Final Project Report

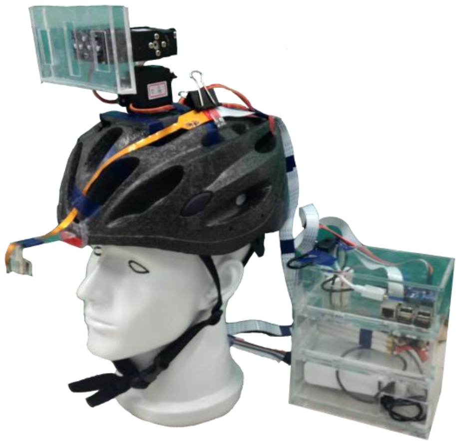
A Head Wearing Eye-Tracking Camera

Project Duration

Sept. 2014 – Dec. 2014

Submission Date

December 14, 2014



Prepared By

Group 20

Qian Wang (wangqian1992511@sjtu.edu.cn)
Liyu Wang (wly1993@sjtu.edu.cn)
Canchao Duan (duancanchao@sjtu.edu.cn)
Jiacheng Cai (sengkc@sjtu.edu.cn)
Yanrong Li (phl_416cat@sjtu.edu.cn)
Undergraduate Engineers, UM-SJTU Joint Institute, Shanghai Jiao Tong University

Prepared For

Prof. Thomas Hamade
Ve / Vm 450 Instructor, UM-SJTU Joint Institute, Shanghai Jiao Tong University

Prof. Xudong Wang
Project Sponsor, UM-SJTU Joint Institute, Shanghai Jiao Tong University

Executive Summary

Photographing is important but not always convenient in people's life. There are many situations that photographing is limited or cannot function as people want, even if people wear head-fixing cameras. In order to address these problems, we designed and manufactured a head wearing eye-tracking camera system. This camera can track the real-time motion of people's eyes and photograph the direction they are looking at. We expect the system can provide a platform for further applications such as social networks, navigation systems, and virtual reality.

By analyzing customer requirements and using QFD, we determined engineering specifications including system weight, viewing angle, response time, clarity of photo, focus center difference and battery life.

By brainstorming, we generated several concepts and specified one, and based on the concept diagram we designed the system. For hardware, the system is divided into on-head part and off-head part. The on-head part includes a helmet, servo combinations, an eye-tracking mini camera, and a cellphone as the main camera. The off-head part is a box containing two batteries, the central controller and the servo driver board. For software, we programmed one application on cellphone to take pictures, and three modules in Raspberry Pi: eye images taking module, eye-tracking module and servo control module. A top module is set in Raspberry Pi to call the submodules.

After manufacturing and assembly of the components, we tested the prototype. After a long-term optimization, all the specifications are met. The cost is also reasonable compared with our budget. We conclude that we have successfully finished this project, and left recommendations for further development of this project.

In this report, an elaborated description including all the process, analysis and discussion is presented.

Table of Contents

1. Introduction

1.1 Background

Photographing is an indispensable part in our life. It has two main functions: to preserve what people want to memorize, and to record people's interests. However, in many cases, traditional photographing by camera cannot satisfy our requirements. Imagine you are driving a car along a road through beautiful landscapes, and you want to record whatever you see aside the road. But you have no free hands to hold the camera, nor are you permitted to stop as you want. Similar situations happen when you are doing outside door sports as riding, skiing and climbing, or you are doing practical experiments or fixing things. Moreover, sometimes you meet interesting things frequently in markets, stores or exhibitions, but it is not an efficient way to hold your camera once and again to take whatever you like. A common end of a forced-photographing is usually an exhausted and bored customer, losing interests to the various displays.

In order to address the problem, people usually wear a camera on the head, and photographing the direction they are facing. However, that is not a satisfying solution. On one hand, turning head is not as convenient as people expected. When a traveler drives with a head-fixing camera, he/she has to turn your head toward the direction he/she wants to record, and this distraction may cause catastrophic traffic accidents. When a worker fixes things in a very narrow space such as underground tunnels or gas lines, it may be impossible to turn his/her head or body freely. On the other hand, our interests are often subliminal and transient. Before we are aware of turning our head and taking pictures, our interests may have been passed or disappeared, and head motion cannot follow our mind closely. In summary, a current head-fixing camera is not applicable properly and may even cause severe safety issues.

Figure 1.1.1. A head-fixing camera: Gear-Pro High-Definition Sport Action Camera

1.2 Proposed Solution

Fortunately, there is one organ of mankind that moves freely and truly reflects our interests – **human eye**. Even if we are driving, our eyes still move fast to scan the landscapes, while monitoring the road traffic. Even if we are stuck in a narrow space, our eyes still move flexibly in a large range. While we are hanging around, it is self-evident that our eyes keep at our interests. Thus, it is possible to address the above problem with an eye-tracking camera, that is, a camera taking pictures in the direction of our eye-sight, instead of our face. To simulate the eye-sight, such a camera should still be set on head, but it will not be fixed. So we name it a *head-wearing* camera instead of a *head-fixing* camera.

Intuitively, this system needs a camera to track our eye-motion, and a rotation device should control the main camera based on the information from the eye-tracking camera. The main camera of the system should take pictures automatically, and save the pictures or upload it to cloud for further usage.

1.3 Customer Requirements

We gathered information from oven 20 students in UM-SJTU JI and we believe the following customer requirements should be met:

- As a wearable smart device, the prototype should be **comfortable** to wear and **easy** to carry.
- The response of the system should be **quick** if people's eye moves in certain pattern.
- The photographing **viewing angle** should be large enough so that it can take pictures even if the user is looking sideways.
- The operating system should be **compatible** to most cellphones[1] in order to permit the user to flexibly use their own cellphone as the main camera.
- The photo should be **clear**.
- The photo should be **valid**, that is, including and centering the object the user wants to shoot.
- The system should work for a **long time** in case it is inconvenient to get charged.

[1] The reason we use cellphone as our main camera will be explained in Section 3.3.2

1.4 Project Significance

This project is an innovative promotion of the traditional head-fixing camera. With this project, a user will have higher freedom and more convenience to take pictures. Besides personal usage, a museum or supermarket can ask their visitors and customers to wear this device to gather the information of their interests, so that analyze their favors and preferences, and improve their displays and commodities with targets. Psychologists can use it to analyze people's psychological activities betrayed by eye motion. Directors and artists can also use it to analyze the audience's interests to improve their compositions. Since the camera can simulate people's eyesight, it can be further developed as an environment monitoring system[2] such as on automobiles to avoid accidents, on navigation system for the blind, or even on military applications. Since it records people's life, it can serve for social networks such as Facebook and Twitter. It can also contribute to virtual reality in sci-fi movies *Robocop* and *Iron Man*, in which people can get all information just through a helmet. This distributed and ubiquitous computing system is an indispensable fundamental for next generation communication systems.

1.5 Related Work

The existing related work is mainly about eye-tracking algorithm. In [1], the eye tracking system uses both shape-based and appearance-based methods to measure the position of eye by using the images which are captured by the side-equipped cameras. The eye images are in low resolution of 300 pixels by 300 pixels. By testing seven subjects, the result succeeds in the rate of 95.7% for gaze estimation. In [2], it introduces a special eye tracking system in low power consumption based on FPGA implementation. The real-time system has 88% eye tracking success rate at 8f/s image rate. Due to its energy limit feature, it will perform well in an embedded system or portable product. In [3], a new way of eye tracking is applied in a smart camera. Previous works always involve in other hardware to process the image while the real-time eye tracking system integrates the whole algorithm into the smart camera itself. In future, the algorithm can be improved to be used in varying conditions. However, for the previous works, a novel integration of eye-tracking and camera is not implemented. In [4], it really takes advantage of the eye and gaze tracking system for further application. The users behaviors are carefully recorded when they are browsing. By combining the feedbacks from the eye tracking system and some other factors, such as Wi-Fi, the whole system can offer data for deep study, such as customers' shopping habits. But there is a serious problem: if the object is not in the center, it will be out of focus. Obviously the usage of these pictures is limited.

[2] This potential usage is raised by Prof. Thomas Hamade, UM-SJTU JI.

Our prototype realized the real-time continuous eye-tracking algorithm. The purpose of our prototype requires that this algorithm should be low-cost in resources, from the view of time and memory. Given those particular scenarios in section 1.1, our implem entation found a balance between the accuracy and resources in reality. The detailed d iscussion will be in section 4.3.2.2.

2. Engineering Specifications

In Section 1.3, eight customer requirements are proposed, which include “comfortable to wear”, “easy to carry”, “quick response”, “large viewing angle”, “compatibility of OS”, “clarity of photo”, “validity of photo” and “use for a long time”. Here we convert the customer requirements to engineering specifications and quantify them.

In order to achieve the user-friendliness in convenience and easiness, the most important factor should be **weight**. It is required not only to have a low **total weight**, but also a satisfying **weight on the user’s head**. Especially, to make the prototype comfortable to wear, we need care more about the weight on the use’s head. Thus, we can derive the corresponding value on the first two rows of QFD.

“Quick response” means the device should spend as less time as possible to adjust its viewing angle correctly. It is known, commonly, that if one individual is looking at one point for more than two seconds, then it can nearly be concluded that he or she is gazing at this object. Thus, one second is set as the **time to rotate the servo by 180 degrees**. So, one more second can be used to take pictures.

Generally, human beings have a horizontal **viewing angle** of 220 °and a vertical viewing angle of 120 °. After we consider the shooting angle of the cell phone, it is nature for the servo to have a horizontal viewing angle of 120 °and a vertical viewing angle of 120 °.

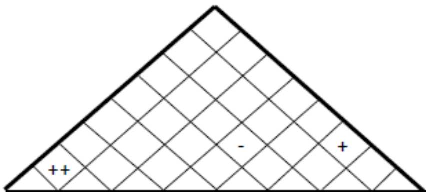
If the user wants to use this device for a long time, definitely the **battery life** should be as long as possible. Moreover, nearly all the operations influence the battery life. It is obvious that the rotation of servo, the process of eye-tracking system and the shoot of photo needs power. Therefore, the more complicated the eye-tracking system is, the more power is needed by the battery.

When using our prototype, the customers may not want to buy a new cell phone. This requires our prototype to be compatible with most cell phones. It is impossible to fulfill all the customers’ needs because they may use different OS. According to the idea of agile development, we should ensure that our prototype should be acceptable for the largest group of customers. That is why, “compatibility of OS”, which means **the large market share**, has been proposed.

The requirement of “validity” means the photo taken by the cellphone should at the same direction as what the user expects. Since our camera is expected to run in an auto-focusing mode, the precision of the focal length is completely determined by the precision of the focusing center[3]. The **focusing center difference** is calculated by the angle between one line connecting the eyeball to the real focusing point and the other line connect the eyeball to the focus derived by the device.

Even if the focus is precise enough, this cannot be equivalent to the saying that the focusing is acceptable. The user always pursues the clarity of the photo as well. To quantify this specification, the **Robert edge detection operator** can be used to judge whether the photo around the focus is really clear enough or not. The calculation of this value will be introduced in Section 6.3.

