Foot Motion Capture for Firefighter

A Design Project Report

Presented to the School of Electrical and Computer Engineering of Cornell University in Partial Fulfillment of the Requirements for the Degree of Master of Engineering, Electrical and Computer Engineering

Submitted by Di Huang, Liwei Han, Yining Gong

MEng Field Adivisor: Joseph Skovira

MEng Outside Advisor: Huiju Park

Degree Date: January 2017

Abstract

Master of Engineering Program

School of Electrical and Computer Engineering Cornell University

Design Project Report

Project Title: Foot Motion Capture for Firefighter

Author: Di Huang, Liwei Han, Yining Gong

Abstract: At the present time, there is only anecdotal evidence concerning possible design issues with Firefighter boots. This motion capture system is used to define the possible problems when Firefighter boots are being used. The system has three functions:

- data detection
- data collection
- data display.

SV03A rotary sensors are being used in this system to detect the angle change data of ankle of the foot. Its biggest advantage is that it fits into the small room of Firefighter boots. Data collection part uses Arduino Uno microcontroller to realize its function. Once the angle change has been detected by the sensor, microcontroller will filter the data and transfer it from voltage to angle data and finally store it locally on a SD card by using the SD card shield. Last function for this system is to display data in animation form. After finishing collecting data in an experiment, data will be uploaded to a Processing program and display the data out in an animation form.

Individual work

Di Huang:

-- Took active part in the sensor selection progress.

Proposed the shortcomings of application of accelerometer and optical system. Sparked two other ways to realize this project: angular sensor and RFID approach.

- -- Found an ideal sensor fitting in our specific design requirement for size.
- -- Fulfilled the linearity verification test as mentioned in the implementation part with Liwei.
- -- Came up with the primary idea about how to place the sensor on foot and the set-up for the sensor part.
- -- Built up the first version of the sensor component (the one with wooden stick) as described in the design part for sensor with Liwei.
- -- Drew the stereomodel in AutoCAD for 3D printer with the help of my dad to build up the second version of the sensor component (the plastic one printed with 3D printer) as described in the design part.
- -- Fulfilled the programming work with the software Processing, which is proposed by Liwei, to visualize the angle data produced by Arduino.
- -- Took part in implementing all the experiments as mentioned in the implementation part.
- -- Did the documentation about discussions and meetings, as well as experiments.

Liwei Han:

- -- Help select the appropriate sensor for this project.
- -- Test the SV03A rotary sensor.
- -- Assemble the SD card shield with microcontroller.
- -- Build circuit to connect sensor with microcontroller.
- -- Program the microcontroller to collect data.
- -- Wrote average filter function in microcontroller to smooth the data.
- -- Take active part in every experiment

Yining Gong:

- -- Search for the possible solutions to the problem. Raised the solution of RFID and talked with teammates and then discard this plan.
- -- Take part in the implementation of the system. Helped in finding materials which is suitable for fixing the system to body and building circuit.
- -- Test the system and collect the data of the tests. And documented the final report.

Executive Summary

The goal of this motion capture project is to measure and collect the data relative to discomfort caused by the nature of the protective boots for firefighters.

In order to measure the movement of the feet, we designed a system incorporated with sensor and microcontroller. Our project mainly consists four parts:

- sensor selection,
- sensor displacement
- data collection and data display

For the sensor selection, after comparing with lots of different sensors, including pressure sensors and accelerometers, we finally choose the SV03A rotatory sensor which is produced by Murata. The biggest advantage of this specific sensor is its size. The thin type is 2.1mm. The length and width are 12mm and 11mm. So the size of this sensor satisfy our design.

Second, in order to make the measurement accurate, we carefully place our sensor and made it fixed on the foot. The fist thought is just to use wooden stick to connect two test points, but the drawbacks for this version is obvious that it is too fragile and wooden stick is too hard and can not be bent to fit the curve surface of ankle. With the help of professor Skovira, we finally design a connection rod which is made by a 3D printer to effectively connect two test points. It is a plastic stick with one little protrude on one side to fit into the hole in the center of sensor and to make the sensor rotate with the motion of the stick.

Then we choose Arduino Uno microcontroller to be our data collecting device under the consideration of memory size, power supply, number of I/O pins and transfer speed. Also, we programmed the microcontroller to transfer voltage data to angle data with average filter function to make the data more accurate and store them locally.

The final part is displaying the data. We used a software called processing which is based on Java and a good APIs for us to learn. We use it to process the data we collect and animate the foot motion to give a direct visual feel. In details, we use two lines to sketch our leg and foot. Once the data is uploaded to the program, the line for the foot will rotate accordingly with the data.

After comparing the results of two test conditions: test with bare foot and test with wearing boots, we can easily see the difference of these two data: the data of bare foot conditions has bigger range than the other, also it has a faster data changing speed.

