Vision Correcting Display - Master of Engineering Capstone Final Report

Sijia Teng
Jia Zeng
Vivek Claver

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2017-96
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-96.html

May 12, 2017

Acknowledgement

The author would like to thank Professor Brian A. Barsky, Dr. David Grosof, Leah Tom, and all other researchers in Vision Correcting Display Group for many kinds of assistance and support.

University of California, Berkeley College of Engineering

MASTER OF ENGINEERING - SPRING 2017

Department of Electrical Engineering and Computer Science (EECS)

Visual Computing and Computer Graphics (VCCG)

VISION CORRECTING DISPLAY

SIJIA TENG

This **Masters Project Paper** fulfills the Master of Engineering degree requirement.

Approved by:

1. Capstone Project Advisor:

Signature: _____ Date 8 May 2017

Print Name/Department: BRIAN A. BARSKY, Professor / EECS

2. Faculty Committee Member #2:

Signature: _____ Date 5/8/2017

Print Name/Department: DAVID H. GROSOF, Ph.D. / Unaffiliated non-faculty

Abstract

This is the report of the the M.Eng. Capstone project - Vision Correcting Display. Vision Correcting Display aims to enable people with eye aberrations to see a sharp and clear image on the screen without using corrective eyewears. The Vision Correcting Display research group, led by Prof. Brian Barsky, includes both masters and undergraduate students. This report mainly focuses on the works done by the author, Scarlett (Sijia) Teng, and Sophie (Jia) Zeng, and Vivek Claver. In Chapter One, individual technical contribution of the author is introduced; Chapter Two shows how the author and the group members apply engineering leadership methods to the project.

Vision Correcting Display

Master of Engineering Capstone Final Report

Scarlett (Sijia) Teng

with Sophie (Jia) Zeng and Vivek Claver

UC Berkeley

Author Note

Sijia Teng, Electronic Engineering and Computer Science Department, with concentration of Visual Computing and Computer Graphics, UC Berkeley.

The author is currently a master's student advised by Prof. Brian A. Barsky and conducting research in his vision-correcting display project.

Correspondence concerning this article should be addressed to Scarlett Teng, Department of EECS, 510 Soda, UC Berkeley, CA 94720. Contact sijia_teng@berkeley.edu.

Acknowledgements

Professor Brian A. Barsky, Dr. David Grosof, Leah Tom, and all other researchers in Vision Correcting Display Group for many kinds of assistance and support.

Executive Summary

This is the report of the the M.Eng. Capstone project - Vision Correcting Display. Vision Correcting Display aims to enable people with eye aberrations to see a sharp and clear image on the screen without using corrective eyewears. The Vision Correcting Display research group, led by Prof. Brian Barsky, includes both masters and undergraduate students. This report mainly focuses on the works done by the author, Scarlett (Sijia) Teng, and Sophie (Jia) Zeng, and Vivek Claver. In Chapter One, individual technical contribution of the author is introduced; Chapter Two shows how the author and the group members apply engineering leadership methods to the project.

Chapter 1 Technical Contribution

1. Introduction

The goal of the whole project group is to develop vision correcting displays which enable people with visual aberrations to see a clear image on it without corrective eyewears. Our group already had a working prototype in previous works (Huang & Barsky, 2014), and the remaining works focus on improving the display's performance and solve problems in practical use, including both software processing and hardware implementation.

In software processing, on the one hand, our group members are improving the image processing algorithm to speed up and improve the contrast and resolution. On the other hand, we also rearrange the image for people who have different eye aberrations in their two eyes, which is called the binocular vision problem. To solve this problem, we need to combine two images into one, which is elaborated in this paper.

In hardware implementation, we use two masks to cover the screen. The first one is a pinhole mask used to filter the light rays emitting from the screen, which ensures the one to one ray tracing we use for the optics geometry calculations that enable aberration correction. Our group members are aiming ultimately to replace the pinhole mask with a lenslet array, which has similar light-field shaping functions, in order to increase brightness. The second mask is a parallax barrier that is used to aid the separation of the display, so that one half-image goes to each eye.

The work breakdown in our capstone group can be described as Fig. 1.

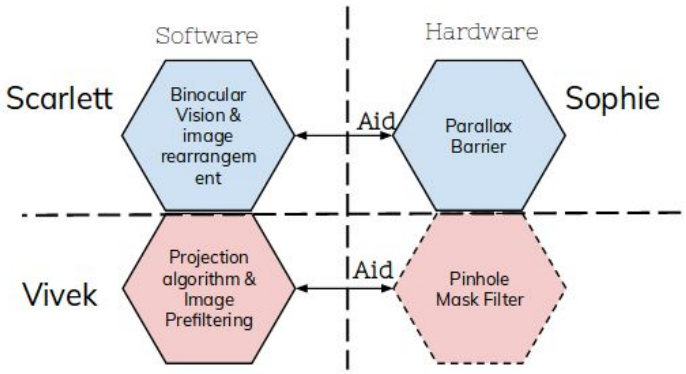


Fig. 1 Work breakdown illustration (jointly with Sophie).

My work during this year aims to 1) solve the binocular vision problem and 2) do observer-to-display (eye-screen) distance detection. The binocular vision problem is introduced in section 2, and the solution we propose is applying a parallax barrier model. Applying this model involves development of both image processing and hardware. In this paper, I focus on the image processing part, and my partner, Sophie, focuses on the physical setup. The distance detection work aims to enable the automatic adjustment function of the parallax barrier in future devices, and is introduced in section 3.

2. Solution to the Binocular Vision Problem

2.1 The binocular vision problem

Binocular vision problem in this context refers to people with different visual aberrations in their two eyes. However, the previous work showed how to correct only one eye's aberration: we suppose the user will perceive a clear image on the retina, we do ray tracing from the retina, go through the lens and cornea which are the main source of aberration, and back to the display to compute an altered image as the back-traced result; because light rays are invariant under

inversed paths, after filtering the light rays emitting from the display by the pinhole array, user will perceive the clear image in the retina. As can be seen, the path of the ray we trace depends heavily on the aberration of the lens and therefore so does the computed image displayed on the screen. If people have different aberrations in their two eyes (as is common), there will be two different tracing results and different processed images, but we only have one screen. So, the problem is how we can compute two half-images corresponding to the left eye and right eye, display on a single screen, and route each computed half-image only to the correct eye.