Quality Function Development (QFD)



Customer Requirement	Weight	Weight(head)	Weight(total)	Robert edge detection operato	Focusing center difference	OS Market Share	Time to rotate 180 dgr	Total rotatable angle	Device battery life	*LLSM Eye Tracking Sys.*	*A LP FPGA of Eye tracking*	*RT eye tracking using SC*
Comfortable to wear	8	9	9	0	0	0	0	0	0	5	1	1
Easy to carry	7	3	9	0	0	0	0	0	0	5	1	1
Quick response	2	0	0	0	0	0	9	0	0	5	5	5
Large viewing angle	1	0	0	0	0	0	0	9	0	1	3	3
Compatibility of OS	3	0	0	0	0	9	0	0	0	3	3	3
Clear photo	4	0	0	9	0	0	0	0	0	3	3	3
Validity of photo	5	0	0	0	9	0	0	0	0	1	5	5
Use for a long time	6	0	0	1	1	1	1	0	9	1	5	5
Measurement Unit		g	g	-	dgr	%	s	dgr	day			
Target Value		1000	2000	20	5	80	1	120	1			
Total		93	135	42	51	33	24	9	54			
Normalized		0.24	0.35	0.11	0.13	0.09	0.06	0.02	0.14			

(Benchmarks: LLSM Eye Tracking Sys., A LP FPGA of Eye tracking, RT eye tracking using SC)

Figure 2.1 Quality Function Development

We used the method of Quality Function Development (QFD) to convert the customer requirements to engineering specifications and to identify the importance of each specification.

[3] This is to respond the third comment in DR #1.

Weight	2000 gram in total (1000 gram on head)
Battery Life	1 day under reasonable use
Focusing Center Difference	5 °
Edge Detection Operator	20 (for 240*320 pixels)
OS Market Share	80%
Response Time	1 second
Viewing Angle	120 °horizontally and 120 °vertically

Table 2.2 Engineering Specifications

3. Concept

3.1 Concept Generation

To implement an eye-tracking photograph system, the following components are needed:

- **Hardware base**, which provides a platform to all the other components.
- **Eye-tracking system**, which is used to track the eye and send the eye image to the central controller.
- **Central controller**, which processes the eye image, judges the eye position, and control the rotation and photograph of the main camera.
- **Rotation device**, which rotate the loaded main camera towards certain direction.
- **Main camera**, which photographs.

Figure 3.1.1 shows our concept diagram.

With brainstorming, we listed some candidates for each part.

For hardware base, we may choose riding helmet and head band.

For eye-tracking system, we can choose one or two cameras (depending on whether we want the camera to track one eye separately or two eyes simultaneously). We can also choose whether to use a mirror to change the light path.

For central controller, we have a large number of choices, and we listed three as our candidates: Arduino, Microchip PIC32, and Raspberry Pi.

For rotation device, we need the device to provide two degrees of freedom, so that the camera can move both horizontally and vertically to cover a large angle of view. We listed two candidates: a pan-tilt used in model airplane and a servo combination.

For main camera, we wish it can be controlled by programming. The picture quality should be high. So we listed several candidates: mini-camera, cell phone, and digital camera.

The morphological chart shows that any combination of the components can work.

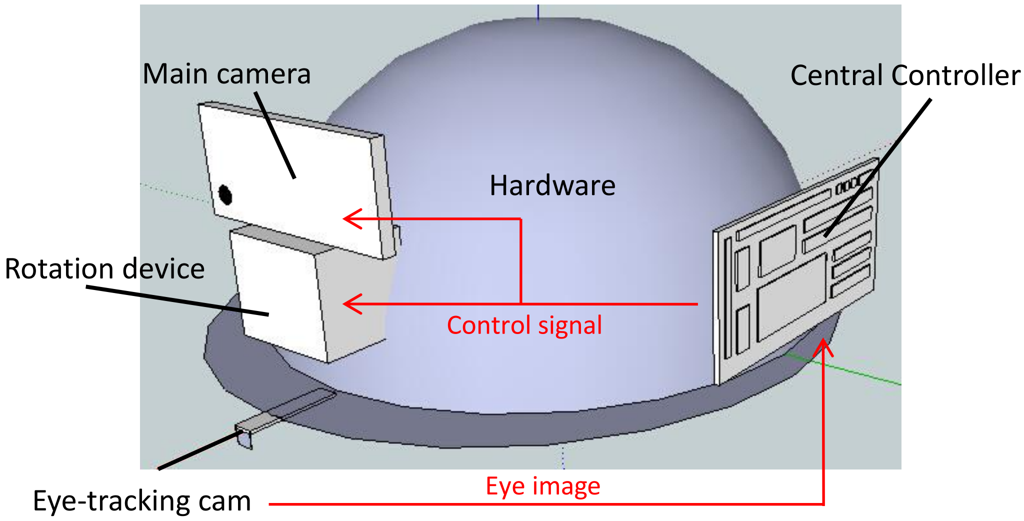


Figure 3.1.1 Concept diagram

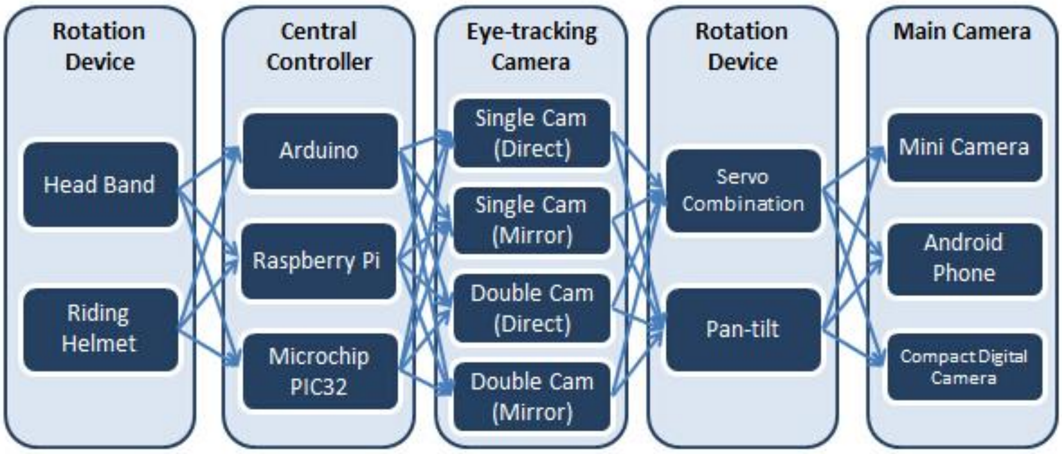


Figure 3.1.2 Morphological chart.

3.2 Concept Selection

Among all combinations, we analyzed the pros and cons of each candidate, and chose the most fitting one.

For hardware base, we chose the **riding helmet**. Typically, a riding helmet is made of Expandable Polystyrene, light and solid, which satisfies the requirement of safety and comfort. Head band, however, is not stable. We tried to wear a headband, but it slipped. Moreover, the head band does not provide so much space to fix the other components.

For eye-tracking system, we chose **one single camera without a reflection optical system**. Although a reflection optical system, such as a mirror, permits us to set the eye-tracking camera closely to the base instead of being suspended in front of our forehead, and reflects the eye image into the eye-tracking camera, however, it is harder to stably suspend a mirror than to suspend a mini-camera. The reason we don't use two separate cameras is because two cameras requires two microcontrollers to process the image, and that will increase the complexity of both hardware and software very much.

For central controller, we chose **Raspberry Pi**. Arduino is open sourced but the performance is not quite stable, and the processing unit is the weakest among all candidates. Microchip PIC32 has strong performance, but it is closed sourced. Given that we will integrate and transplant many libraries onto the controller, an open source should be a better choice. Also, neither PIC32 nor Arduino has an operating system, which provides no convenience to organize the program. Raspberry Pi is open sourced and has an embedded Linux operating system. It meets all of our requirements.

For rotation device, we tried both the pan-tilt and the servo combination. In test, pan-tilt showed an unexpected function: the inner gyroscope will adjust its position automatically, which was not a required function. If we banned the gyroscope, the feedback control mechanism will be gone. While a servo combination is concisely-designed and easily-controlled. It is also lighter than the pan-tile. Therefore, we chose **servo combinations** as our rotation device.

For main camera, a mini camera is small and light, but the image quality is poor. Some compact digital cameras such as CANON of NIXON are compatible to the Raspberry Pi and can take extremely high-quality pictures, but the weight and volume are unaffordable. Therefore, we chose a compromising device: an **Android cellphone** as the main camera. Cellphones take high-quality pictures, and are not too heavy or too large. Applying an Android cellphone also expands users' freedom, because one can put his own cellphone on the system instead of being forced to use an unchangeable one. Most importantly, an Android cellphone has infinite potential for development. We can program the cellphone to process the image, or upload it to the cloud by Internet connection, which is indispensable for future usage such as data mining and mobile computing.

4. Design

4.1 Overview

After we specified the concept, we designed the system. We divided the design process into two steps: **hardware design** and **software design**. The hardware consists of two parts: on-head part and off-head part. The software consists of four modules: a top module scheduling three sub-modules, which are eye-tracking, servo control and Android photograph app.

4.2 Hardware Design

4.2.1 Issues & Approaches

We sorted the problems in hardware design into five categories:

- Specify the model of each component.
- Decide which components are on-head and which are off-head.
- Decide how to integrate the on-head components.
- Decide how to integrate the off-head components.
- Decide how to connect on-head part and off-head part.

To do hardware design, we did measurements, drew some ancillary components with CAD, and did some basic tests for different designs.

4.2.2 Design Description

4.2.2.1 On-head part and off-head part

At the very beginning, we specified all the components according to our requirement, considering the weight, volume and function. Since we need to decrease the weight of on-head part as much as possible, we list the necessity of all components to be on-head from the highest to the lowest. Among them, the latter three can be off-head with wired or wireless connection. For these three things, we put them in a bandbox carried in a backpack.

Component	**Model**
Eye-tracking camera	OV5647 5Mpixel mini-camera for Raspberry Pi
Main camera	MIUI 2S

Rotation device	MG995 servo * 2 & servo frame
Hardware base	TOSUOD T4010002 Riding Helmet
Battery I	Pisen Powertwins 12000mAh Powerbank
Battery II	ACE 11.1v 2200mAh battery pack
Central controller	Raspberry Pi II Model B+

Table 4.2.2.1 Components and models.

4.2.2.2 Set cellphone onto the servos

The servo combination provides a long-U-shaped frame which is fixed on and can rotate with the servo axis, so if we set the cellphone on the frame, its direction can be controlled as we want. For the long-U-shaped frame, the material is aluminum board with thickness of 2mm.

In order to help fix, we drew a set of ancillary components with CAD. It is basically a transparent box, and its size just fits the cellphone so that the phone will not sway. At the front face of the box, that is #1, two rectangle holes were opened to expose the camera and the rectangle hole in the middle is used for putting in and taking out the phone. Here we use MIUI 2S, and the corresponding size of the box is 132mm*66.5mm*12mm.

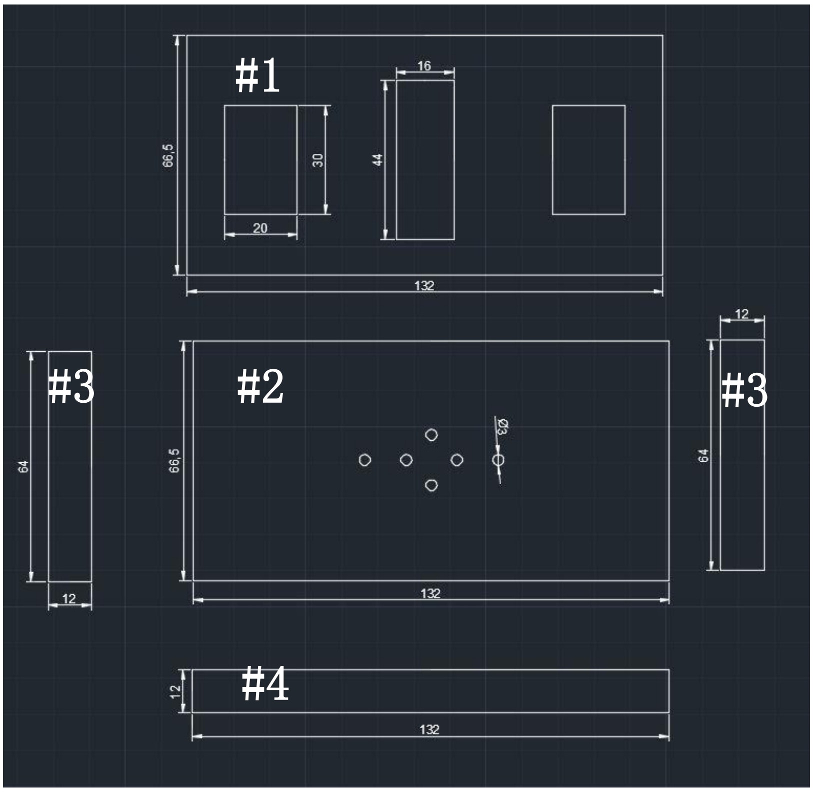


Figure 4.2.2.2 CAD scheme for box to hold the cellphone.