Contents

1) Motivation
2) System Requirements
2.1 Overview
2.2 System Requirements1
3) Project Issues
3.1 Sensor Selection2
3.2 Sensor Displacement
3.3 Data Collection
3.4 Data Display4
3.5 Issue Summary4
4) Range of solutions4
5) Design and Implementation
5.1 Design6
5.2 Implementation13
6) Comparison with original design21
7) Final result
8) Challenges23
9) Conclusion23
10) Future Work
11) Acknowledgement
References
Appendix A: Code for Arduino26
Appendix B: Code for processing
Appendix C: Sample of raw data
Appendix D: modeling for 3D printer

1. Motivation

Firefighters need to work in a tough environment. Their protective gear therefore, needs to be rugged and impervious to harsh environmental conditions. However, the bulky nature of the protective boots that are required gear makes firefighters uncomfortable and they often complain about this. Cornell's department of Fiber Science & Apparel Design is working with fire fighters to improve the fit and function of protective gear.

In order to define the possible problems, we propose a system to measure the foot motion for firefighters when protective gear is being used. By collecting the data of several joints of feet, we want to show the sources that cause the discomfort when firefighters are wearing the boots to help the Cornell's department of Fiber Science & Apparel Design to begin to determine the nature of the issue and develop potential solutions.

2. System Requirements

2.1 Overview

This project which is associated with the Cornell's department of Fiber Science & Apparel Design is a totally new project. Therefore, we have limited source for us to look up. During the process of doing this project, we have discussing lots of details about this project with Professor Park and Professor Skovira to change the system requirements which can be realized within one year with our best efforts. The issues we faced can mainly be divided into three parts: sensor, microcontroller and data display. The details of these system requirements and issues are as follows:

2.2 System Requirements

1) The system should use motion capture techniques, including rotary sensors to measure motion parameters.

2) Measurements should be stored locally and the data should be displayed in an animation form.

3) Data should be as accurate as possible and must show the correct trend as time increasing.

4) The system must be light enough to be included with the boots and have the ability to log the sensors every 100 milliseconds.

5) The component of sensing of foot motion should be small and thin enough to be placed on

target landmarks on the foot inside the boot.

6) The boot sensor system must be designed to record motion appropriate for capturing at least one significant movement.

7) The sensor system must also be able to survive daily deployment.

8) Rechargeable battery life should be at least 24 hours.

9) On-board memory must be large enough to record telemetry for each event.

10) The system must be capable of data download with a minimum requirement of manual download.

3. Project Issues

3.1 Sensor Selection

Our first problem is to find an appropriate sensor which should be small and precise enough to fit into firefighters' boots and detect angle change of several joints of our feet when we wear boots and walk.

Professor Park from the Department of Fiber Science & Apparel Design told us that they already have the device that can be used to detect the angle change when we are walking. However, their device is can only put outside the firefighters' boots and it is too big to be put into the boots. They think that this measurement is very inaccurate, since it can only detect the boots movement not the truly feet movement. After borrowing the firefighter boots from the Cornell's department of Fiber Science & Apparel Design, we found that when we wear the boots, the remain space is really small, so we need to carefully choose the sensors that can fit into the boots. Also, this sensor should be accurate enough to help us collect the precise data.

3.2 Sensor Displacement

When we find the appropriate sensors, next issue is that how and where we should put these sensors on our feet. This issue is nontrivial, because if we want to capture the motion, we need to build reference axis first, and how to build the reference axis depend on how we put these sensors.

Also, if we cannot make the sensor fixed on foot, the data we collect will be very inaccurate. As

we all know, the surface of our feet is not always smooth and the sensors may can not be easily fixed on it. Therefore, we need to figure out how to stabilize these sensors on foot.

Another issue about the sensor displacement is the connection device we use. As you can imagine, to detect the angle change of our joint, we need to fixed our sensor on one point and using a connection device to connect another point on foot that will make the angle change when we are walking. However, this device need to be made specifically due to the sensor we select to actually connect these two points.

At last, all the device we use should not be easily broken while we are doing experiments. Therefore, we need to consider what materials we should use for the connection device and how to connect them stably and safely.

3.3 Data Collection

After dealing with the issues above, our next problem is how to collect the data and transfer them to the server. As mentioned in the system requirements, we need to choose a suitable microcontroller which can meet all our system requirements for this project. Things we need to consider:

- when we select the microcontroller is how much memory do we need?
- what is the range of power supply of this microcontroller?
- does it transfer speed fast enough?
- how many I/O pins does it have?

For this project, our system does not require too much memory on board. But considering the future development on our system, we should not only use limited on- board memory to store data locally. Instead, we need to consider using extra device to enlarge our data store space on board.

As for the power issue, on the one hand, we need to minimize the power we use to let our system stay longer. On the other hand, because our device should be fixed on the person who wears it as he moves, the battery case should not be too heavy.