2.2 Approach to solve the binocular vision problem

The method we use to solve the binocular vision problem is similar to the auto-stereoscopy technology (Dodgson, 2005), which we called the parallax barrier model. The auto-stereoscope also aims to combine two different images corresponding to the left and right eyes, and also frees people from headgears or glasses by applying a physical setup called the parallax barrier onto the screen.

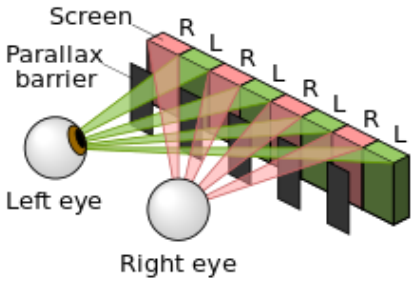


Fig. 2 Illustration of the parallax barrier (Parallax Barrier, Wikipedia, 2017).

Specifically, this method does image interleaving first, which means vertically cutting the images corresponding to the left and right eyes stripe by stripe and combining them together. Then it applies a physical mask called the parallax barrier in front of the interleaved image. As is

shown in Fig. 2, the black stripes of the barrier block the left eye image from the right eye, and the right eye image from the left eye, and the transparent stripes allow the eye sees the corresponding images (which are the same extent horizontally and vertically as the monocularly presented image in previous work of Huang and Barsky, but which are defined in the binocular case by each eye's half-image made up of alternating vertical strips).

Problems we observed based on previous work by Chen and Tom are: first, we interleaving the two images together so each of them only contains half of the original image information; second, we use a barrier to block the image so users can perceive black stripes.

To solve the problems, we referred to an article of auto-stereoscope (Perlin et. al, 2000) that uses a multi-time-complexing barrier. In their work, instead of using a static barrier, they use a transparent display to implement a fast switching barrier. In phase one, the odd stripes display the left eye image, and the even stripes display the right eye image; in phase two, the odd stripes display the right one, the even stripe display the left one, and the black barriers switch its transparent positions. In this way, the left eye will perceive the odd stripe image at phase 1 and the even stripe at phase 2, so there will be no information loss for either eye's image. If the switching speed is high enough, the user will not notice the black stripes of the barrier either. Thus, the problems brought by the static barrier are solved.

However, the time-multiplexing proposal was infeasible. According to our simulation, we found out that no common LCD screen can afford this high switching rate, which is 180 Hz (Dodgson, 2005). Only special screen like the "pi cell" liquid crystal screen (Liquid Crystal Technologies, 2008) or the Ferroelectric liquid crystal screen (FLC) (Lagerwall & Clark, 2004) can achieve a sufficiently high switching rate. This method requires a totally different device for

our display, a dynamic shutter with high spatial resolution, instead of what we do, adding a simple static masks to a common electronic display.

I find a solution by looking back to earlier work. Previous work in this lab had only simulated for one eye's input image. But since the left and right eyes' perceived images are exactly complementary when combined in the brain for the binocular display we are aiming to design, a whole image with no information loss is possible to achieved. In addition, if we use a parallax barrier with black stripes thin enough, it will also be less difficult for the mind's binocular perceptual apparatus to fuse and combine. With this interest in rendering the binocular view through fine vertical 'fencing' to each eye, we focused on the simpler static barrier model.

2.3 Build the parallax barrier model

2.3.1 Parameter calculation and developing procedure

To calculate the parameters of the parallax barrier, we use the geometry relationship as illustrated in Fig. 3, where E is the distance between the two eyes, D is the distance between the eyes and the screen, and G is the distance between the parallax barrier and the screen. Find two similar triangles and apply the similar rate function, we get (see Appendix A for the detailed derivation of the formulas):

$$\text{barrier width } B = (G \times E)/D, \tag{1}$$

$$\text{and image width } I = (G \times E)/(D - G). \tag{2}$$

We can see that neither B nor I is depended on the viewing angle of the eye and the screen. While the pinhole mask requires a particular viewing direction, the parallax barrier don't.

This shows applying the parallax barrier to the pinhole mask on the display will not further constrain the viewing area.

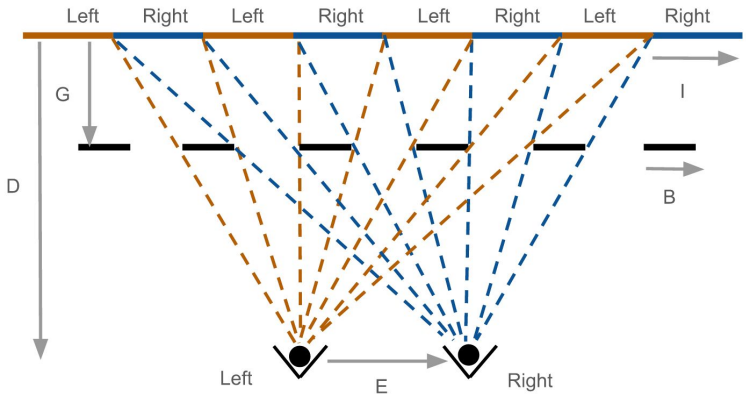


Fig. 3 Parameter calculation function illustration.

The procedure of developing a parallax barrier model is: first, we get the E value from user as a constant; second, we choose some pairs of B and G parameters to be used in building the physical setup; third, we calculate the D value to test the feasibility; then, use the D, G and E value to calculate I to be used in image interleaving process. After having some sets of parameters, the work can be divided into two parts: one is building the barrier physical setup which Sophie elaborates in her paper; the other is interleaving image which is discussed in section 2.3.3.

2.3.2 Experiment setup and process

The previous works have many software simulations, so what we do is to develop the prototype and test with real person. Our observers were the author and Sophie, visually normal myopic young adults, corrected to normal by contact lenses and eyeglasses, with no history of amblyopia, head trauma, stroke, or other serious disease of the central nervous system. In our

experiment, if the user can view an image without information loss, and only view the left (or right) image when closing the other eye, our prototype can be proved to work well.

To set up the experiment, we use an image with different size of text to do the experiment. The other thing we need to do is to mark the left and right eye images. One way is to use different color, but our brain will get confused when having two colors as input. So, instead, we use a line at top to denote the right eye image as shown in Fig. 4, and after combination, people can only see the line on the top when open the right eye.