4.2.2.3 Set servos onto the helmet

We drew another set of ancillary components with CAD, which provides connection between servos and the helmet. Since we need a plane to fix the servos, we also need to repair the top of the helmet, because it was not flat enough automatically. We fix the servos in #5 by screws and nuts and fix the #6 in the helmet by hot melt glue. Between #5 and #6, we used glue to combine them together.

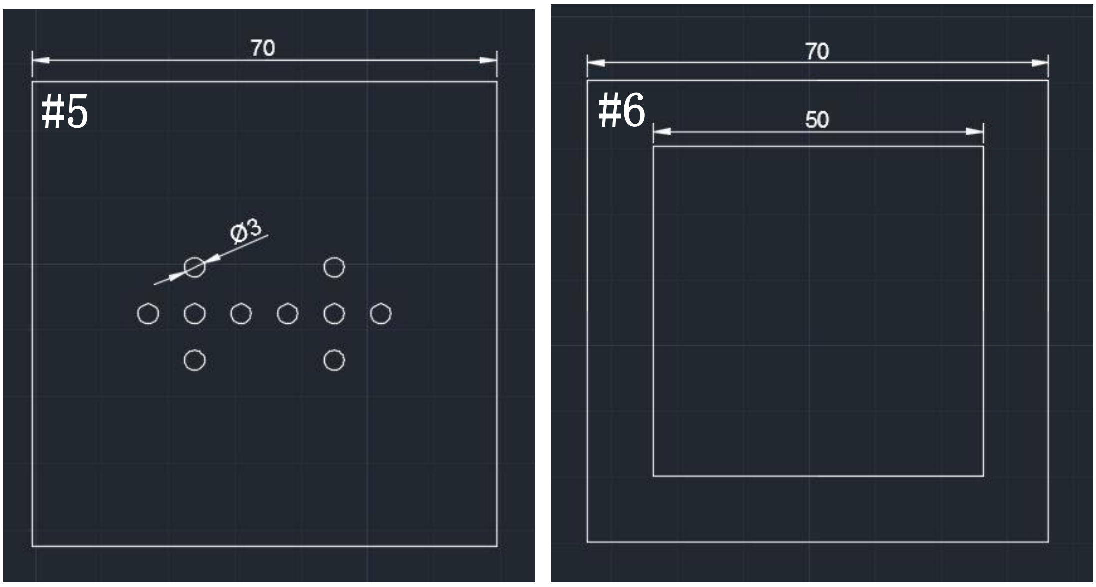


Figure 4.2.2.3 CAD scheme for connection component between servos and helmet.

4.2.2.4 Set eye-tracking camera onto the helmet

The eye-tracking mini camera needs to catch eye image directly, so it has to be suspended in front of our forehead, thus has a line-of-sight vision to our eyes. In order to keep the suspension stable, we designed two bases. #8 was for fixing the mini camera and #9 was fixed onto the helmet. We connected these two bases by a long rectangle component, #7, which extended the mini camera to the front of the eyes.

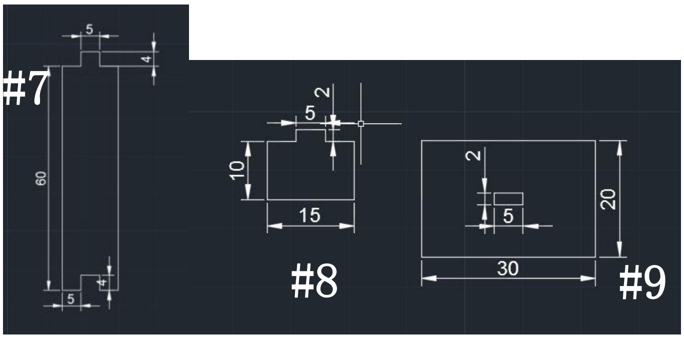
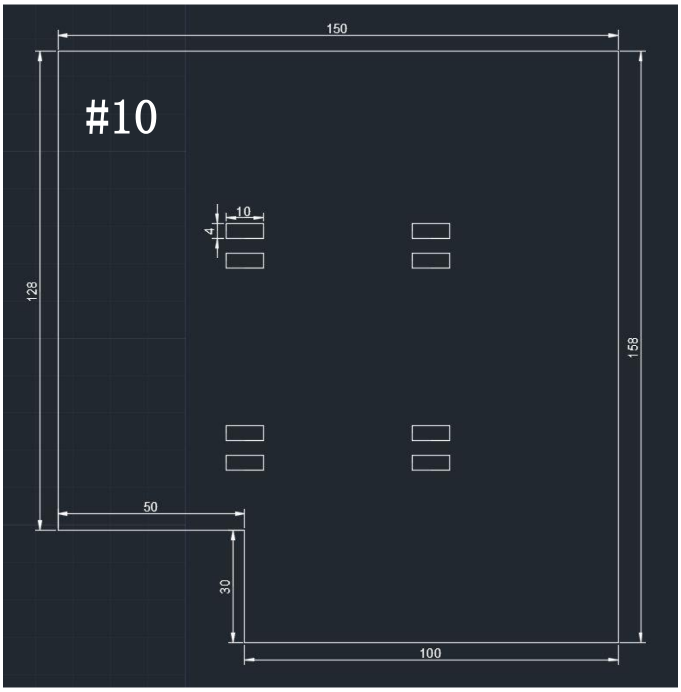


Figure 4.2.2.4 CAD scheme for connection component between mini camera and the helmet.

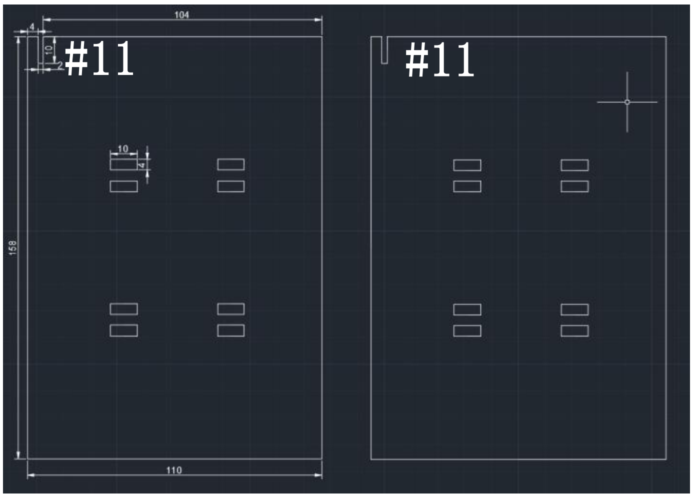
4.2.2.5 Integrate Off-head Part

The off-head part, according to our design, contains two batteries and the microcontroller. Moreover, one of the batteries cannot directly apply power to servos because of an unmatched voltage, and the servos need driver to directly get controlled. So we needed an extra servo driver board, which we chose 11.1V to 5-6V servo control extended board and inserted it onto a piece of bread board. In order to carry all the components conveniently, we need to integrate them. We designed a box with size 150mm*110mm*158mm, within which we attached the boards and batteries. On the box, there are holes to stretch out the wires. There are three clapboards so that we can put in and take out all the components conveniently. This box can be carried in a backpack.

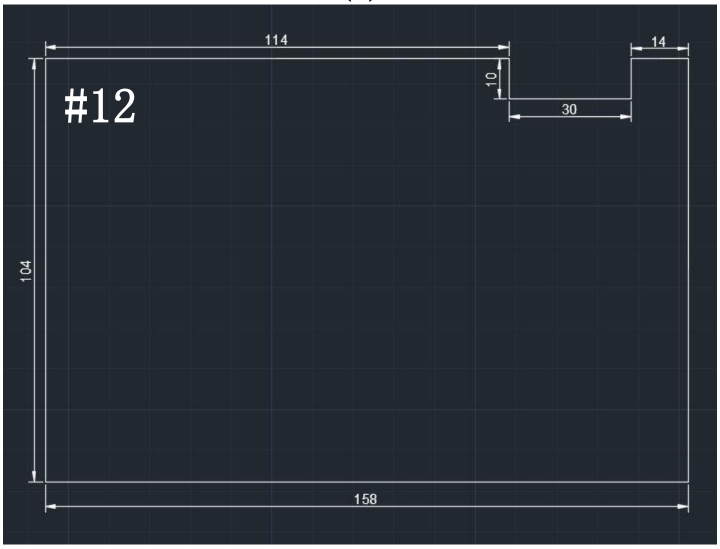
The main external frames were made of five rectangle boards from #10 to #13 with different size of holes. These five boards were stuck together by the 502 glue. These holes were especially set for fixing other components. #10 is the back face, #11 is the side face, #12 is the upper face and #13 is the bottom face.



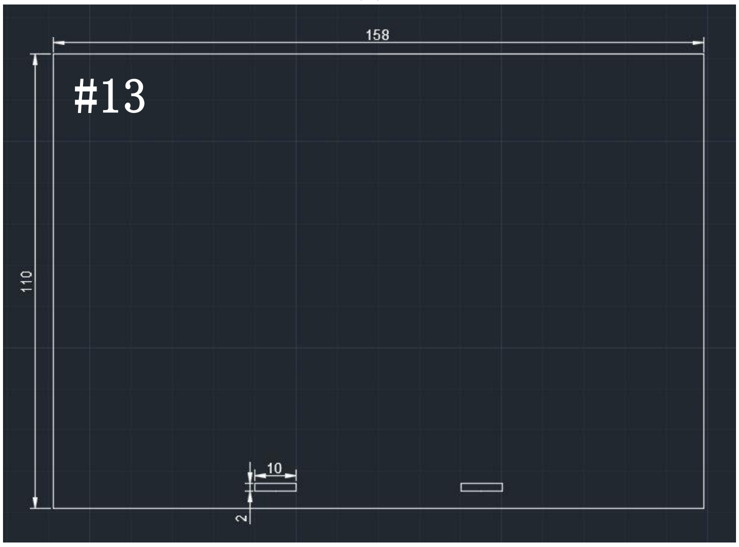
(a)



(b)



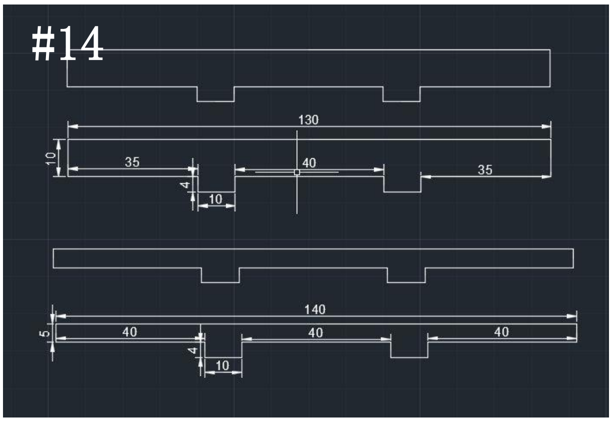
(c)