Next, the microcontroller we use should have fast enough transfer speed to meet the system requirement that it should have the ability to log the sensors every 100 milliseconds.

Another factor we need to take into consideration is that the number of I/O pins it has. In more details, how many analog I/O pins and how many digital I/O pins on board. Although right now, we only detect one joint on foot and only use one of them. But again, for the future development on our device, we need to set aside enough I/O pins.

Finally, because the data from the sensors are voltage or current signals and what we really want is the change of angle, so we need to convert the voltage or current signals to the actual parameters that we want to measure.

3.4 Data Display

As required by Professor Park, we need to display our data in animation form to let researchers have a direct visualization feeling about the foot motion. This asks us to find an animation software and learn how to use it for this project.

3.5 Issues Summary

- What is the best suitable sensor for this project?
 - small size
 - excellent accuracy
- Where and how can we display this sensor on foot?
 - stabilize the sensor on foot
 - how to design connection rod
- How can we collect and store the data?
 - microcontroller's memory size, power supply, transfer speed and number of I/O pins.
 - use voltage data to represent angle change.
- What tools should we use to display data in animation form?
 - select software
 - learn how to program with it.

4. Range of solutions

To achieve the mission of foot motion capture, several executable plans are proposed. To fulfill this target, the first and main choice we have to make is the selection of sensors, since different kinds of systems could be built up with different types of sensors, which will lead to totally different solutions. Because the space in the boot is not very large so the key issue of the selection of sensor is that this selected sensor must be small and thin enough to be put on the foot, which is in the boot. Another thing we must take into consideration is that this sensor must not affect the usual motion of the foot so that the data we collect is meaningful.

The first plan is to use accelerometer. There are several plans talking about motion capture using accelerometer available. However, we will need to take second order integration here to get the moving distance of foot from its acceleration from the data collected from the accelerometer. Because of accuracy issue, errors would be accumulated in the meantime, which could result in tremendous errors in the end, which leads the whole system to fail. Besides, this approach requires to know the orientation of the sensor with high accuracy to distinguish the gravity measurements and the physical acceleration of the sensor. Small error of orientation estimation would produce high error of measured acceleration, which could be accumulated into large errors in velocity and distance estimation as mentioned above. Thus, this plan is not available to us because we can't get accurate data from the sensor and inaccurate data would be a disaster to the whole system in this plan.

Another approach proposed is to use inertial measurement units (IMUs), which contains a combination of gyroscope, magnetometer, and accelerometer. there is available system using IMU for motion capture but the IMUs are too large to fit in the boots so we discard this plan either.

The third approach is to use optical system. This is the most popular technique that is widely used in already existing motion capture systems these days. Some infra-red emitting markers or reflective dots are worn by tester. Camera is used here to capture the movement of these dots so that we can get the movement of body by analyzing the position changes of these optical dots. However, this plan is not suitable enough for our situation as well because the boot would block the propagation for light rays.

Besides, there are also several other methods available. For example, we can use RFID to capture the movement. The obvious advantage of RFID tag is its size. The inclination of tags could be estimated by analyzing the polarization angle of the RFID tag response signals. Specially, the tag antennas have a single linear polarization, hence the direction of electronic filed back scatted from the tag antenna follows the longitude degrees from end to end of antenna conductor. But the main disadvantage of this approach is obvious as well. The working distance between RFID tags and tag antenna is usually about 10 centimeters, which can't satisfy the design requirement of the whole motion capture system.

Our final choice is to use angular sensor directly. One-unit design comprises two plastic bars that are placed on the body segments, and a flexible angle sensor element. When body moves,

5

the bars aligned with bodily axis would be taken to move as well, resulting in the angular change of the angle sensor. By getting the relative angles of the movement, we can then build the animation of the movement of the foot based on that. Actually this is the most direct way to get the motion of foot and the main problem for this system is that how to attach the plastic bars to the foot so that the movement of foot can result in the change of the relative angles between these two plastic bars accurately.



Figure 1 – The intuition of the whole set-up

5. Design and Implementation

5.1 Design

Design Overview

After the consideration and comparison about possible solutions mentioned above, our final designed work flow is as follows:

First, we choose angular sensor to capture the angle of ankle while walking. One-unit design comprises two plastic bars that are placed on the body segments, and a flexible angle sensor element. When body moves, the bars aligned with bodily axis would be taken to move as well, resulting in the angular change of angular sensor, which is a kind of angular sensor. Actually this

is the most direct way to get the motion of foot. The biggest challenge is to align sensors with bodily axis and to place them constantly.

Once we get the signal for sensors, we use microcontroller to collect these data, store them locally, and finally upload them to PC. The key issue here is the allocation of memory and the way to upload. We achieve the communication between PC and microcontroller by SD card.