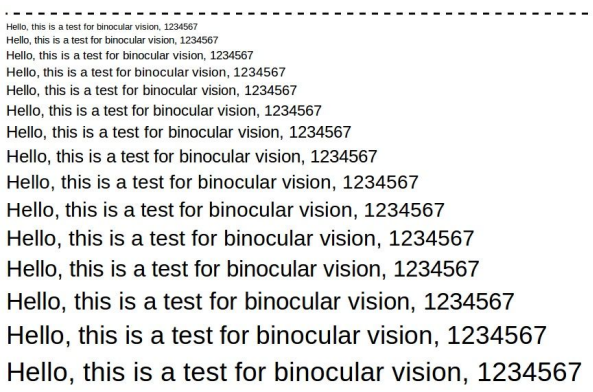


Fig. 4 Image used in experiment.

One problem we encountered was the image shifting phenomenon. In Fig. 5, we show the top of the simulated right eye perceived image. Ideally, we can only see the black bars, but now we can see the parts for right eye which do not have the black line shift out to be visible from period to period.

We have two hypotheses on the cause: one is the viewing angle issue, meaning eyes have different angle in viewing different part of the screen; the other is the accumulated error. As is discussed in Sophie's paper, we rule out this hypothesis. So, the only reason should be the accumulated error: if the image stripe is a little longer than the calculation result, it will

accumulated and the stripe will be shifted and cannot match corresponding the barrier. In section 2.3.3 we will introduce how we solve the problem.

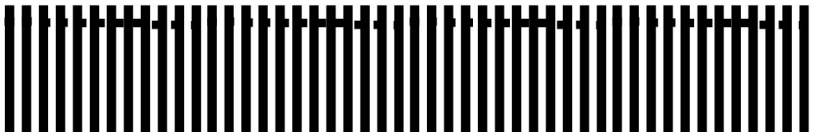


Fig. 5 Illustration of the image shifting phenomenon.

2.3.3 Image interleaving process

Given the left and right eye images, and the width of each stripe in the final image, what the image interleaving does is to cut the image and then combine.

The difficulty arises from the relationship between the discrete pixel size in relation to the barrier width (and the separation of the eyes). What we use in parameter calculation in section 2.3.1 is in unit of inch, but images computed for aberration correction can be rendered only in discrete pixels.

The first step is to get the PPI. PPI means pixel per inch, using which we can convert the inch unit width I into pixel unit. Our program gives the user several choice of setting or calculating the PPI: 1. get PPI value from direct input; 2. calculate PPI by width or height; 3. calculate PPI by the display model's diagonal length.

If the user knows the PPI of the display, he or she can directly use the value as an input to our program. If not, the user can feed a measured horizontal and vertical length of the screen, and the current resolution of the display. The formula to calculate the PPI is:

$$PPI = R_h/h \text{ , if view from the horizontal direction,}$$

$$\text{or } PPI = R_w/w \text{ , if view from the vertical direction.} \quad (3)$$

R_h is the horizontal resolution, and R_v is the vertical resolution of the display. w is the width, and h is the height of the screen.

If the user do not know and do not want to manually measure and do the input, we also offer a third choice where the model ID of the device is obtained and so as the manufacture screen size. Usually the screen size s is given by the diagonal length, so w and h in Eq. (3) can be obtained by:

$$w = s \cdot R_h / \sqrt{R_h^2 + R_v^2}, \tag{4}$$

$$\text{and } h = s \cdot R_v / \sqrt{R_h^2 + R_v^2}. \tag{5}$$

Using PPI, we convert the I in inch unit into pixel unit I_{px}. The next step is to interleave the image using I_{px}. As is described, the problem is that I_{px} can not be ensured to be an integer. But when manipulating the image, the smallest unit is one pixel.

The first method we used is rounding to nearest. In this way, 3.2 pixels would round to 3 pixels and 3.7 pixels would round to 4 pixels. However, this would cause the accumulated error as is shown in section 2.3.2.

The second method that can reduce the accumulated error is using compensation. For example, if $I_{px} = 3.2px$, I would use 4 stripes in width of 3 pixels, and then 1 stripe in width of 4 pixels, and 4 stripes in 3 pixels again… In this way, although each stripe has an error in either direction of up to one pixel, the error will be compensated by other stripes and the error would not be accumulated: it would be controlled to be small, if spatially distributed quite widely. The

implementation of this method, comparing with the first one, is shown in Fig. 6. The principle of this is to remain the floating point until it has to be round to the nearest integer.

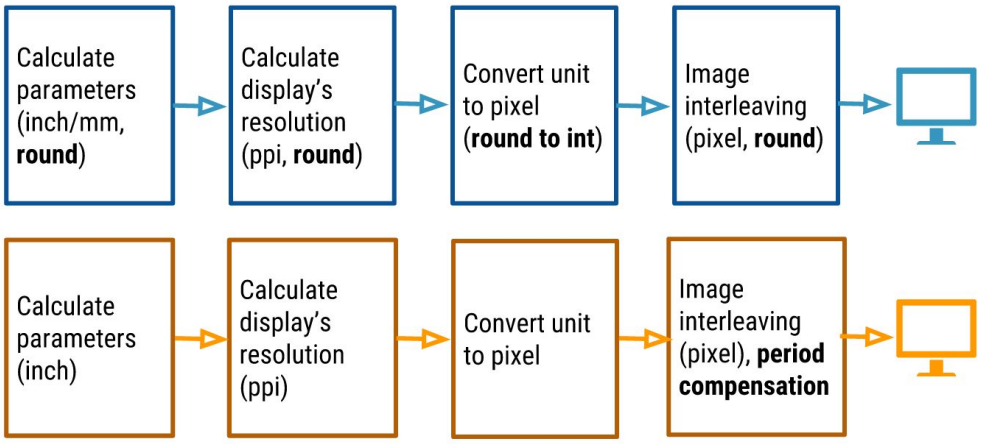
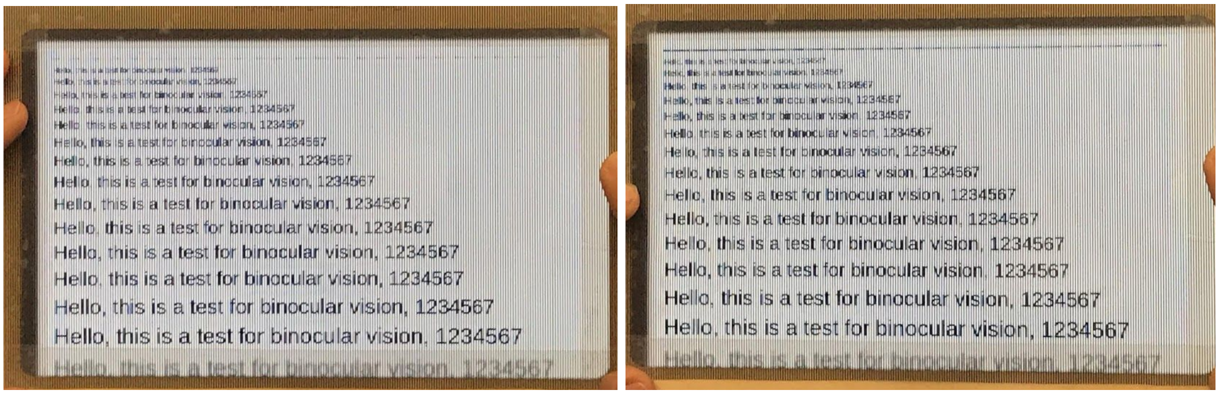