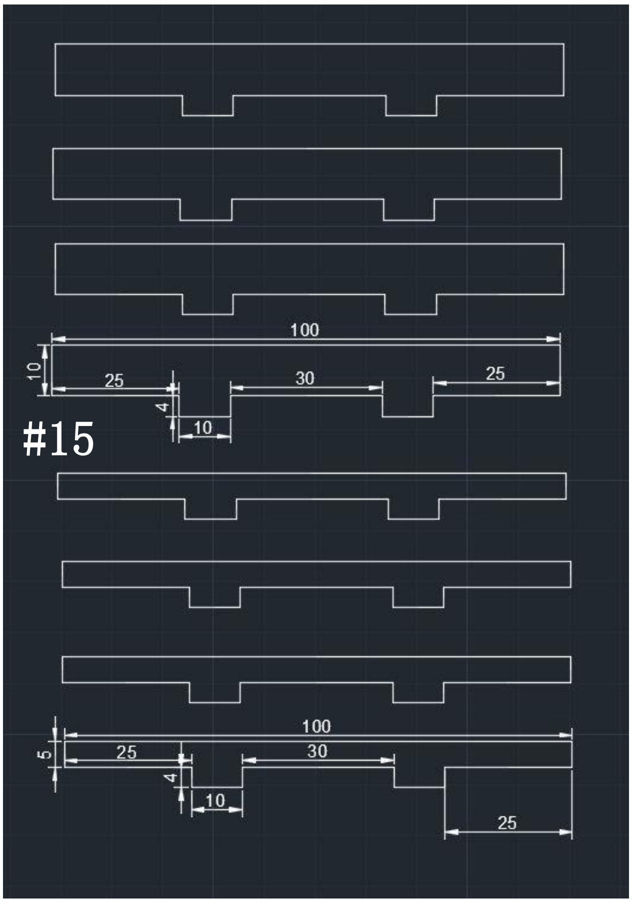
(d)

Figure 4.2.2.5.1 CAD scheme for off-head box (part I).

The following twelve supporting components are used to hold the clapboards. They were fixed into the rectangle holes of the back and side frame boards by the 502 glue. #14 is the supporting components for #10 and #15 is for #11.



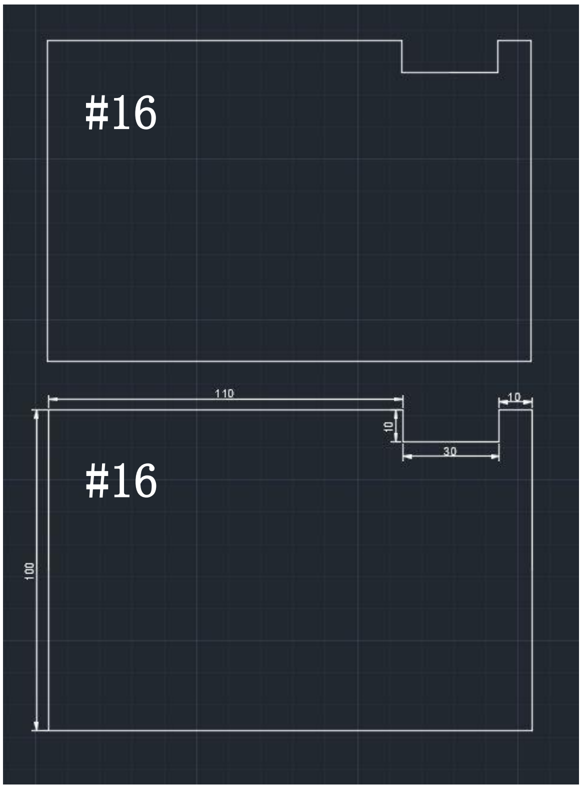
(a)



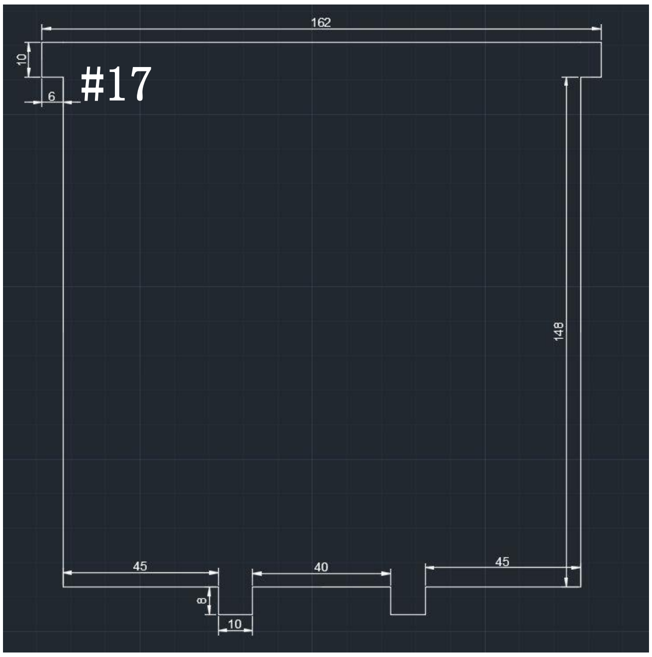
(b)

Figure 4.2.2.5.2 CAD scheme for off-head box (part II).

The followings are three clapboards and all of them are movable. They were installed and connected into the box by the supporting components and specially-designed rectangle holes. #16 is the clapboard inside the box and #17 is the front face.



(a)



(b)

Figure 4.2.2.5.3 CAD scheme for off-head box (part III).

4.2.2.6 Connect On-head Part and Off-head Part

Due to the connection demand of the on-head part and the off-head part, we need to extend the Flexible Printed Circuits (FPC) between the eye-tracking camera and the Raspberry Pi board. The connection joint of two FPCs are extremely fragile so we need to fully combine the contact surface of the two FPCs. Therefore, we designed #18, and the assembly of #18 will be discussed with details in Section 5.2.

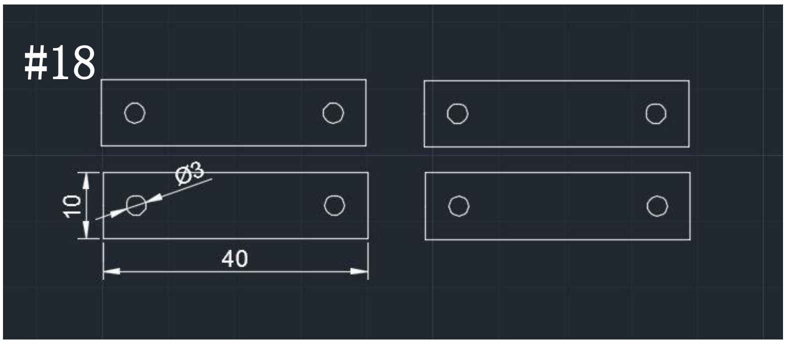


Figure 4.2.2.6 CAD scheme for fastening components of extended wire

4.3 Software Design[4]

4.3.1 Issues & Approaches

We sorted the problems in hardware design into five categories:

- Decide how to locate the eyes for different users.
- Decide the center of two eyes.
- Decide how to rotate the servo according to the eye-image processing.
- Decide how to control the cell photo to take a picture.
- Decide how to increase the efficiency of the whole system.

To do software design, we designed a low-cost eye-tracking algorithm and PWM based servo control with python, and implemented an Android App to control photographing. All the modules are coordinated with multi-thread programming.

[4] This section is to response the first comment in Design Review #3.

4.3.2 Design Description

4.3.2.1 Top-level module

To enhance the operating speed of our prototype, we use multi-thread programming. There are several benefits to do so. Firstly, we could simply initialize the eye-tracking camera only once and use the sequence-shooting mode. Moreover, the sampling rate to track the eyes is increased. Lastly, the whole program becomes easier to maintain. Figure 4.3.2.2.1 shows the flow chart of the top level architecture. In the following parts, the implementation of eye-tracking, servo control and Android photograph app will be discussed separately in details.

4.3.2.2 Eye-tracking algorithm

There are two threads related to eye-tracking. One thread takes charge of taking the photo of user's eyes continuously. The other thread is able to find the exact location of user's eyes and their corresponding centers.

Photographing user's eyes

In the first thread, the camera should first be initialized. The initialization takes a long time. Hence, to improve the system performance, frequent initialization should be avoided. Thus, sequence-shooting mode is applied to the design. After the initialization, the camera will take photos for use's eyes continuously. These photos will be used in future process.

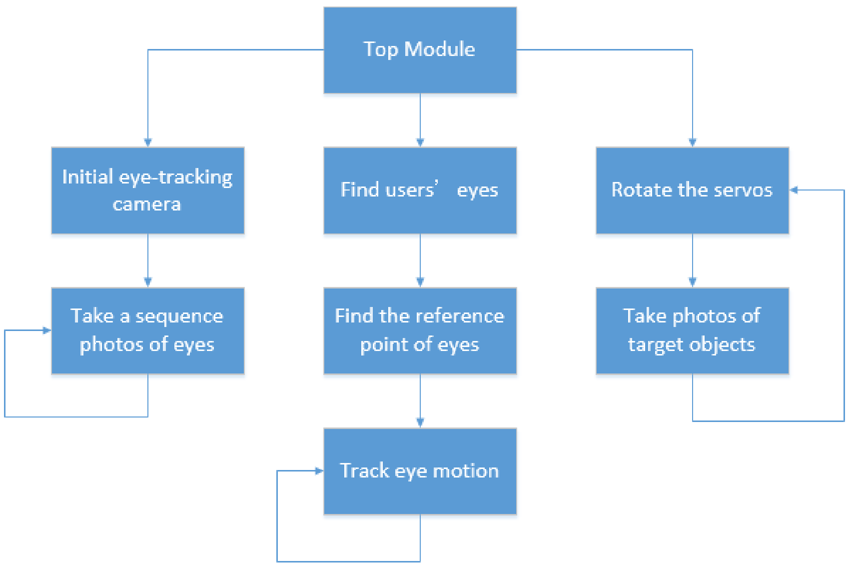


Figure 4.3.2.2.1 Software flowchart of the complete system.

Locating user's eyes

Different users may have different location of eyes. So, in the first step, it is necessary to localize two eyes. This part is following the idea that eyes are the organ always in motion, while others are basically static. This kind of motion will leads to an obvious fluctuation of the RGB value on some pixels. Thus, we can make use of this phenomenon to capture the regions of two eyes.

Firstly, we should take five pictures when the user is looking up, down, to the left, to the right and straight ahead. The large fluctuation of the RGB value on one pixel can be shown by the large difference of the RGB value of these five pictures on that pixel. For each pixel, the average unsigned integer value of each RGB channel is calculated. Then, the difference between this average value and the corresponding channel of that pixel on all these five pictures are also calculated. By now, fifteen values of difference can be obtained for each pixel. They should be added together to indicate the level of fluctuation of that pixel. All these sums will be stored in a matrix with the same size as the picture.

For convenience, we set a threshold and derive a polarized picture. The values stored in the matrix mentioned above are linearly mapped to unsigned eight-bit integers, from 0 to 255. According to our experiment, we use the threshold of 230 to polarize the matrix. More precisely, all the values which are smaller than 230 will be set to 0, while the others will be set to 255. The following figure shows an example of a polarized figure.

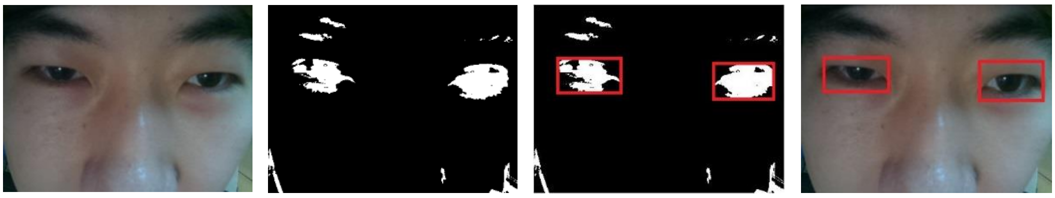


Figure 4.3.2.2.2 Eye-image processing: finding eye position.