Once the data is transferred to PC, the last step is to display them. Data are turned into animation to give people a direct point of view of foot motion. The chosen programming software is Processing. Processing is a software used for visualization by coding in Java. At the end, the final product is a visual interface to display the motion of foot in the boot.

To achieve the above work flow, the whole design is divided into three parts: sensor part, microcontroller part, data display.

The sensor part takes the responsibility of measuring angle change of foot motion. Then the microcontroller part takes charge of collecting and storing data. Finally, the data display part will show the data in the form of a foot motion animation.

Sensor

The first step for sensor design is to decide what kind of sensor to be used for our situation. After looking up to large amount of documents and references, possible alternatives could be sorted into three categories:

- mechanical approach
- optical approach
- magnetic approach

However, shortcomings of application of accelerometer and optical system, which are widely used in many mature motion capture products, are very obvious and couldn't fit our specific situation. After several effective talks with Prof. Skovira and Prof. Park, we sparked two other ways to realize this project: angular sensor and RFID approach. Since the working distance issue of RFID, finally we focused on the angular sensor way.

The next step is the specific type selection of angular sensor. The key issue here is that the angular sensor is supposed to be thin and small enough to be put on the foot in boot. After searching and reading the data sheet of some products, we found an ideal product fitting in our specific situation to some degree with the help of Prof. Skovira.

Our final choice is rotary position sensor SV03A produced by Murata. The biggest advantage of this specific sensor is its size. The thin type is 2.1mm. The length and width are 12mm and 11mm.

So the size of this sensor satisfy our design.



Figure 2 – The chosen angular sensor (from rotary position sensor datasheet, see Reference 3)

The basic idea of this sensor is when something inside the central hole rotates, it will drive the white part in the center of sensor rotating in the meantime. The degree change of rotated angle is proportional to the output voltage. Besides, the sensor is designed on the assumption that it is to be used with the output terminals directly connected to the A/D port of a microcontroller. So the only thing it needs as the external circuit is the input power voltage, which is 5V.

The next mission is to find a suitable way to fix the sensor on the foot.

After many experiments, the outcome of first version is to insert a small little plastic stick into the hole in the center of sensor. Since the size and shape of this stick fit in the hole very well, when rotating this stick, the sensor will rotate with the motion of the stick as well. In the center of the stick, a metal filament stretches out. Bend that metal filament over and glue it on the outside wooden stick. In that way, the outside wooden stick and little plastic part could be considered as a whole. When the whole component rotates, it will drive the sensor to rotate at the same time. Since the joints to put sensors are not flat at all, we use a flexible plastic piece to help us out. Drill two holes and glue the slice and other parts backside so that the sensor could be fixed on the plastic slice.



Figure 3 – The first version of sensor part

This is version could be considered as just an attempt since the drawbacks of this design are obvious. The whole system we built is fragile and there are too many components in this design. So it's always broken when bent a little bit. Besides, the wooden stick is too hard and can't be bent to fit the curve surface of ankle.

After discussion with Prof. Skovira, we decided to use 3D printer to print some solid figure specifically constructed as we wish. With 3D printer, we constructed the old wooden stick and metal filament as a whole so that there are less components than in the old design. There is also a bending on the stick to fit the curve surface of our ankle. What's more, since the material 3D printer uses is plastic, the new design is more ductile than the first version. In total, the new version is not as fragile as the old one.



Figure 4 – The second version of sensor part

Microcontroller

The functionality of microcontroller is to achieve the main processing work as a "brain". It reads in the data collected by angular sensor, which is discrete voltage, converts them to angles and stores these angles locally in a SD card in the format of a .txt file.

The selection of microcontroller should be based on the consideration of memory and transfer speed. Our final choice is Arduino Uno microcontroller associated with a SD card shield, since it can be easily programmed and meet all our demands for this project.



Figure 5 – Microcontroller and SD card

The input power supply for the Arduino we choose is 6V, which is powered by 4 batteries and each is 1.5V. The number of analog I/O pins it has is 6 which is enough for our project.

For the details about programming, it includes two parts. The first part is to let the Arduino read the data from the analog pin and use the serial monitor to observe the data. The sample rate we set before was 10ms. But when we stepped into the optimization stage, we changed the sample rate to 1ms and take the average of every 10 read data. This is for the purpose of "filtering" data and omitting the effect of some extreme data caused by noise. Even though the sample rate we set in program is 1ms, after taking the average, the "true" sample rate is still 10ms.



Figure 6 – data with "filtering" (left) and without "filtering" (right)

Figure 5 shows the comparation between data with/without "filtering". It's easy to tell that when we use the "filtering" mechanism, the derived curve is smoother.

Next, after comparing with the Ethernet shield and Bluetooth shield, we came up with the idea to use SD card shield to collect and store the data. We soldered the SD card with the Arduino and write a program to write data to the SD card. The program is to store the data read by the analog pin to the SD card and convert it to the angle type. This is easy to realize because the linear relationship between angles and output voltage determined by the sensor.