Fig. 6 Implementation flow chart. Top: method 1; bottom: method 2 (jointly with Sophie).

2.4 Results of the parallax barrier model

After image interleaving and printing out our parallax barrier prototype, we did in-person experiment. The parameters of the prototype are: $B = 1/64\ inch$, and $G = 1/4\ inch$. The design was for users to view at a distance of 20 inch. The result as viewed by our observers is what we expected: the left eye can see only the left eye image which does not have the black line, and the right eye can see only the image with the black line on the top, and user can view the text at bottom fifth line, as is shown in Fig. 7.



(a) (b)

Fig. 7 In-person test: viewing result of the parallax barrier model. (a) Left eye; (b) right eye. Photo was taken by embedded iPhone7 camera at 20 inches.

This proves our prototype can work well. But in the left eye image we can still notice some black dots on the top. But each of the visible black dot is less than one pixel in width. This is because that there are some errors in each stripe because we have to convert the non-integer value into integer pixel value. Also because the prototype is handmade and we cannot ensure that it is flat enough. We can use industry manufacturer to generate better result.

3. Implementation of Eye-Screen Distance Detection

3.1 Background of distance detection

The aim of doing automatic measurement of viewer distance is to enable rapid, dynamic adjustment of the barrier width B and thereby to make the parallax barrier maintain image segregation for a range of viewer distances to the screen.

The parallax barrier model introduced in section 2 is implemented using a printed transparent sheet with a frame, and the size cannot be changed after being manufactured. Regarding the Eq. (1) and (2), both the barrier width B and the image width I need to be changed to accommodate different eye-screen distances D. In practical use, we will have a monochrome transparent LCD that enables to display the width changeable barrier. Unlike to the high switching rate displays we discussed in section 2.2, we only need common transparent displays. Given a transparent LCD responsible for this in the future, what we need to implement now is calculating the barrier width we want to display on the LCD.

The eye-camera distance calculation is based on the pixel distance of the two pupils of the face image captured by the camera (as if the axis defined by the line between the eyes is in the fronto-parallel plane with respect to the camera; that is, we assume rotations of the head that move that line out of the plane are negligible). So, the eye-screen distance is actually eye-camera distance. When the face is close to the camera, the pixel pupil distance is large, vice versa. We can record some pupil distance D_{pupil0} and eye-camera $D_{eye-camera0}$ distance pairs as reference and calculated the new eye-camera distance $D_{eye-camera}$ by the a newly captured pupil distance D_{pupil} (see detailed derivation in Appendix B):

$$D_{eye-camera} = D_{eye-camera0} \times D_{pupil0} / D_{pupil}, \quad (6)$$

This method has an assumption that the pupil distance in real world (the term “real world” is used to differ from the pupil distance in image plane, which has the unit of pixel and is measured by the camera image) stays unchanged, but it only holds for cases when the eyes focus to infinity. Reading the display keeps our eyes focusing on a close object, where the vergence

movement of the eye makes the pupil distance in real world varies. The closer the eyes focus on, the smaller the eye-center distance is. So this makes our formula underestimate the real eye-camera distance when the eye position is closer to the screen than the referenced position, and overestimate it when farther (see Appendix B for detailed derivation). The implementation of this algorithm is illustrated in Fig. 8.

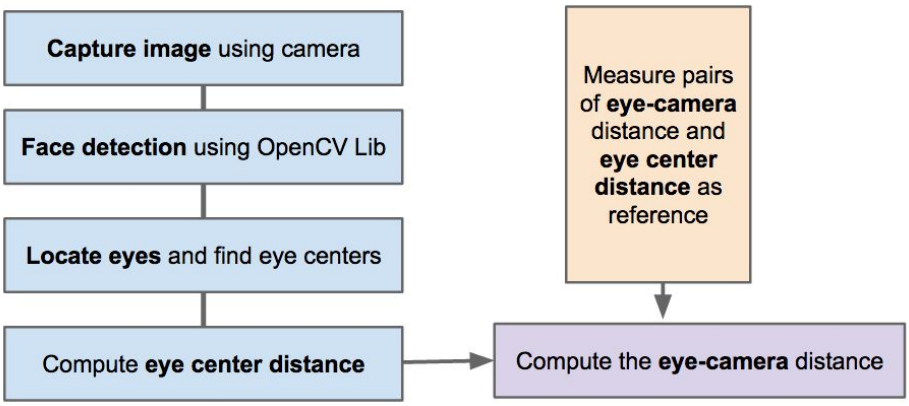


Fig. 8 Illustration of the eye-camera distance detection algorithm (jointly with Sophie).

A key requirement of this is tracking the eyes of the user. Our former researcher, Wenjing Ke, has implemented a real-time eye tracking program which marks the pupil centers' positions in the image sequences of the face as captured by the (ordinary, visible light, webcam-quality) camera.

However, the result is not good enough because the eye-detecting program has a high false detection rate, so we need to make the eye detection more stable and accurate, especially in the cases of using the vision correcting display. In section 3.2, I compare some eye detection algorithms and analyze our chosen method. In section 3.3, I show the implementation and the results of setting the size of the bounding box.

3.2 The eye detection algorithms

There are generally two kinds of eye detection algorithm: one is directly detecting the eyes, the other is first detecting the face and then localize the eyes' regions based on a natural proportional parameter of the face, which we regard as an "indirect" method.