From this figure, we can easily point out the position of two eyes from the two largest white areas. According to this thought, we use *floodfill algorithm* [5] to find out two largest white areas. For each white connected region, we should record the upper bound and the lower bound of horizontal and vertical direction. By now the region of each eye can be determined.

At the same time, this localization process is also used to calibrate the eyeball's motion. Each time the eye moves toward one direction, the position of the eye is recorded. These recorded values will be used as a set of references for eye-tracking. Detailed explanations are given in Section 4.3.2.2.3.
Considering the reference values provided by the calibration will affect the successive eye-tracking remarkably, this process must be precise. In real tests, we found a set of

locations for the user's eye to look at, tied a laser on the servo, and commanded the servo to rotate to the corresponding directions during the process. In this way, the laser can guide the eyeball of the user to finish the value localization and calibration.

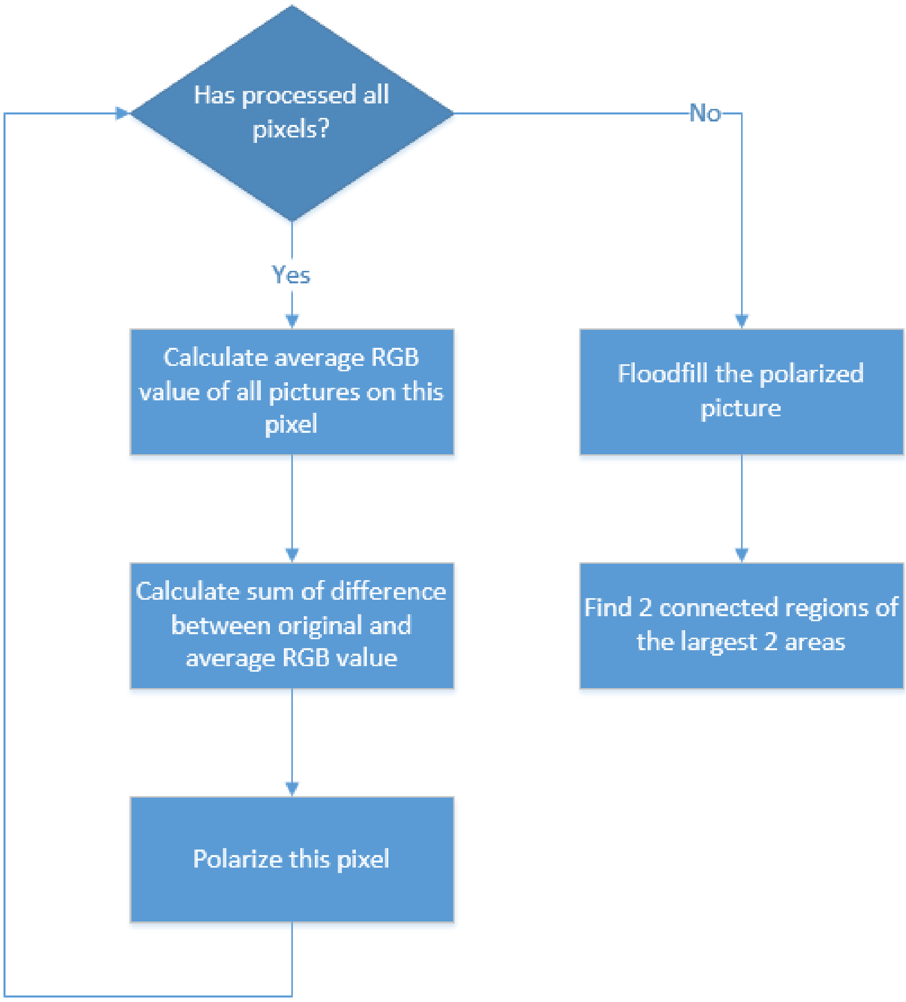


Figure 4.3.2.2.3 Locate the user's eyes

Finding the original centers of eyes

Specially, we should exactly indicate the centers of user's two eyes when he or she is looking straight forward. These two points can be used as two reference points when tracking the motion of eyes. This method will also be used to find the real-time centers of user's eyeballs. Therefore, it is required that it should be as fast as possible, as long as it can provide a satisfying result.

Since the region of user's eyes has already been determined in the last part, we do not need to process the whole picture now. Instead, this region, which consists of two rectangles, can be cropped and processed separately. For convenience, user's right eye is used as an example here.

The picture is converted into grey-scale mode at the beginning. This will reduce the amount of data by one third. After that, the picture will be polarized as what we mentioned in the previous part with a different threshold. Since we are concentrating on the black area now, we set threshold as 30 instead of 230. Then the contour of an eye will be clearly presented.

To fix the location of the center of the eye, we accumulate the grey-scale value of each row and each column on the polarized picture. The eyeball is approximated into a circle on the polarized picture. Hence, we can use the intersection of two diameters as the center of this circle, the eyeball. Fortunately, each diameter can be determined by the row or the column with the largest accumulated grey-scale value.

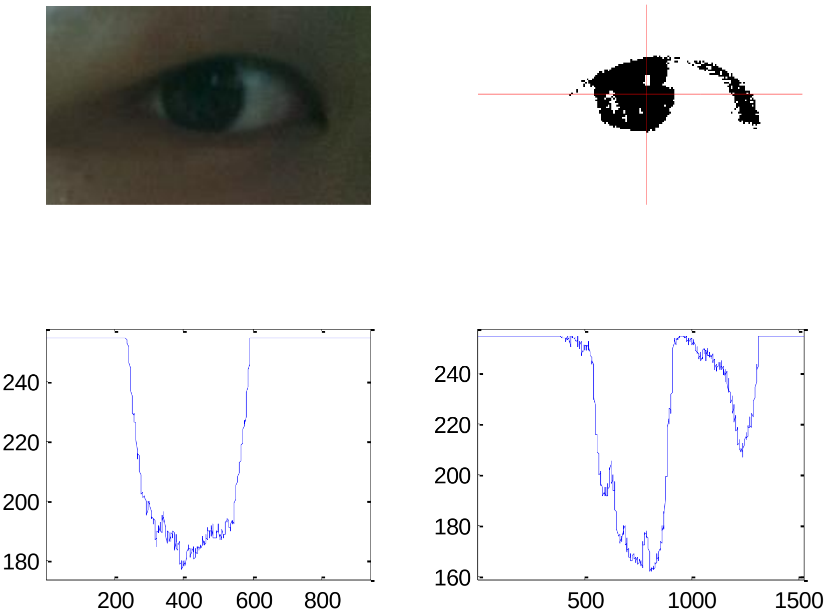


Figure 4.3.2.2.4. Find the original centers of eyes.

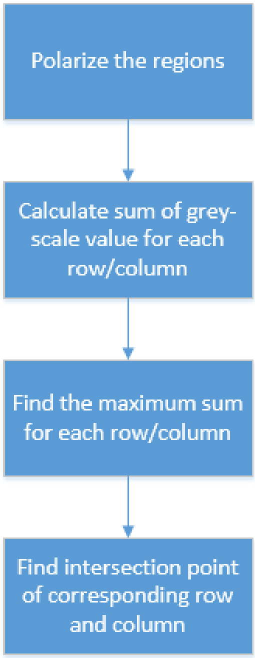


Figure 4.3.2.2.5. Flowchart of finding the center of the eye.

4.3.2.2.3 Tracking the motion of eyes

By using the method mentioned in the previous part, we can find the coordinates of the center of eyeball arbitrarily. Since we have gained the reference points for both left and right eye from calibration, we can calculate the displacement of the eyeballs as well. According to the experiment, we find that the range of horizontal eyeball displacement is 110 pixels, and the range of vertical eyeball displacement is 56 pixels. By linear mapping, we can derive the relation between the displacement of the eyeball and the rotating angle of the servo.

$$\theta_{horizontal} = \frac{90 * (55 + P_x - C_x)}{55}$$

$$\theta_{vertical} = \frac{90 * \left(28 + P_y - C_y\right)}{28}$$

The eye position is calculated at each shot of eye-image, which is 0.5 second according to our test, but not until the eye gazes at somewhere will the cellphone take pictures. Therefore, we calculate the square deviation of five successive eye positions. If they are smaller than some threshold, the photographing command will be triggered, which means a longer-than 2.5 second static eyeball position

4.3.2.3 Servo control

The Microcontroller could not drive the servo directly, so we need an additional servo driver board to supply enough power for the servos.

Board to Board connection
Here we use I^2C to build up board to board communication.

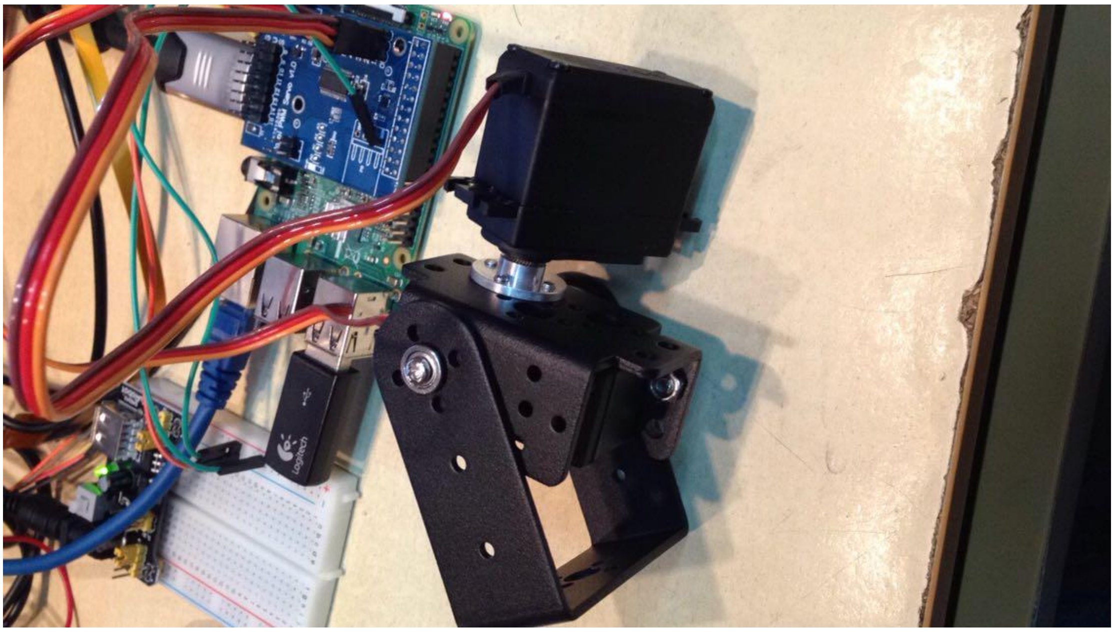


Figure 4.3.2.3.1 Connect Raspberry Pi with servo driver board.

As the Raspberry Pi has the I^2C standard interface, we just enabled the I^2C connection and the system library will automatically adjust the frequency and the frame for us.

On Board control
As we are using python for the whole system construction, we found a python driver for the servo control board. We justified the application interface so that it could control the board and achieve a rotation with an error less than 0.4 degree.

The final version of the interface is:

```
pwm.setPWM(int servo_num, int base, int target_voltage)
```

The range of target_voltage is integer from 150 to 600, corresponding to 0-180 degree.

4.3.2.4 Android photograph app

Network Protocol

The communication between Raspberry Pi and Android phone can be explained through OSI Model.