The detailed program could be found in the Appendix.

Visualization

Our goal is to use animation to show the movement of feet in boots. The Software we find to transfer the data into animation is Processing, which is a flexible software sketchbook and a language for learning how to code within the context of the visual arts.

The programming of software Processing is based on Java. The biggest advantage is that this software is embedded with many built packets and functions, which are very handy to call and implement.

The input for the program is a .txt file produced by Arduino. The contents of the .txt file are discrete angle data reflecting the movements captured by the angular sensor. The output is the animation of the movement for foot.

Specifically, we use two lines to sketch our leg and foot. The vertical line represents leg. It is fixed and gives us a reference for the movement of foot. The horizontal line represents foot. One end of the line is joint with the end of the vertical line, which is fixed. The other end moves accordingly

with respect to the input angles. Also in the final implementation we use two sets of such lines described above to represent the movement of bare foot and that of foot in boot. These two sets of lines move synchronously to give us a direct comparison of the difference between the two movements.



Figure 7 – The output animation produced by Processing

Figure 7 shows a screenshot of dynamic Processing output. The biggest difference between the movement of bare foot (the left figure) and the movement of foot in boot (the right figure) is that the first one moves faster than the latter, as well as the range of the first movement is larger than that of the latter.

The programming could be divided into three steps:

- find the path and read in the angle data in .txt file.
- construct two vectors and move them accordingly with respect to the read data.
- draw the lines as shown in Figure 5. The frame is drawn every 10ms which corresponds to the sample rate of Arduino. (The actual sample rate is 1ms. But because of the "filtering" mechanism, the "true" sample rate is filtered to 10ms)

The detailed program could be found in the Appendix.

The integrated system

Putting above discussion together, the whole set-up is as follows:

We put the sensor on where the joint is on our ankle. The sensor is fixed with medical bandage. Microcontroller is fixed on another part of our body and could be on the boot as well. The power supply consists four 1.5V batteries and can supply 6V voltage in total. The whole set-up is shown in Figure 8.



Figure 8 – The whole set-up

5.2 Implementation

Implementation Overview

During the process of implementing our design, the first step is to check the linearity of our sensor. Once the linearity is guaranteed, before trying to test our set-up with ankle, we conducted experiment on knee first because the area of knee is larger for us to place sensor, as well as the angle change of knee is bigger than that of ankle when we walking.

Once successfully implementing experiment on knee, several sets of experiments on ankle were conducted with bare foot, normal shoes and fire boot. Besides, some adjustments and improvements are made during the experiment process thanks to talks with Prof. Skovira and Prof. Park.

Sensor Verification

After receiving our sensors, the first step is to do some experiments to check the linearity of this sensor.



Figure 9 – Linearity check with drawn circle



Figure 10 – Rotating sensor to check linearity

Figure 9 and Figure 10 shows the implementation process of linearity verification. Generally speaking, the way we took is to connect the sensor with power and scope correctly, fix the sensor on paper and draw a circle around it. Each time we rotated the sensor so that the output voltage would be changed by 0.2V. Then we marked the angle change on the circle.



Figure 11 – linearity verification graph generated by Excel

The expected result should be that all the angles between two marks are equal, which is impossible. When we drew the angles according to the circle in Excel, which is shown by Figure 11, it's easy to tell that the output voltage is proportional to the rotated angle even though the relationship is not absolutely linear because of errors. So the linearity of the sensor satisfies our requirement.

Test on knee

As mentioned above, before implementation test on the ankle, we started testing our equipment on the knee.

The implementation on knee is easier compared with that on ankle. Firstly, the area of knee is larger for us to place the sensor and actuator. What's more, the angle change of knee is bigger than that of ankle when we are walking. So it is meaningless for us to conduct the experiment on ankle without obtaining expected results from experiments on the knee.



Figure 12 – Test on knee

We put the sensor on the knee. The green part as shown in Figure 12, together with the angular sensor will be fixed on the knee joint. One end of the red stick is fixed in the rotary part of the angular sensor. The other end is aligned with leg and moves together with leg when we are walking. The red part will drive the sensor to rotate at the same time so that the angle change of the knee could be collected. During the implementation, we put the sensor on our knee. Microcontroller is fixed on another part of our body. Then just walk as usual to collect the angle change change.



Figure 13 – The result of test on knee

Figure 13 shows the result of test on knee. It's easy to conclude that the movement of knee is periodic. And the range of the movement is almost 10 degrees.

First attempt on ankle

Once we got the expected result from the test on the knee, we began our test on the ankle.

The process is very similar with the process for the test on knee. We put the sensor on the ankle. The green part as shown in Figure 10, together with the angular sensor will be fixed on the ankle joint. One end of the red stick is fixed in the rotary part of the angular sensor. The other end is aligned with outstep and moves together with outstep when we are walking. The red part will drive the sensor to rotate at the same time so that the angle change of the ankle could be collected.