The direct detection method can be implemented using OpenCV's Haar Cascade eye detection model. But it can falsely detect the nose or other darker corners on the image. Another algorithm (Sinhal, 2017) improves the old OpenCV's framework by adding the frequent false detection as a negative training set to the original positive dataset of the eyes. It has better accuracy than the former one.

Indirect method is more stable in eye detection on the whole. Ke [provide last name] used the indirect method and a gradient algorithm to detect the center of the pupils after detecting the face and get the region of the eye (Barth & Timm, 2011). This algorithm maximizes the gradient inner product value to find the center. Another indirect method we found use a Snakuscule method to find the pupil center (Garg, Tripathi & Cutrell, 2016). This method maximizes the energy difference between an inner circle and an outer ring to find the outline of the pupil. This algorithm is good for real time detection, and can handle eyeglasses, shadows, poor contrast, thick eyelashes, slight variations in pose and scale, but does not work well if the pupil is not completely visible (Garg, Tripathi & Cutrell, 2016). Neither of the algorithm would detect the eye without successfully finding the face.

In the context of the vision correcting display, the camera can usually detect the whole face (can use indirect method), the users do not need to wear glasses, and when looking at the

screen, the eyelid may cover parts of the eyes (when Snakuscule method does not work well). For these reasons, we decide to continue with Ke's program.

3.3 Eye center detection improvements by tighter bounding box method

The method we use to improve Ke's program is using a smaller bounding box to localize the eyes. We notice that Ke's program works well in the default condition where face is vertical (so the line connecting the two eyes is horizontal) and the camera's view of the face clear of hair and other obstructions; however, it has more false detections when the head is tilted out of vertical. In Fig. 9, we use similar terms in plane motion description to denote different types of head movement (tilt back and forth as pitch; tilt left and right as roll; shake left and right as yaw). Since in our project, we only need to find the center of the pupil, so we can tighten (reduce) the eye bounding box to reduce interference from other dark noise.

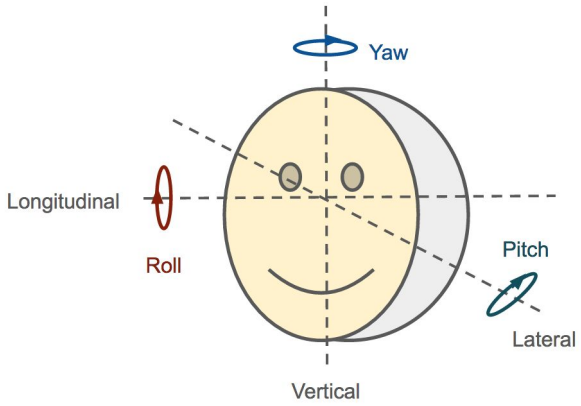


Fig. 9 Terms of head motions.

We test the results by running two programs with large and small bounding box on the author (of which the eye and health conditions are described in section 2.3.3) using the same camera image. The best bounding box we use has the parameters: $P_t = 29H_f$, $P_s = 22W_f$,

$H_e = 20H_f$, and $W_e = 20W_f$. Where $W_f \times H_f$ is the face bounding box of the face detection result. Fig. 10 illustrates the parameters.

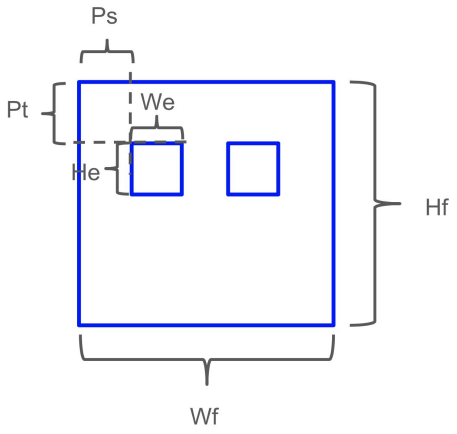
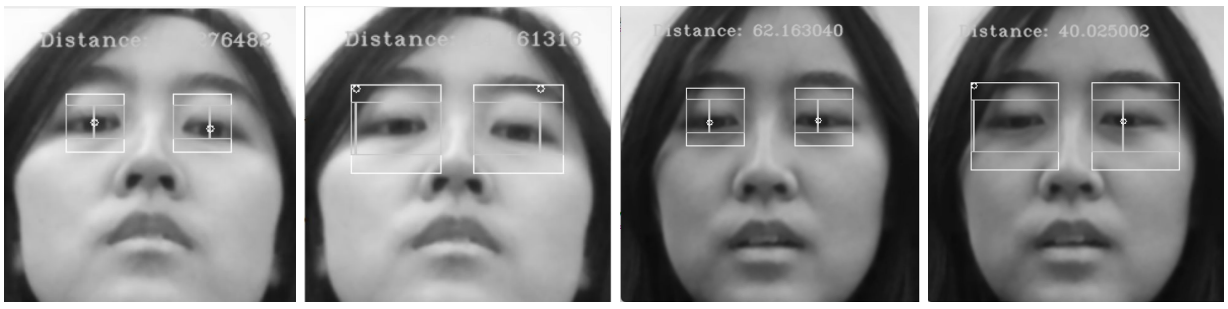
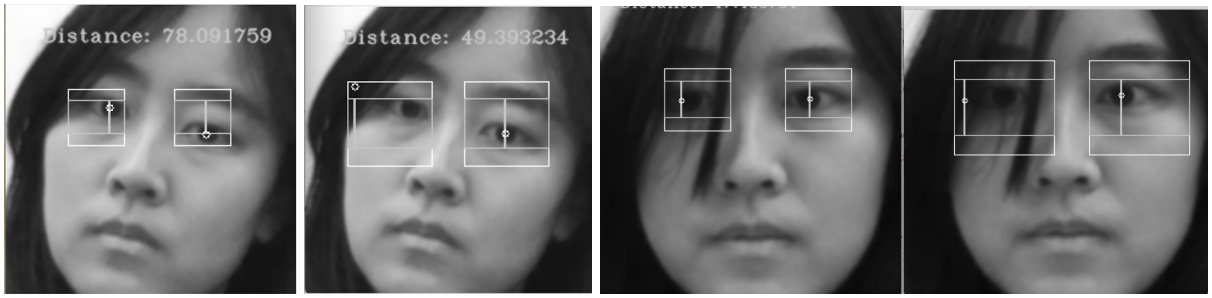


Fig. 10 Illustration of the eye bounding box parameters.