The wireless connection standard is defined in Layer 1-2 as IEEE 802.11. The network socket, including logical IP address and port is defined in Layer 3-4. A socket can be viewed as <IP, port> pair. This kind of relationship is also defined as TCP/IP protocol standard. As our project is based on network standard, existing software can be used to construct the network infrastructure. The Android application, which is a completely customized application, is working in Layer 7, and supported the lower layer.

To summarize, our project followed the standard of OSI model [6] to construct the network infrastructure using wireless connection, TCP/IP protocol, and we developed the Android application in Application Layer to be supported by network infrastructure.

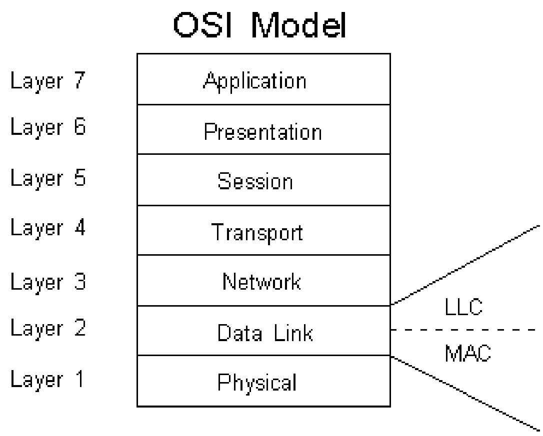


Figure 4.3.2.4.1 OSI Model for communication network.

Wireless Connection

Raspberry Pi B+ does not support wireless network in native. We extended its function by adding a wireless module. Working with this module, the Raspberry Pi is able to work in IEEE 802.11 AP standard mode. It turns Raspberry Pi into a wireless router, which can be connected by various types of devices, like mobile phones, notebooks.

The wireless module Ralink RT5370 is natively supported by Raspberry Pi. It means we do not have to compile the driver. It saves lots of time by skipping the driver installation, and directly jumps to the network configuration.

Figure 4.3.2.4.2 Antenna of Ralink RT5370

Model	Ralink RT5370
Wireless Standards	IEEE 802.11n, IEEE 802.11g, IEEE 802.11b
Frequency Band	2.4GHz ISM (Industrial Scientific Medical) Band
Peak Throughput	150Mbps
Channel	1 - 14 channels (Universal Domain Selection)
Data Security	16/128-bit WEP Encryption, WPA, WPA-PSK, WPA2, WPA2-PSK, TKIP/AES
Host Interface	High speed USB2.0/1.1 Interface
Antenna Gain	2dBi

Table 4.3.2.4.1 Parameters of the antenna Ralink RT5370

The Linux OS supports networking in native, but it does not support wireless 802.11 Access Point mode, and dynamic IP address allocation by default. We installed Hostapd to work in AP mode, and Udhcpd to allocate IP address.

```
sudo apt-get install hostapd udhcpd
```

By properly editing the configuration files, raspberry Pi turns into a network router.

Development of Socket Camera

The Android application is a camera photographing triggered by a network request. We call it “Socket Camera”. In order to develop the application, the computer was configured to the required environment. We also defined the software requirements as target to achieve.

Development environment

- Android Studio
- Android SDK
- Android Emulator
- Java Development Kit (JDK)

- Java Runtime Environment (JRE)

Software requirements

- Always listen for network requests
- When a network request is received, photographing is triggered.
- Return to listening mode after action
- Min support: Android 4.0; Max support Android 4.4

The Socket Camera is in listening mode once it starts. When a network request is received, the socket camera application triggers center focus of camera preview, followed by photographing, and closes the connection. It returns to listening mode once the connection is closed. Therefore, the Socket Camera is able to take multiple photos when multiple network requests are received.

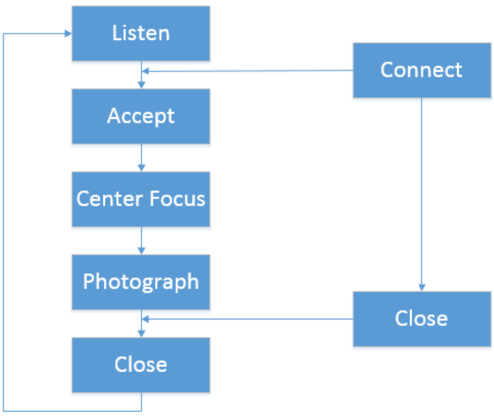


Figure 4.3.2.4.3 Communication between Socket Camera and Raspberry Pi

The application UI is simple. The upper part shows the network information and the lower part display the camera preview. A manual photographing button is added for debug purpose.

Figure 4.3.2.4.4 User Interface of the Android application

5. Manufacturing

5.1 Ancillary Components

We drew the design of ancillary components with CAD, and manufactured them with laser cutter. The material we used was Acrylic plate with thickness 2 mm &2.5mm & 4 mm. Among them, Board #1 ~ 4 were made with 2.5 mm Acrylic plate, and Board #5, #7, #16~18 were made with 2 mm Acrylic plate, and the remaining boards were made with 4 mm Acrylic plate.

5.2 Components Assembling

For on-head part, the assembling was stepwise:

- Filed the top of the helmet flat. The outer shell of the riding helmet was made with EPS (Expanded Polystyrene), so the filing was easy and fast. After filing, there is a plane with size about 120mm*120mm.
- Assembled the box that loads the cellphone. Assembled Board #1, 2, 3， and 4 with the 502 glue into the box. Fixed the Board #2 on the long-U-shaped frame of the servo with the 502 glue.
- Glued Board #5 and #6 together to form a connection part. Fixed Board #5 onto the servo using 3mm screws, and attached Board #6 on the flat top of the helmet with insulating tap. The tap goes through the holes on the helmet for several turns for a firm immobilization. Then, we used hot melt glue to fill in the gap between the helmet and board #6。
- Used the 502 glue to combine #7, 8, and 9 together. Used insulating tap to fix the mini-camera on the #8 board, facing our eyes. Fixed the whole part in the front face of the helmet by the hot melt glue.

For off-head part, we made a box with Acrylic board. There are three rooms of the box. The upper room is for the Raspberry Pi board; the middle room is for the ACE battery pack and the bottom room is for the Pisen Power Bank. We assembled the box by the following steps:

- Used the 502 glue to combine the #10~13 board together according to its corresponding position for the box.
- Installed the #14 supporting components into the #10 back frame and the #15 components into the #11 side frame.

- The #16 and #17 boards are movable. The #16 can be put into the corresponding supporting parts #14 and #15. The #17 is played as the front frame of the box and can be set into the #11 and #13 boards.

Then we put the Raspberry Pi board and two batteries into the appointed room. Then we used wires to connect the on-head and off-head. The wires are 200cm DuPont Line used to connect servo and servo power source, and 200cm high precision 15 pin Flexible Printed Circuits (FPC) line used to connect eye-tracking camera and Raspberry Pi. The FPCs were fastened by the #18.

5.3 Prototype

Figure 5.3.1 shows the prototype of our design.

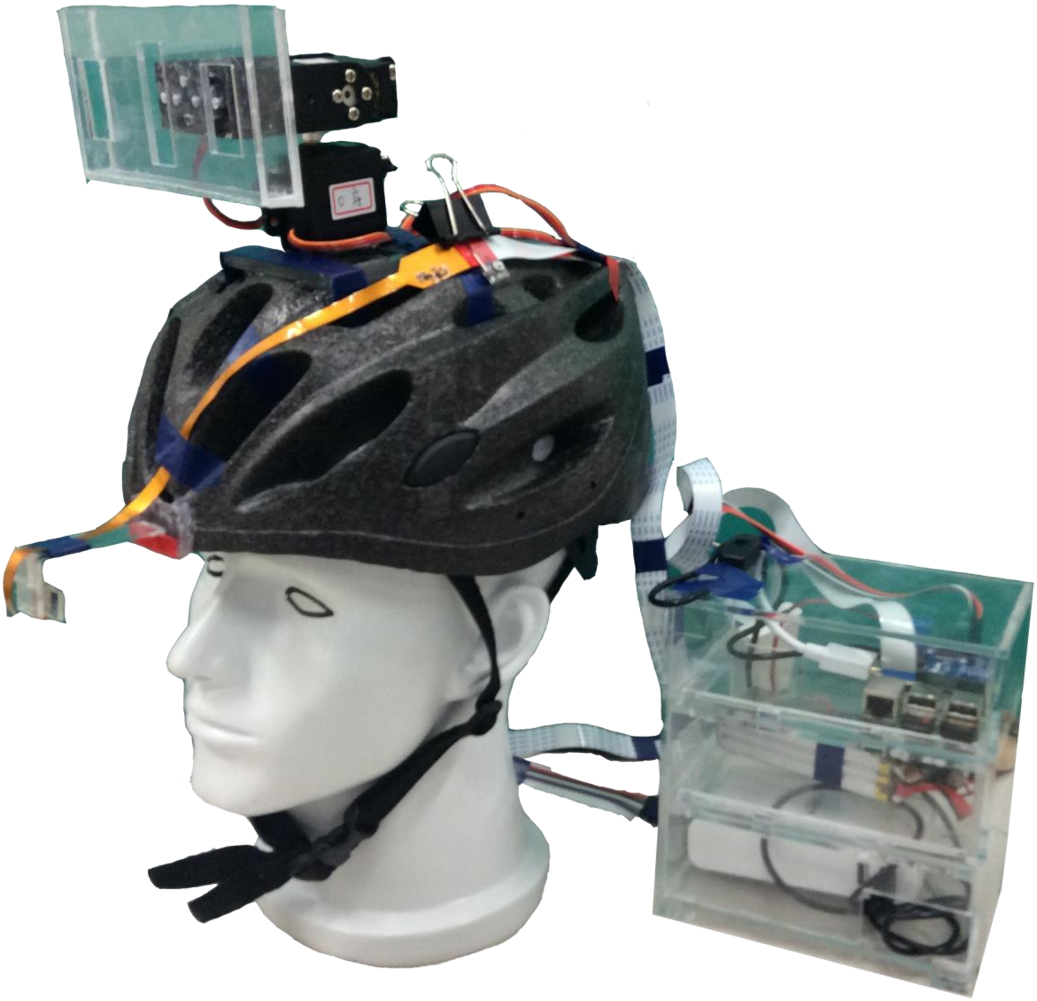


Figure 5.3.1 Prototype

5.4 Total Expenses

Here is a list of current expenses. It includes all the materials we have purchased.