During the implementation, microcontroller is fixed on the leg. Then we walked as usual to collect the angle change of ankle.



Figure 14 – First attempt on ankle



Figure 15 – The result of first attempt on ankle

Figure 15 shows the result we got from the first attempt on the ankle test. This result is not ideal enough. There are periodic angle changes as expected, but the range of these angle changes is almost 8 degrees, which is supposed to be larger according to Prof. Park's experiences.

Second attempt on ankle

After discussion with Prof. Skovira, we decided to extend the length of our stick to narrow down the errors. So we printed a new longer black stick and conducted the process as mentioned above.



Figure 16 – The comparison between two sticks in the first and second attempts

The comparison between the old and the new sticks is shown in Figure 16. As we can see, the new stick (black) is much longer than the old stick (red).



Figure 17 – The result with shorter stick



Figure 18 – The result with longer stick

Figure 17 and Figure 18 show the result with shorter and longer stick. The range of angle is almost 8 degrees with shorter stick as shown in Figure 17. The range of angle is almost 12 degrees with longer stick as shown in Figure 18. The result with longer stick is better than the result with shorter stick.

Implementation on bare foot and foot in boot

Once our design passed the test on the ankle, we did experiment with the improved set-up. We conducted two sets of experiments at this stage. One is with bare foot, the other is with foot in boot. For each set of experiment, the process is the same as the one above.



Figure 19 – Experiments on bare foot and foot in boot



Figure 20 – The result of experiments on bare foot and foot in boot

Figure 19 shows the process during the experiments on bare foot and foot in boot. Figure 20 shows the result. From the contradict of these two pictures, we can see that when wearing the boots or not will bring big difference to the frequency of movement and the actions that the feet take. Details about the difference are discussed in Result part.

Implementation on bare foot, foot in required boots and running shoes

Our final experiment is asked by Prof. Park. Because he wants to know the movement difference among bare foot, foot in required boots and running shoes, we conducted several sets of experiments without/with different kinds of shoes. For each set of experiment, the process is the same as the one above. But because of the limitation of time, these experiments haven't been finished yet.

6. comparison with original design

Our original goal of the project is to design a system which could capture the movement of the foot and then store the data into database for future analysis use and build animation of the movement of foot based on the data the system collected. The system could capture the movement of different part of the foot so that the movement could be shown in a 3-D animation. We expect to let the tester to wear our gear and walk to collect data. The data will be collected when the tester is in following situations: barefoot, wearing ordinary shoes and wearing the certain boots. And we could compare the movement of the same tester when wearing different shoes.

And the system that we build now is capable of capture the movement of the ankle of the foot and the result will in text form. We then show the 2-D animation of the ankle movement using these data. We didn't build a database for storing the data either. Now we have done several tests of the system. We first let the tester wear the gear and walk in barefoot and collect the data. And then the tester is asked to wear the gear and put his foot in his shoes to get the data. And he is asked to wear two different kinds of boots when wearing the gear as the third and forth experiment. We then put the collected data into Excel and then use line chart to show the movement of the foot and do analysis based on the line chart. We also use a software called processing to show the animation of the ankle. Because our data collected is the change of relative angles of the foot and the leg, thus we use two lines to represent leg and the foot and build the animation based the data.

Our system is different from our original goal from three parts: first, we didn't capture the movement of the whole foot, instead, we capture the movement of the ankle only. We didn't achieve our original goal because to get the information for the whole foot, we need to test the movement of different parts of the foot and need to analysis how each data is related to each other and use these data to build the animation, which is too challenging for us. The second difference is that we didn't build a database, instead, we keep the data in text form. We do this change because right now we don't need to do complicated queries, but we may need to use database in the future. The last difference is that we build 2-D animation instead of 3-D animation. We make this change because we only measure the movement of the ankle thus 2-D animation is clearly enough to show how the ankle moves when wearing different kind of shoes.

7. Final Result

The sample of raw data collected is shown in Appendix C. We use these data to generate our line chart, which could show the difference between the movement of foot when wearing the boot and not.



Figure 21 – The movement of foot when wearing the boot



Figure 22 – The movement of foot without boot

From the contradict of these two pictures, we can see that when wearing the boots, the range of movement is smaller than with bare foot. Besides, the frequency of movement when wearing boots is lower than the frequency of movement with bare foot. The two difference may be caused by the weight of boot, which impedes the normal movement of subject.

8.Challenges

When designing the system, we have encountered several challenges.

Choosing suitable sensor is essential to our project and is not an easy task since the popular solutions available all fails to meet our demand as discussed in part (4). And we could only try to search the sensors we want and discuss whether the solution could succeed or not. And finally we came to the angular sensor and found it suitable for our system.