The smaller bounding box we use proves to have the same high true detection, and has less false detection in the tilted cases. Fig. 11 shows the results of comparing the small bounding box with the original large bounding box which has the parameters: $P_t = 26H_f$, $P_s = 15W_f$, $H_e = 30H_f$, and $W_e = 30W_f$. They both work well with normal face positions, but the smaller one works better on most of the uncommon cases. One thing to notice is that, since the face detection does not work for a highly rolling face, and the eye region localization also does not work well. Fig. 11 (c) is the maximum rolling angle of the face, which is around 10 degrees.



(a) (b)



(c) (d)

Fig. 11 Eye center detection results comparing the small and the large eye bounding box for the tilted cases: (a) pitch (tilt back), (b) squeeze eyes, (c) roll (tilt right), and (d) hair blocked.

3.4 Results after combine with distance detection

The improved bounding box makes the eye detection program more stable, which we can use to do the eye-camera distance detection. Comparing with physical measurements of the distance of the eyes to the camera, we get an average 0.49% relative error for a normal face in distance from 30cm to 90cm. For the tilted examples, we get an average 0.28% relative error for tilting back, 0.29% relative error for tilting right, and 0.43% relative error for translation (see specific table of results in Sophie's report). This proves our eye-camera distance detection method works well.

In the above tilted cases, we emphasize that we do not design for the circumstance of head rotation around the vertical axis of the head, which causes the two eye-camera distances to be different for the left and right eyes, because the parallax barrier requires the face to be parallel to the screen. We assume users should not yaw their heads to produce rotation around the vertical axis of the head.

4. Future Work

Future works include: 1) make the distance detection more stable to false detection, i.e., identify false detection by sudden change in distance results and ignore the change; 2) combine the parallax barrier model with the pinhole mask or the lenslet array mask, and deal with the Moire effect; 3) implement using transparent LCD; 4) deal with the estimate error of the eye-camera distance detection.

Chapter 2 Engineering Leadership

1. Introduction

The second chapter of the report is to focus on the engineering leadership abilities we adapted throughout the project and to provide context of the project. We have identified three key dimensions of the engineering analysis: our project management strategy, the social context of our research, and industry analysis of the potential market for our research.

2. Project Management

Our long term research project is currently at a research stage. Our group is led by Prof. Brian Barsky of the Visual Computing Lab, and consists of Prof. Barsky, vision science professional David Grosof, graduate and undergraduate students from UC Berkeley. There are also other student researchers working remotely. Our group also has close collaborative ties with the MIT Media Lab.

Our current research address two major problems based on previous researches. The first problem is how to improve the performance of the display. This involves the optimization of the algorithms used to calculate the way to project the image, which can be divided into the two tasks: first, establishing the projection relationship between the display and the eye; second, the prefiltering of the image once this relationship is established. It also involves the design of better hardware to project the prefiltered image.

The second problem is the binocular vision problem, which determines how to accommodate people with different aberrations in their left and right eyes. Both problems

involve a combination of software and hardware considerations. The division of the workload among us three MEng students is the following: Sophie and Scarlett are working on the binocular vision problem together, while Vivek is working on the optimization of the display algorithms.

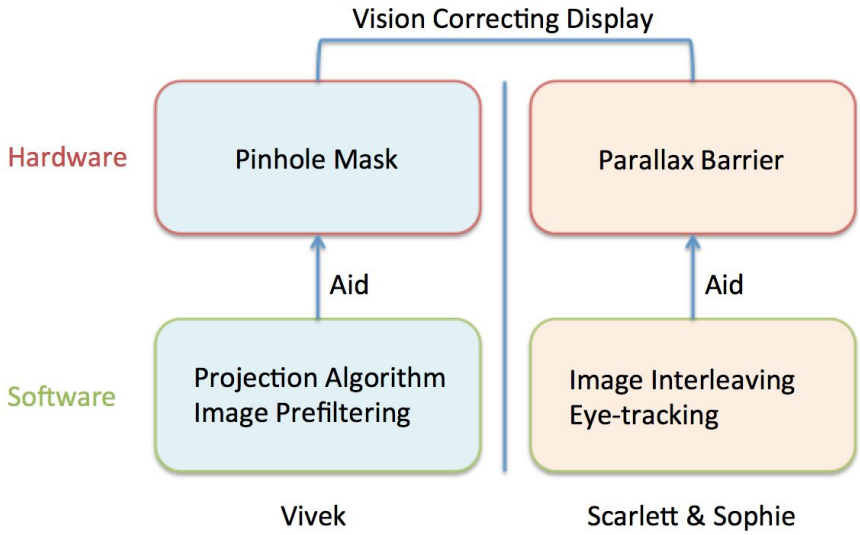


Fig. 10 Group organization (figure referred to Sophie).

Based on the organization of our group and the time limitations, it is not reasonable to use the waterfall model where one team needs to wait until the other finishes. So we instead use the agile model of project management (Lotz, 2013), which is better for parallel tasking between the two different teams. Besides, since the agile model is more tailored towards software development, yet our project also contains many hardware implementation and experiments, we adopted a flexible project management model that extends the agile model (Smith & Oltmann, 2010). Specifically, we implement our project management method in the following ways:

1. Self-organizing small teams.

Due to the diverse nature of the sub-problems of our research, we have divided our research group into several small teams. Each team has several meetings every week. The meetings are for choosing weekly goals, dividing work between each member of the team, synthesizing work that has been accomplished by individual members, and write status reports.

2. Combination of software and hardware research.

Since hardware costs more (in direct cash outlay and production delays) than software, and because re-designs are particularly painful, we do software simulations before hardware implementations. For example, we simulate the results of using different kinds of time-multiplexing barriers and pinhole masks before implementing them on expensive devices. Techniques from software development management can therefore be applied to the construction of the simulations. For the hardware experiments that require spending time all together, we improve effectiveness and efficiency by making full preparations beforehand, with the preparation in a multi-functional group being that everyone takes charge of one area: materials by someone with mechanical experience, process design by physics and vision experts, etc.

3. Information exchanging and work synthesis among the research group.

Every week we hold a meeting with the whole research group where each team gives a status report. The goals of these meetings are to find problems in the compatibility of the different results, absorb ideas from different perspectives, and inform everyone in the group as well as the remote researchers about the overall achieved progress. The meetings are presided by Prof. Barsky and David Grosof, who with their knowledge on the subject give feedback about the work achieved and suggestions about the venues in which to pursue research.