Quantity	Part Description	Purchased From	Price (each) / RMB
1	MIUI 2S second hand	Online: taobao.com	737.6
2	1 meter extended wire for Raspberry PI camera	Online: taobao.com	23
1	11.1v to 5v-6v voltage converter for servo	Online: taobao.com	42
1	black servo frame	Online: taobao.com	6
1	silver servo frame	Online: taobao.com	6.5
1	OV7670 Raspberry PI camera module 0.3 Megapixels	Online: taobao.com	31
3	0.3/0.5/0.75 meter extended wire for Raspberry PI camera	Online: taobao.com	9
1	0.2 meter extended GPIO wire for Raspberry PI	Online: taobao.com	24
1	Acrylic board material & process cost	Online: taobao.com	259
150	m3 stainless steel screw	Online: taobao.com	0.06
50	m3 stainless steel nut	Online: taobao.com	0.06
50	m3 stainless steel spacer	Online: taobao.com	0.05
5	Deli 502 glue 8g	Online: taobao.com	2
5	Deli 502 glue 3g	Online: taobao.com	1.4
2	Deli AB rubber	Online: taobao.com	10
1	servo control extended board for Raspberry PI	Online: taobao.com	77
2	MG995 servo	Online: taobao.com	54
100	m3 nylon screw	Online: taobao.com	0.18
100	m3 nylon nut	Online: taobao.com	0.1
10	Hot melt glue stick	Online: taobao.com	0.54
1	TP-link WN725N 150M USB wireless network card	Online: taobao.com	45
1	59cm resin head model	Online: taobao.com	105
1	WoCase headband for GoPro 3+/3/2/1	Online: taobao.com	199
1	brushless Pan-Tilt for Aerial Photography	Online: taobao.com	332

1	OV5647 camera module 5 Megapixels	Online: taobao.com	144
1	Raspberry PI USB camera 3 Megapixels	Online: taobao.com	178
2	Raspberry PI card slot from TF to SD	Online: taobao.com	4.5
2	SanDisk TF card 8 Gigabyte	Online: taobao.com	46.3
1	servo driver board	Online: taobao.com	39
2	Raspberry Pi Model B+	Online: taobao.com	215.9
2	Raspberry Pi power line with switch	Online: taobao.com	13
1	HDMI to VGA convertor	Online: taobao.com	59
1	Raspberry Pi camera module with wire 5 Megapixels	Online: taobao.com	160
1	GUB SS riding helmet	Online: taobao.com	168
1	SanDisk SD card 32 Gigabyte	Online: taobao.com	86
1	VGA to HDMI convertor	Online: taobao.com	66
1	VGA to micro HDMI convertor	Online: taobao.com	69
1	Ralink RT5370 USB wireless network card	Online: taobao.com	40
1	EDUP EP-N8508GS USB wireless network card	Online: taobao.com	34
1	Powtoon software for education plans Student	Online: taobao.com	74
1	power line to USB wire for servo	Online: taobao.com	10
1	Pisen Power Twins 12000mAh power bank	Online: taobao.com	245
1	tool set (file)	Online: taobao.com	28
1	ACE 11.1v 2200mAh battery pack	Online: taobao.com	195
-	Total	-	4284.4

Table 5.4.1 List of current expenses

6. Validation

6.1 Weight

After we built up the prototype, we used the electronic balance in JI lab to measure the weight of our prototype. The part worn on the head is 646g < 1000g. The off-head part is 1318g. Hence, the total weight of our prototype is 646g+1318g=1964g < 2000g. Thus our prototype meets the specification of weight.

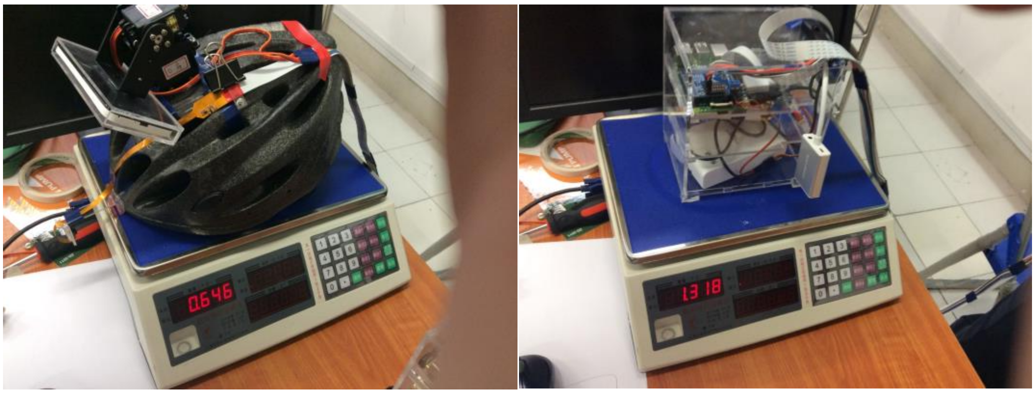


Figure 6.1.1 Weighing the on-head and off-head parts

6.2 Viewing Angle

After we built up the servo system, we test the maximum angle for it. For horizontal rotation, the viewing angle is 70.9 °+72.3 °=143.2 °> 120 °.

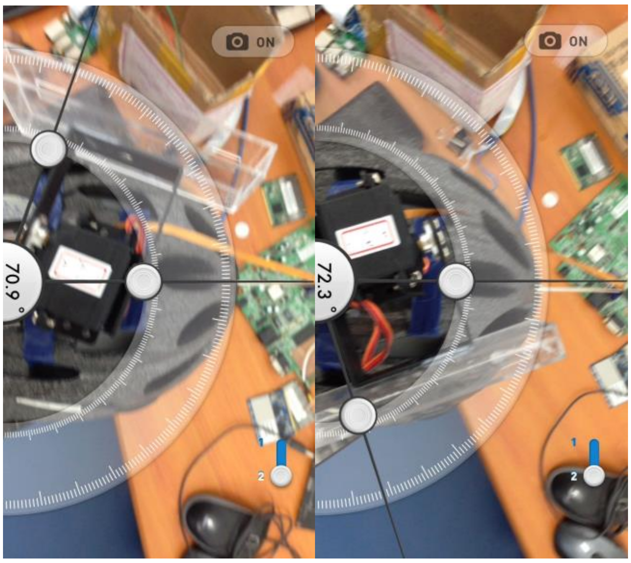


Figure 6.2.1. Measuring the horizontal viewing angle

For vertical rotation, the viewing angle is 37.7 °+83.0 °=120.7 °> 120 °.

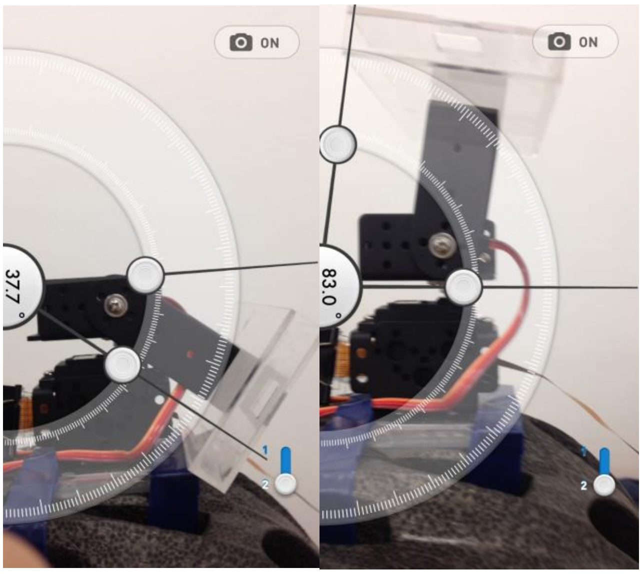


Figure 6.2.2. Measuring the horizontal viewing angle

Thus our prototype passed the viewing angle specification.

6.3 Robert Edge Detection Operator

This number represents the quality of the picture. It is basically a sum of the difference between adjacent pixels. A larger number means a larger difference, thus representing a better quality of the picture.

We tested pictures taken under static situation and the result is positive: The result 70.5055 is larger than 20. Thus for static photographing, we pass the test.

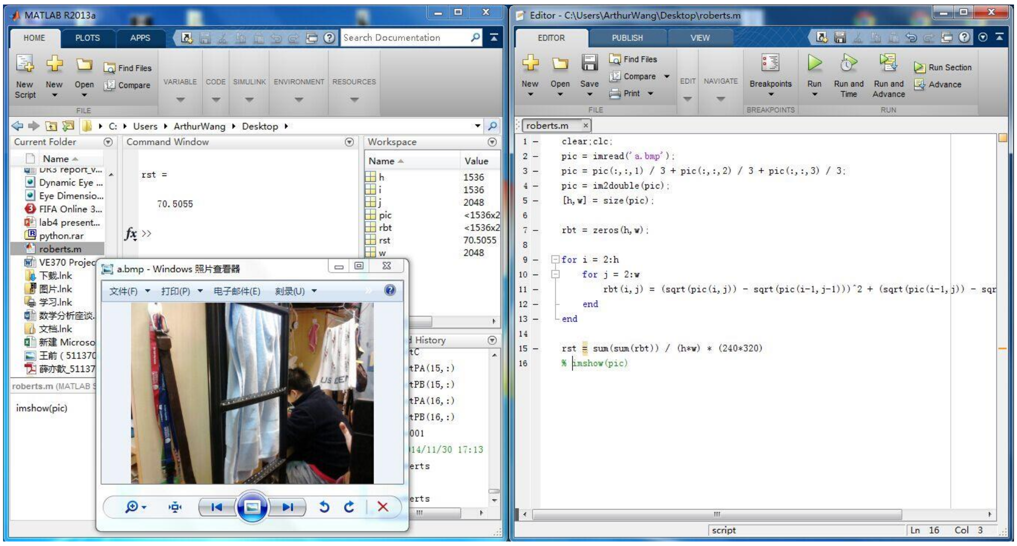


Figure 6.3.1 Calculating the Edge Detection Operator

6.4 Focusing Center Difference

We used result comparison to test the Focusing Center errors.

We let user focus on these points. By the tied laser on the servo, the direction of the servo, which is also the direction of the cellphone camera, can be shown clearly. We drew an array on the blackboard and tested for each point the deviation between aimed point and the laser point. The distance between the user and the array is 2.5 meters, while the average deviation is 20.54cm. Therefore, the average focusing center difference(measured by angle) is around 4.7 degrees < 5 degrees. So our focus is quite accurate.

Figure 6.4.1 Reference board used in Focusing Center Difference test.

6.5 Response time

Since Design Review #3, we exploited several optimization plans to minimize the response time:

- Use better sampling strategy

We use two threads. One takes eye motion pictures continuously while the other one calculates the eye motion results.

- Take eye-pictures with smaller range

Currently we take 640*480 pixels pictures of eye image, without a degradation of the picture resolution. Because the range of the eye-pictures are also shrinked.

Our prototype now detects eye motion every 0.5 second and it successfully meet our 1s response time specification.

6.6 Battery life

In Design Review #3 we only used one Pisen Powertwins 12000mAh USB Powerbank as our battery to provide both the central controller and the servos with power, and the battery life was only 3 hours. Now we add one more ACE 11.1v 2200mAh battery pack to provide power for servos separately, and the whole system can work more than 24 hours continuously.

7. Operation

The operation steps for users to use our prototype are as following:

- Install the App on the cellphone.
- Turn on the App.
- Wear the helmet.
- Calibrate and localize the eye with the guidance of the laser.
- Put the cellphone into the box attached on the U-shape frame of servos.
- Wait for 30 seconds and the system will start working.
- Gaze at some direction for more than 2.5 second and a picture will be taken.

8. Discussion

8.1 Strengths

- The eye-tracking algorithm is low-cost. It satisfies user's requirements in most cases with simple principles and short running time.
- The manufacturing process is simple. All components and machines are easy to acquire.
- The system is flexible. It permits user to use their own cellphones. The pictures taken are directly stored in the cellphone so that the customized apps can further process the information.

8.2 Weaknesses

- The stability of the helmet may influence the precision of the picture. It is impossible to fasten the helmet tightly enough, because that will make user uncomfortable. Therefore, during strenuous exercise, the position of the helmet as well as the eye-tracking camera will change a little bit. That may cause the eye-tracking malfunction and the system may take wrong pictures.
- The precision of the algorithm still needs improving, and this improvement may elongate the processing time, which directly leads to a longer response time.

8.3 Safety Issues

We have considered safety issues at the beginning of the design, and we believe our prototype is very safe for users based on following facts:

- The riding helmet is light and solid to give protection to our head.
- The relatively fragile plastic box is not carried on the head, including the battery which has very small probability to leak or explosive. They can be carried in a backpack.
- We embedded small pieces of metal shields between any protruding structures and the head, in order to avoid sudden impact and the puncturing of the helmet.

8.4 Environmental Impact

Our prototype is environmentally friendly based on the following facts:

- There is limited waste during manufacturing. The only waste comes from the Acrylic board scraps, which has been proved environmentally friendly.
- The only electromagnetic radiation comes from Wifi signal between cellphone and Raspberry Pi. The transmission power is 20dBm, much smaller than cellphone transmission power (50~70 dBm). So the radiation is less than putting one cellphone in a backpack, which is proved harmless to the environment as well as human body.

