.5%.

3.5 Process

As previously indicated, the tests have been outlined in previous Section 3 portions as well as Section 2. By combining this without current status and research we find that our group is in a great spot. 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 say that we stand behind our results, we need to run further trials. We are looking to run different scenarios such as different colors, different backdrops, and even different lighting. To be concise about our findings and elaborate on for our client we will need further examples and test cases.

We have found a few things that we are able to confidently report 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 ComSci department to acquire enough Kinects to get the iPi to work. Once we have the full resources we will begin further testing.

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

Figure 1. Project Block Diagram

"SDMay10-01Block Diagram." *Matlab - Documents*, Ryan Hansen.

Figure 2. Microsoft Kinect

"Microsoft Kinect - Gaming Breakthrough." *Microsoft - Product*, www.microsoft.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.