3. Social and Industry Context

In order to understand the scope of our project fully, it is important to address the social context of our project, which is the context of the prevalence of visual aberrations among the general population. According to Huang and Barsky (2014), "global surveys estimate that 153 million people worldwide are visually impaired due to uncorrected refractive errors" (p. 15). They further addressed that "246 million people have low vision (below 20/60), 43% of which is due to simple uncorrected refractive errors (such as myopia, hyperopia, or astigmatism)" (Huang & Barksy, 2014, p.15). The targeted user is anyone whose eyes' refractive errors cannot be simply corrected with eyewear equipped with conventional (including multifocal and progressive) spectacle lenses or contact lenses.

To date, there are mainly three ways to correct aberrations of the eye: eye glasses, contact lens, and refractive surgery. Market research reports reveal that the demand for glasses and contact lens has greatly increased, "as the US population ages and the number of people with vision related health complications increases" (Glasses & Contact Lens Manufacturing in the US: Market Research Report, 2016, p5). If our research leads to a product coming to market, it will take up a portion of the market share of the current glasses and contact lens industry for it improves upon the weakness of glasses and contact lenses to correct high order aberrations.

4. Industry Analysis

Having analyzed the social and industrial dimension of our project, we can now perform an industry analysis of our potential market and identify potential competitors. The bargaining power of our customers would be low, as for our targeted users, who possess high order

aberrations, there currently exists no product that can correct their vision perfectly without resorting to surgery. The threat of a substitute product is low for a similar reason. However, if we want to broaden our target market to all people suffering from ocular aberrations, however mild they might be, then the bargaining power and the threat of substitute become very strong since there are cheaper and more applicable devices such as glasses and contact lenses that can solve the vision problems efficiently. The current industry structure also makes the threat of new entrants very low, since Prof. Brian Barsky has conducted research in this domain for a long time in the academic world and is its most prominent representative. It will take a lot of time and effort for a new competitor to catch up and enter the field of vision correcting displays. The material required to manufacture the displays can be found from a lot of different suppliers, and the physical design of the display is not technologically complex, therefore there is not much bargaining power from the suppliers or manufacturers. Moreover, since the product would be new and would not exist in the market before our introducing it, there would be no immediate rivalry from existing competition. All of these facts lead to the following conclusion: if our research leads to our bringing a product to market, it will be important to maintain our technological uniqueness, and it would be more efficient to target specifically users who suffer from high order aberrations.

References

Boehm, B., & Turner, R. (2003). *Balancing Agility and Discipline: A Guide for the Perplexed, Portable Documents*. Addison-Wesley Professional. Chapter One p.5 - p.25.

Glasses & Contact Lens Manufacturing in the US: Market Research Report. (n.d.). Retrieved

October 17, 2016, from http://www.ibisworld.com/industry/default.aspx?indid=882.

Garg, S., Tripathi, A., & Cutrell, E. (2016, March). Accurate eye center localization using Snakuscule. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on* (pp. 1-8). IEEE.

Dodgson, N. A. (2005). Autostereoscopic 3D displays. *Computer*, *38*(8), 31-36.

Efrat, N., Didyk, P., Foshey, M., Matusik, W., & Levin, A. (2016). Cinema 3D: large scale automultiscopic display. *ACM Transactions on Graphics (TOG)*, *35*(4), 59.

Huang, F. C., & Barsky, B. A. (2014, June). 27.3: Computational Approaches to Aberration Compensation for Vision Correcting Displays. In *SID Symposium Digest of Technical Papers* (Vol. 45, No. 1, pp. 357-359).

Huang, F. C., Wetzstein, G., Barsky, B. A., & Raskar, R. (2014). Eyeglasses-free display: towards correcting visual aberrations with computational light field displays. *ACM Transactions on Graphics (TOG)*, *33*(4), 59.

Lagerwall, S. T., & Clark, N. A. (2004). Submicrosecond bistable electro-optic switching in liquid crystals. In *Crystals That Flow: Classic Papers from the History of Liquid Crystals* (pp. 553-557). CRC Press.

Liquid Crystal Technologies (2008). Pi Cell (Optically Compensated Bend – OCB Cell). Retrived online: http://www.liquidcrystaltechnologies.com/tech_support/Pi_Cell.htm.

Lotz, M. (2013). Waterfall vs. Agile: Which is the Right Development Methodology for Your Project? Retrived online: http://www.seguetech.com/waterfall-vs-agile-methodology/.

Parallax Barrier. (n.d.) In *Wikipedia*. Retrieved May 3, 2017, from https://en.wikipedia.org/wiki/Parallax_barrier.

Peterka, T., Kooima, R. L., Girado, J. I., Ge, J., Sandin, D. J., Johnson, A., ... & DeFanti, T. A. (2007, March). Dynallax: solid state dynamic parallax barrier autostereoscopic VR display. In *Virtual Reality Conference, 2007. VR'07. IEEE* (pp. 155-162). IEEE.

Perlin, K., Paxia, S., & Kollin, J. (2000). An Autostereoscopic Display; SIGGRAPH 2000 Conference Proceedings. *New Orleans, Louisiana. July*, 23-28.

Sinhal, K. (2017). Training a better Haar and LBP cascade based Eye Detector using OpenCV. Retrieved online: https://www.learnopencv.com/training-better-haar-lbp-cascade-eye-detector-opencv/.

Smith, P. G., Oltmann, J. (2010). Flexible Project Management: Extending Agile Techniques beyond Software Projects. *2010 PMI Global Congress Proceedings – Washington, DC*.

Timm, F., & Barth, E. (2011). Accurate Eye Centre Localisation by Means of Gradients. *VISAPP*, *11*, 125-130.

Wang, Z., & Xiao, S. (2013). Simulation of Human Eye Optical System Properties and Depth of Field Variation. *International Journal of Machine Learning and Computing*, *3*(5), 413.

Appendix A

Derivation of the Parallax Barrier Model Parameters Formula

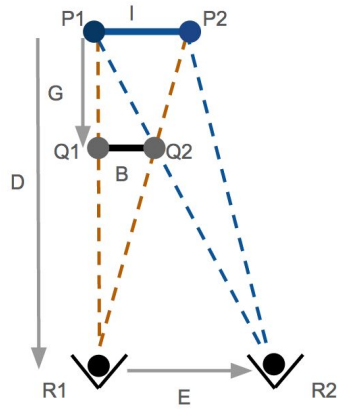


Fig. A1 Illustration of the similar triangles for parameters calculation.