8.5 Economic Analysis

The direct cost of this prototype, excluding the components bought but not used in the end, is 2927.4 RMB in total. The detailed bill is listed in Appendix C.

For massive manufacturing, the cost should be lower:

- The Acrylic components can be produced in batches.

- The other components, although they are finished goods, can be acquired at a price much lower than retail price if in massive manufacturing.

We approximate the cost of each individual system, if manufactured in massive batches, should be less than 1000 RMB.

9. Recommendations

We have two kinds of recommendations for this project.

First is to further improve the prototype performance. We recommend to work on the eye-tracking algorithm to reach a more satisfying balance between accuracy and efficiency. Further researchers may also try to transplant our program onto other platforms rather than Raspberry Pi.

Second is to develop new functions based on this prototype. Since the pictures are stored in the cellphone, and our ultimate goal is to utilize this project in social networks, cloud computing and next generation communication systems, further researchers may develop new Apps for these usages. For example, they can try to use it to analyze customer's interests in markets and expositions. They may also try to develop some auto-matching and auto-recognizing algorithms to help the user get more real-time information.

10. Conclusions

This project, from proposal to final delivery, is a totally innovative piece of work. During the three month, we integrated communication and networks, embedded system design, algorithm design and implementation, and hardware manufacturing into this project. We also practiced our technical communication skills and gained much design experience. It is a mix of multi-area knowledge we learnt during our undergraduate education. Now our project has been successfully pushed into the last stage, and we are deeply convinced that a perfect demonstration will be presented on Design Expo.

11. Acknowledgement

We here sincerely acknowledge:

- The guidance from the sponsor faculty: Prof. Xudong Wang at UM-SJTU JI;
- The guidance from the instructor: Prof. Thomas Hamade at UM-SJTU JI;
- Technical advising from graduate student: Mr. Yuhang Zhang at UM-SJTU JI.

References

[1] Hong, A. K. A., Pelz, J., & Cockburn, J. (2012, November). Lightweight, low-cost, side-mounted mobile eye tracking system. In *Image Processing Workshop (WNYIPW), 2012 Western New York* (pp. 1-4). IEEE.

[2] Ando, T., Moshnyaga, V. G., & Hashimoto, K. (2012, March). A low-power FPGA implementation of eye tracking. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on* (pp. 1573-1576). IEEE.

[3] Mehrubeoglu, M., Pham, L. M., Le, H. T., Muddu, R., & Ryu, D. (2011, October). Real-time eye tracking using a smart camera. In *Applied Imagery Pattern Recognition Workshop (AIPR), 2011 IEEE* (pp. 1-7). IEEE.

[4] Rallapalli, S., Ganesan, A., Chintalapudi, K., Padmanabhan, V. N., & Qiu, L. (2014, September). Enabling physical analytics in retail stores using smart glasses. In *Proceedings of the 20th annual international conference on Mobile computing and networking* (pp. 115-126). ACM.

[5]Asundi, A., & Wensen, Z. (1998). Fast phase-unwrapping algorithm based on a gray-scale mask and flood fill. Applied optics, 37(23), 5416-5420.

[6]Zimmermann, H. (1980). OSI reference model--The ISO model of architecture for open systems interconnection. Communications, IEEE Transactions on, 28(4), 425-432.

Appendix A: Responses to Comments

This section is independently written to response the questions raised in DR #1, #2, and #3

A1. Responses to Comments in Design Review #1

1. Why not use a single camera to monitor the eyeball and shoot the view at the same time?

A: This is not likely to implement. Monitoring the eyeball requires a precise image, and the shot image should also be clear. Both of them require a proper focal length, respectively. But obviously the focal length adjustment cannot follow the fast switching frequency between eyeball image and environment image. If the two things are shot at the same time, then at least one of them will be sacrificed. The camera will definitely malfunction.

2. Why not use an advanced chip or development platform (such as in smart phones) instead of the large and heavy Raspberry Pi?

A: We are not skilled enough to directly control so advanced platform, which is a job in advanced engineering in top companies. Current smart phone obviously is not compatible with our hardware, while Raspberry Pi is an open, rich in resource, flexible platform. Raspberry Pi is actually not heavy. Each weighs around 50g. It is expected to meet our specifications.

3. A specification of focal length should be added.

A: This comment has been responded in Section 1.3.

A2. Responses to Comments in Design Review #2

1. How to demonstrate your prototype in Design Expo?

A: We will ask some user to try our prototype. We will ask the user to stare at whatever he/she wants, and check if the picture taken is his/her eyesight focus. We will also let the user to judge the satisfaction to the pictures.

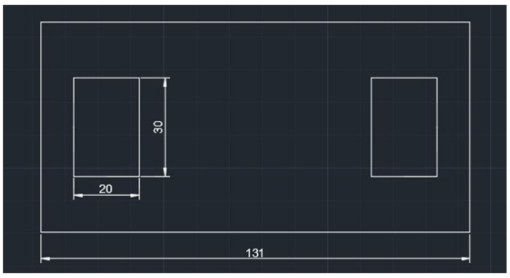
A3. Responses to Comments in Design Review #3

1. Your team should include more in algorithm part.

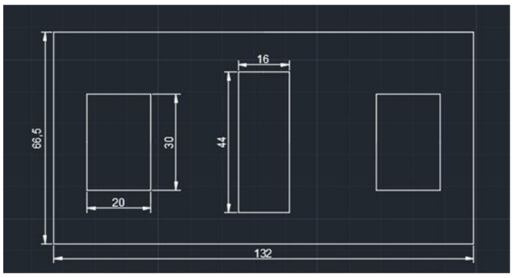
A: Yes. Because the presentation time was limited, and our prototype needs to be demonstrated, so we didn't have much time on software design and algorithms. We have included a nine-page elaborated description of software design. Please check Section 4.3.

Appendix B: Engineering Changes Notice (ECN)

WAS



IS



Notes:
Need to add a rectangle hole in the front face of the box for cellphone to facilitate taking out of the cellphone.

Ve450 Group 20	
Project: A Head Wearing Eye-Tracking Camera	
Ref Drawing: Cellphone box	
Engineer: Liyu Wang	12/10/2014
Proj. Mgr: Yanrong Li	12/11/2014

Appendix C: Bill of Material

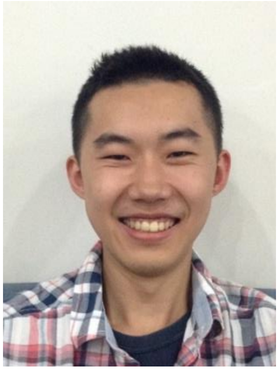
Quantity	Part Description	Purchased From	Price (each)
1	MIUI 2S second hand	Online: taobao.com	737.6
2	1 meter extended wire for Raspberry PI camera	Online: taobao.com	23
1	11.1v to 5v-6v voltage converter for servo	Online: taobao.com	42
1	black servo frame	Online: taobao.com	6
1	Acrylic board material & process cost	Online: taobao.com	259
150	m3 stainless steel screw	Online: taobao.com	0.06
50	m3 stainless steel nut	Online: taobao.com	0.06
50	m3 stainless steel spacer	Online: taobao.com	0.05
5	Deli 502 glue 8g	Online: taobao.com	2
5	Deli 502 glue 3g	Online: taobao.com	1.4
2	Deli AB rubber	Online: taobao.com	10
1	servo control extended board for Raspberry PI	Online: taobao.com	77
2	MG995 servo	Online: taobao.com	54
10	Hot melt glue stick	Online: taobao.com	0.54
1	59cm resin head model	Online: taobao.com	105
2	Raspberry PI card slot from TF to SD	Online: taobao.com	4.5
1	servo driver board	Online: taobao.com	39
1	Raspberry Pi Model B+	Online: taobao.com	215.9
2	Raspberry Pi power line with switch	Online: taobao.com	13
1	HDMI to VGA convertor	Online: taobao.com	59
1	Raspberry Pi camera module with wire 5 Megapixels	Online: taobao.com	160
1	GUB SS riding helmet	Online: taobao.com	168
1	SanDisk SD card 32 Gigabyte	Online: taobao.com	86
1	VGA to HDMI convertor	Online: taobao.com	66
1	VGA to micro HDMI convertor	Online: taobao.com	69
1	Ralink RT5370 USB wireless network card	Online: taobao.com	40

1	Powtoon software for education plans Student	Online: taobao.com	74
1	power line to USB wire for servo	Online: taobao.com	10
1	Pisen Power Twins 12000mAh power bank	Online: taobao.com	245
1	tool set (file)	Online: taobao.com	28
1	ACE 11.1v 2200mAh battery pack	Online: taobao.com	195
-	Total	-	2927.4

Appendix D: Biography

Appendix D: Biography

Yanrong Li (Phoenix)



I am a senior student in ECE at UM-SJTU Joint Institute. In the past three years, I cultivated strong leadership, communication skills, fast studying abilities and problem solving skills. I got the highest GPA in my major, awarded three times with National Scholarship, did several research projects, accomplished five TAs and won a series of international academic awards. I am interested in communications, networking, embedded systems and distributed computing. In the future, I plan to go to the US to pursuit a MS or PhD degree in computer engineering, and hunt a job in the top technology companies, or even start a business on my own.

In this project, I am the project manager. I assign work to team members and organize the team meeting. Currently I am also working on literature review and structure design. After this stage, I will devote myself into software developing.

Qian Wang (Arthur)

I am currently an undergraduate student of Shanghai Jiao Tong University, UM-SJTU Joint Institute, majoring in Electrical and Computer Engineering.

I am enthusiastic about computer science, especially the area of algorithm design and analysis. During my university life, I always emphasize the application of algorithms to the engineering domain and people's daily life. I also determine to further chase my computer science dream by pursuing a master or PhD degree in this area and applying my skills into the real world through the programs.

In this project, I am taking charge of the software design. More precisely, on the current stage, I focus mainly on the eye-tracking algorithm. After this stage, I will concentrate on the control of servo and camera.

Jiacheng Cai (Kenneth)

I am currently a senior student of Electrical and Computer Engineering at Shanghai Jiao Tong University. I also work as a Program Manager intern in Microsoft China now. Over the past few years, I've completed several business software projects in different teams to gain various types of experiences. I hope to become a software engineer when I graduate. I enjoy to design and develop computer applications. I am looking for a career in IT industry working with people to figure out new ideas and engineer good products.
In this project, I am responsible for software engineering. Currently I am focusing on embed systems development, eye tracking implementation and camera control. For the next stage, my duty may be changed to function testing.

Canchao Duan (Charlie)

I am a senior student in UM-SJTU Joint Institute and my major is Electrical and Computer Engineering.

I got addictive to programming when I was 12. Since then, I never wondered about my future career again. I believe that my gift leads me to become a programmer and I would enjoy my job a lot. I won prizes in programming contests, contributed to a web project of my institute information system, and this summer, I accomplished a great project as an intern in Microsoft. In future, I would like to go to a graduate school in the field of computer science and then work as a software engineer.

In this project, I am in charge of the hardware driver part and now I am working on the servo control system and the API calling issue of the sensor camera. Next step, I will help Arthur to implement his algorithm on our prototype.

Liyu Wang (Leon)

My name is Wang Liyu and I am now the senior student in UM-SJTU Joint Institute. I major in Electrical and Computer Engineering and I pursue a Computer Engineering job in international corporations in the future. What's more, I have plenty of things to do in my leisure time due to my rich hobbies, such as having parties, going for travelling, seeing movies and so on. And also, as you can see, I am an easygoing guy and very willing to make new friends.

In this project, I am responsible for the structure design. In more specific, a large structure combines all the components into a wearable system; small structures designs for controlling the focus length and position of the camera above the head. Furthermore, I am also the accountant for the team.