We try to use a wooden stick as part of our system. And then we discover that the wooden stick is too fragile and easy to break. To overcome this problem, we choose to use 3D printer to build a new stick.

Because we don't know how to use the 3D printer and thus we could only start from scratch and do some experiments to master the skills that we need to build the stick we want using the 3D printer.

We need to find a way to store our data when finishing collecting. We come to the solution using SD card because it is easy to store the data but a more suitable method would be using the database.

9.Conclusion

The goal of this project is to to capture the motion of foot when firefighters are wearing boots Such a motion capture system is successfully constructed and found to meet design specifications. Angular sensor is chosen as the most direct and suitable way to realize this system. One-unit design, which includes two plastic bars and one flexible angle sensor, is aligned with bodily axis so that the movement of body would drive the angular sensor to change at the same time. Once the signal from sensors are collected, an Arduino is used to collect data, store them locally with SD card and upload them to PC. Finally, raw data would be displayed and animated by software Processing. Therefore, the main goal for this project is accomplished.

Analyzing the data we collected, we could see that when wearing the boots, frequency of movement is higher than barefoot and we can see that the change in the relative angles is much smaller when wearing the boot. Thus, we could come to the conclusion that wearing the boots will restrict the ankle movement.

10.Future Work

Now the biggest problem for our system is that when we are walking, relevant shifting between the body part and the sensor may happen, resulting in obvious errors. Besides, to make sure the stick could be bent a little bit to fit the surface of our foot, flexibility must be guaranteed to some degree. However, this will cause the angle change not as accurate as it is supposed to be, since the stick may bend when we are walking. These problems must be solved in the future work in the first place.

For the optimization, now the system could only support data collected from one boot. It could be improved to support collecting data from several boots simultaneously. A database could be added to support storing and querying for specific data.

Also improving the way of data transfer could be a valuable change in the future. Now the data is transported by SD card. It could be optimized to use WIFI or Bluetooth to transfer the collected data.

For field use, we are expecting to expand the system to communicate to Bluetooth enabled server in fire house. Individual data for each firefighter would then be saved in the database for later analysis.

11. Acknowledgement

We can't finish this project without my advisor: Dr. Joseph Skovira and Dr. Huiju Park. Thanks for their patient guidance. Thanks for their advice when we can't find a way to design.

References:

1.Arduino datasheet from official webpage; Available from <u>https://www.arduino.cc/en/Main/ArduinoBoardUno</u>

2.Arduino Language reference; Available from <u>https://www.arduino.cc/en/Reference/HomePage</u>

3.Rotary Position Sensor SV03 Series; Available from <u>http://www.murata.com/zh-</u> cn/products/sensor/rotaryposition?intcid5=com_xxx_xxx_cmn_bc_xxx#header_0_RegionLangu agePanel

4.Processing software reference; Available from <u>https://processing.org/reference/</u>

5. Documentation for comparing the derived angles Gait Analysis: Normal And Pathological Function / Edition 1 by Jacquelin Perry, Bill Schoneberger

Appendix A: Code for Arduino

Module Name: Sensor Data collect

Date: 2015.11.20

Developer: Liwei Han

Description: This module code mainly takes responsibility of collecting the data from sensor and storing them on the SD card in .txt format.

```
#include <SD.h>
// cs pin for sd card shield
int sensorPin = 2; // cs pin for sd card shield
float value = 0.00; // sensor read-in analog I/O pin
float temp = 0.00; // value from sensor
float temp = 0.00; // trace?
                                         // transferred volatage value
float temp = 0.00;  // transferred volatage value
float angle1 = 0.00;  // transferred angle value every millisecond
float angle = 0.00;  // summed value of every 10 angle1 value
float ini_value = 0.00; // store initial sensor value
int count = 0.00;
                                     // counter for filter function
void setup()
{
 Serial.begin(9600); // set data rate 9600 bits per second
 Serial.println("Initializing Card");
 pinMode(CS pin, OUTPUT); // I/O pin 10 : output
 if(!SD.begin(CS pin)) // check SD shield state
   Serial.println("Card Failed");
  return;
 }
 Serial.println("Card Ready");
 String title = "angle"; // title in the output file
 ini value = analogRead(sensorPin); // record initial position's value
 // create test.txt in SD card
File dataFile = SD.open("test.txt", FILE WRITE);
 if(dataFile)
 {
    dataFile.println(title); // write title to the file
    dataFile.close();
 }
}
```

```
void loop() {
 value = analogRead(sensorPin); // value from sensor
 temp = (value-ini_value) / 214.6; // transfer to the voltage value
  count = count + 1;
  angle1 = temp * 60;
                                       // transfer to the angle value
 angle = angle + angle1;
                                       // sum angle1 value (used by filter)
 // average filter function
   if (count > 10) {
  count = 0.00;
   angle = angle / 10;
  File dataFile = SD.open("test.txt", FILE WRITE);
   if(dataFile)
  {
  dataFile.println(angle); // write filtered data to the file
  dataFile.close();
  Serial.println(angle);
   angle = 0.00;
  }
  }
 delay(1);
                                 // log the sensors every millisecond
}
```