Fig. A1 shows a part of the image and barrier from Fig. 3. Since $\Delta P_1 Q_1 Q_2 \sim \Delta P_1 R_1 R_2$, we have:

$$B/E = G/D, \tag{A1}$$

$$\text{so } B = G \cdot E/D. \tag{A2}$$

Since $\Delta R_1 Q_1 Q_2 \sim \Delta R_1 P_1 P_2$, we have: $B/I = (D - G)/D$. By substituting B, we have:

$$GE/I = D - G, \tag{A3}$$

$$\text{so } I = G \cdot E/(D - G). \tag{A4}$$

Then we have Eq. (1) and Eq. (2) derived.

Appendix B

Derivation and Discussion of the Distance Detection Equation

Appendix B.1

In this appendix, we show how we calculate the distance between the eye and the screen (Eq. (6)). Since we use the embedded camera to capture face images to do further calculation, the eye-screen distance is actually the eye-camera distance. In Fig. B1, point O is camera, and F is the focal length of the camera. Q_1Q_2 is the eye position with related distance values D_1 and e_1 we measure as reference. $Q_1'Q_2'$ is the new eye position. We capture the pixel eye center distance e_2 using eye detection method introduced in section 3, and D_2 is what we want to calculate from the detection program. Here, we assume that the distance between the eye centers remains the same while the user is moving, that is: $E_1 = E_2 = E$.

From the thin length equation, we have:

$$Q_1Q_2/P_1P_2 = D_1/F\,, \tag{B1}$$

$$\text{so } E/e_1 = D_1/F\,, \tag{B2}$$

and

$$Q_1'Q_2'/P_1'P_2' = D_2/F\,, \tag{B3}$$

$$\text{so } E/e_2 = D_2/F\,. \tag{B4}$$

From Eq. (B2) and Eq. (B3), we have:

$$e_2/e_1 = D_1/D_2\,, \tag{B5}$$

$$\text{so } D_2^{predict} = D_1 e_1 / e_2 . \tag{B6}$$

Here, we use "predict" to denote that the calculation result D_2 is the predict output of our program. Therefore, we have Eq. (6) derived.

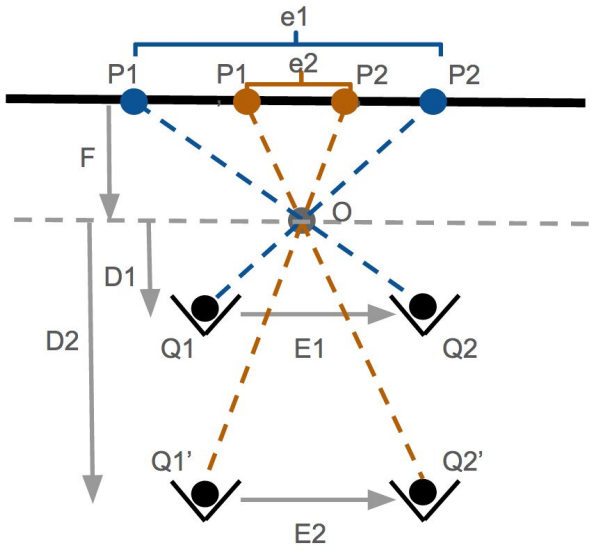


Fig. B1 Illustration of eye-camera distance detection.

Appendix B.2

In this Appendix section we want to prove that we can always get same distance value for different viewing angles on the same plane. This is what we desired for our parallax barrier, because the eye-screen distance D we used in our parallax barrier model is the perpendicular distance to the screen, as is shown in Fig. 3. So if the user moves within a same plane parallel to the screen, the perpendicular distance should be the same. Our distance detection method satisfies this requirement.

From Eq. (B5), we have:

$$e_2/e_1 = D_1/D_2 = D/D = 1 , \tag{B7}$$

$$\text{so } e_2 = e_1 . \tag{B8}$$

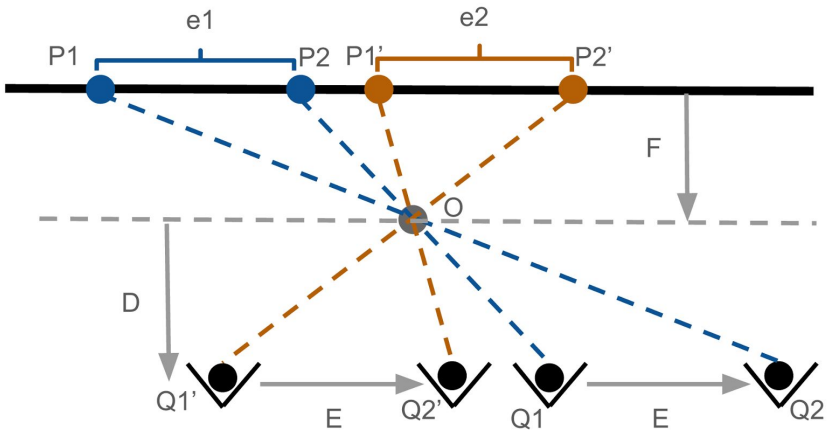


Fig. B2 Illustration of eye-camera distance in different angles.

Appendix B.3

In Appendix B.1, we assume that the distance between the eye centers remains the same while the user is moving. However, this is not always true. Human's pupil distance gets smaller when focusing on nearer object. In our case, $E_1 < E_2$. If we measure E_1 as reference, then from Eq. (B2) we have:

$$E_1/e_1 = D_1/F\,, \tag{B9}$$

$$\text{so } F = D_1 e_1/E_1\,, \tag{B10}$$

Substituting Eq. (B10) in Eq. (B4), we have:

$$\text{so } D_2 = E_2 F/e_2 = E_2 D_1 e_1/E_1 e_2\,. \tag{B11}$$

In this way,

$$D_2^{\,real} = E_2/E_1 \cdot D_2^{\,predict}\,, \tag{B12}$$

$$\text{and } D_2^{\,predict} = E_1/E_2 \cdot D_2^{\,real}\,. \tag{B13}$$

Since $E_1 < E_2$, we have $D_2^{predict} < D_2^{real}$. Therefore, our output result actually underestimates the eye-camera distance if using a closer reference position. Similarly, if using a reference position that is farther, we will get an overestimated result.