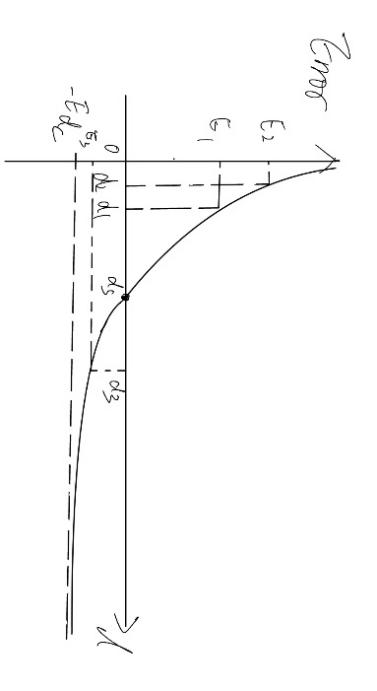
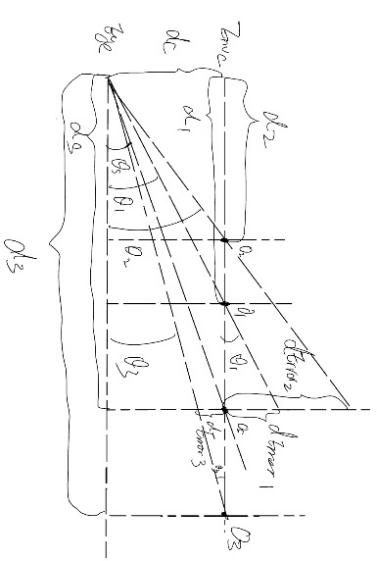
Aim: Mathematical modelling and analysis of the data from the eye tracker.

Week1:

Three data sets (En, Eye2, Eye3) are given and indexed by recording time. The systematic error that needs to be considered is the difference in distance between the environment camera and the eye.

* dc is the distance between the environment camera and the eye.
* ds is the distance between the eye and the calibrated object. (calibration distance)
* d1 is the distance between the eye and the object 1.
* d2 is the distance between the eye and the object 2.
* derror1 is the error for object 1.
* derror2 is the error for object 2.
* derror3 is the error for object 3.
* is the angle between the eye and the calibrated object.
* is the angle between the eye and the object 1.
* is the angle between the eye and the calibrated object.
* is the angle between the eye and the calibrated object.



*Figure 1.1 (left) shows how the vertical error appears and Figure 1.2 (right) shows the relationship between the error and the distance from the calibrated point.*

|  |  |  |
| --- | --- | --- |
| Since: | = | *E1.1* |
| Therefore: | derror1=-dc | *E1.2* |
| When: | ds=d1 |  |
|  | derror1 =0 |  |
| Since: | = | *E1.3* |
| Therefore: | derror3=-dc+ | *E1.4* |
| When: | d3 |  |
|  | derror3 =-dc |  |

*Table 1.1 shows the calculation of systematic error due to dc.*

According to the Figures and table, there is a positive shift in the vertical direction when the testing object is in front of the calibrated object and a negative shift in the vertical direction when the testing object is behind the calibrated object.

The mathematical optimization technique I used in this scenario is the least squares method. Since the data sets are indexed by recording time, 80 percent of five independent variables and one dependent variable which matched each other are used to find the best-fit function. The rest part of data is used for testing the error of the best-fit function.

Yen =1342 -1.2 Xeye\_2 + 0.1\* Yeye\_2 +90.8 \* (log(d) + 10)) \* exp(-10 \* d) ^ (-2)) -0.9\* Xeye\_3 -0.1\*Yeye\_3;

Rate of error of Yen: 3.8%

Xen=85.8930-3.133\*Xeye\_2+2.4082\*Yeye\_2-113.6351\*exp(-1000.\*(d).^-2))+ 3.5896\*Xeye\_3 + 1.2287.\*Yeye\_3;

Rate of error of Xen:3.9%

Week2:

According to the graphs provided by Mingnan, the data should be partitioned through the location of eye position. This method can be used to find the maximum error, since some areas will only be used to test the function’s error.

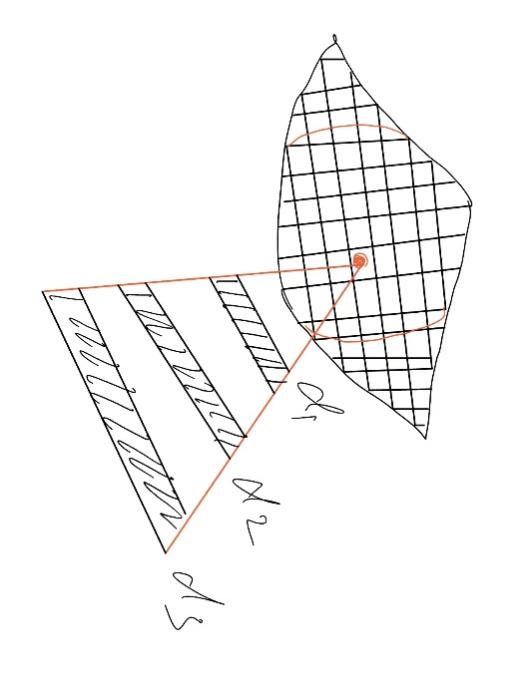


Figure2.1 shows the grid that divides the eye into areas

The size of the grid depends on the number of data collected. The area of each grid will be large if the amount of data is too small to fit a function since some data is missing on specific eye positions.

Since some data is also missing in specific depths, the mean environment position is calculated for each depth within each grid and the mean position of an eye within each grid is also calculated through the mean value to ensure that more data can be used.

In this Matlab code, eye position：

{[4,3], [5,3], [6,3],[7,3],[8,3],[9,3],[2,9],[3,9],[4,9],[5,9],[6,9],[7,9]}

are excluded from fitting the function and can only be used for testing. This simulates the results of the real test.

Without changing the function from week 1, the vertical error term is added to the function of Yen. The error is around 13 degree.

Since the calibration distance varied when the testing distance changed, the function needs to include which one or two calibration distances should be used for a new testing distance. A weight factor might need to be added to different calibration distances.

There is a way to complement error through the relationship between the vergence angle of the left eye and the vergence angle of the right eye. This can be used to collect in-depth information,