Appendix B: Code for processing

Module Name: Processing Data Display

Date: 2016.04.15

Developer: Di Huang

Description: This module code mainly takes responsibility of displaying the data collected and processed by Arduino Uno in the format of .txt.

```
// declare vector used to draw lines
PVector center1;
PVector fix1;
PVector movel;
PVector center2;
PVector fix2;
PVector move2;
// declare buffer reader
BufferedReader reader1;
BufferedReader reader2;
// declare lines to draw
String line1;
String line2;
// declare angles
float angle1;
float angle2;
void setup(){
  size(640, 640); // set up the size of the screen
 // set up the path for readers
  reader1
                                                                       =
createReader("/Users/constance2587/Desktop/project/progressupdat
e/processing/bare.txt");
  reader2
                                                                       =
createReader("/Users/constance2587/Desktop/project/progressupdat
e/processing/boot.txt");
  // set up the start point of two lines
  center1 = new PVector(width / 4, height / 2);
  center2 = new PVector(width * 3 / 4, height / 2);
  frameRate(30); // set up the frame frequency
}
```

```
void draw(){
  background(0); // set up the background color
  try {
   // read data from .txt
    line1 = reader1.readLine();
    line2 = reader2.readLine();
  } catch (IOException e) {
    e.printStackTrace();
    line1 = null;
    line2 = null;
  }
  //once lines in .txt are null, stop reading
  if (line1 == null || line2 == null) {
    noLoop();
  } else {
    //interpret the read data to angles
    String[] pieces1 = split(line1, TAB);
    float angle1 = float(pieces1[0]);
    String[] pieces2 = split(line2, TAB);
    float angle2 = float(pieces2[0]);
   // calculate the end point of two sets of lines
    fix1 = new PVector(0, 200);
    fix1.add(center1);
    movel = new PVector(80, 0);
    movel.rotate(radians(angle1));
    movel.add(fix1);
    fix2 = new PVector(0, 200);
    fix2.add(center2);
    move2 = new PVector(80, 0);
    move2.rotate(radians(angle2));
    move2.add(fix2);
    //set up the propriety of the drawn lines
    stroke(255);
    strokeWeight(4);
//draw the lines
    line(center1.x, center1.y, fix1.x, fix1.y);
    line(fix1.x, fix1.y, move1.x, move1.y);
    line(center2.x, center2.y, fix2.x, fix2.y);
    line(fix2.x, fix2.y, move2.x, move2.y);
  }
```

Appendix C: sample of raw data

angle	angle
0.62	0
0.92	0
0.92	-0.62
1.23	-0.92
1.23	-0.87
1.23	-0.89
1.23	-0.92
1.23	-0.89
1.23	-0.89
1.23	-0.92
1.23	-0.92
1.23	-0.92
0.78	-0.92
0.34	-0.67
0	-0.62
-0.53	-0.62
-1.31	-0.62
-1.82	-0.62
-1.85	-0.62
-2.07	-0.62
-2.35	-0.62
-2.46	-0.62
-2.77	-0.62
-2.8	-0.62
-3.08	-0.62
-3.08	-0.62
-3.38	-0.62
-3.38	-0.62
-3.69	-0.39
-3.94	-0.31
-4	-0.17
-4	0

Appendix D: modeling for 3D printer

Module Name: Modeling for 3D printer

Date: 2016.02.10

Developer: Di Huang

Description: This solid figure is used for building the model for 3D printer to print. It is drawn with AutoCAD.

3S SOLIDWORKS	• 🔐 • 🎭 • 🗐 • 🛐 • 🚦 💣 🖾 •		连杆120-1	😵 Search SOLIDWORKS Help 🔎 🔹 🕞 🗃	1 22
1 🛨 🕲	Kannan/				
2					
Autodesk Simulati					
Features Sketch Evaluate DimXper	t SOLIDWORKS Add-Ins Simulation SOLIDWORKS MBD	Autodesk Simulation CFD 2015			5/2
% ፼ & △ >			of of 2 🖬 🐘 - 🕅 - où - 🕰 😻 - 👼 -		25
% ● 连杆120-1 (献认<<献认>_显示) 					1
- · · · · · · · · · · · · · · · · · · ·					6
					#
◇ 前祝基准面◇ 上収基准面					2
→○ 右抗基准面 ↑ 原本					
· · · · · · · · · · · · · · · · · · ·					
● 【易 台台-拉伸2					
1					
ľ					
	(75			
		/ [-			
	*				
1	to				
<					
SOLIDWORKS Education Edition - Instruction	牌例1			Editing Part Custom a	0