

EMPATHEYES: IMMERSIVE VIRTUAL REALITY SIMULATION OF VISUAL IMPAIRMENTS USING SMART PHONES

A PROJECT REPORT
SUBMITTED IN PARTIAL FULFIMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
Master of Engineering
IN
Computer Science Engineering

BY
Nirmalendu Prakash Diwakar



Computer Science and Automation
Indian Institute of Science
Bangalore – 560 012 (INDIA)

June, 2017

Declaration of Originality

I, **Nirmalendu Prakash Diwakar**, with SR No. **04-04-00-10-41-15-1-12280** hereby declare that the material presented in the thesis titled

EMPATHEYES: IMMERSIVE VIRTUAL REALITY SIMULATION OF VISUAL IMPAIRMENTS USING SMART PHONES

represents original work carried out by me in the **Department of Computer Science and Automation** at **Indian Institute of Science** during the years **2016-17**.

With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Date: 28th June, 2017

Student Signature

In my capacity as supervisor of the above-mentioned work, I certify that the above statements are true to the best of my knowledge, and I have carried out due diligence to ensure the originality of the report.

Advisor Name: Prof. Vijay Natarajan

Advisor Signature

© Nirmalendu Diwakar
June, 2017
All rights reserved

DEDICATED TO

Dr. Swaminathan Manohar

without whom this work wouldn't have been

Acknowledgements

I've been fortunate enough to have worked with people across industries and would like to acknowledge all ophthalmologists, optometrists from LV Prasad Eye Institute (LVPEI) and residents from Narayana Netralaya who were involved in validating this work. Special mention goes towards Dr. Padmamalini Mahendras of Narayana Netralaya who gave constant suggestions on improving usability of the system and correctness of simulations. Dr. Premnandhini Satgunam of LVPEI took time out of their busy schedule and in addition to validating our simulations gave a thorough list of improvements. The insights from these experts was pivotal in research and development of this work. I would also like to acknowledge Abhay Agarwal and Sujeath Pareddy of Microsoft Research, regular discussions with them helped give direction to this research. Finally, none of this could have been achieved if not for Dr. Swaminathan Manohar. I am grateful he gave me a chance to work on a project having societal impact. His vision and guidance gave direction to this research and was instrumental in every step of development of this work.

Abstract

Over 80% of blindness cases in the world today are preventable or treatable. Lack of awareness about visual impairments, their impact on sight, and the negative results of not seeking treatment at the early stages contribute significantly to this state. Further, the lack of empathy among medical fraternity, the patients' family, and caregivers, of the everyday impact of living with such vision impairments is another factor. We have developed EmpathEyes, a system that delivers an immersive virtual reality experience of a range of vision impairments as a possible tool to address this problem. We describe the iterative design process that alternated between implementation, validation by domain experts, and use by lay persons to arrive at a dual-screen solution. We present the details of the implementation and the results of the simulation. Early user trials involving more than 150 lay users demonstrate the potential of EmpathEyes.

Contents

Acknowledgements	i
Abstract	ii
Contents	iii
List of Figures	v
List of Tables	vi
1 Introduction	1
1.1 Contributions	2
1.2 Outline of report	4
1.3 Related Work	4
1.3.1 Simulation goggles, lenses and funnels	4
1.3.2 Personal computers	5
1.3.3 Virtual reality solutions	5
1.3.4 Mediated reality	6
2 Iterative design of EmpathEyes	7
2.1 EmapathEyes: Evolution, iterative design and development	7
2.1.1 The simulated world	7
2.1.2 Phase 1: Desktop based tethered HMD solution	8
2.1.3 Phase 2: Smart phone with Google cardboard	9
2.1.4 Phase 3: Dual device solution	9
2.1.4.1 LV Prasad Eye Institute	9
2.1.4.2 Narayana Netralaya	12

CONTENTS

3	System Architecture	13
3.1	High level design	13
3.2	Rendering pipeline	14
3.3	Network connection	14
3.4	Data synchronisation	15
4	Simulated visual impairments	16
4.1	Optimisations	16
4.2	Simulated vision impairments	17
4.2.1	Diplopia	18
4.2.2	Cataract	19
4.2.3	Colour Vision Deficiency (CVD)	21
4.2.4	Glaucoma	22
4.2.5	Age related Macular Degeneration (AMD)	24
4.2.6	Retinitis Pigmentosa	27
4.2.7	Refractive Errors	27
4.2.8	Retinal Detachment	28
5	Results	31
5.1	User Studies	31
6	Conclusion	36
6.1	Limitations of our Approach	36
6.2	Future Research	37
	Bibliography	39

List of Figures

1.1	UI of EmpathEyes shows monoscopic view on the phone screen along with a list of presets for some of the simulated impairments, inset shows the immersed user wearing a Google cardboard	2
2.1	Screen shots from our virtual house	8
3.1	High level system design	13
3.2	Rendering pipeline	14
4.1	Optimisation techniques	16
4.2	Monocular diplopia simulation screenshot	18
4.3	Cataract simulation screenshot	21
4.4	A stereoscopic screenshot with red-green CVD in the left eye and no CVD in the right eye	22
4.5	Glaucoma simulation screenshot	23
4.6	Field graph showing pixel displacement using algorithm 6	25
4.7	Age related macular degeneration simulation screenshot	26
4.8	4.8a shows Retinitis Pigmentosa vision and 4.8b shows the same view without any visual impairment in low light	27
4.9	Floater rendering and movement	29
4.10	A stereoscopic screenshot of retinal detachment screenshot with left eye having retinal detachment and no visual impairment in the right eye	30
5.1	Pre and post EmpathEyes experience trend on awareness	33
5.2	Pre and post EmpathEyes experience trend on empathy	34
5.3	Pre and post EmpathEyes experience trend on risk comprehension	35

List of Tables

1.1 Simulated Vision Impairments	3
--	---

Chapter 1

Introduction

As of 2010, World Health Organization (WHO) Prevention of Blindness and Deafness Programme estimates 285 million people as clinically blind of which 246 million have severe or moderate visual impairment. Out of the total number of blindness cases worldwide 80% of those could have been prevented [21].

India is home to the largest number of people with visual impairments. Further the vast majority of these cases are due to causes like cataract which is reversible and glaucoma, which if detected and treated early can prevent blindness. Poverty, lack of access to adequate medical facilities and paucity of trained personnel are some of the reasons for such a large number of people with preventable blindness. In this paper we develop a system to address another important factor, that of lack of awareness about visual impairments, their impact on sight, and the negative results in not seeking treatment at the early stages. The secondary issue we address is the lack of empathy among medical fraternity, the patients' family, and caregivers, of the everyday impact of living with such vision impairments. Even in the case of other impairments, which may not be preventable or reversible, a better understanding of the nature and impact of the impairment and an understanding of the progression of the negative effects of the impairment will be great value to both patients and their families.



Figure 1.1: UI of EmpathEyes shows monoscopic view on the phone screen along with a list of presets for some of the simulated impairments, inset shows the immersed user wearing a Google cardboard

1.1 Contributions

The current practice, in city-based clinics in India, is for ophthalmologists and optometrists to show patients and their kin brochures containing artistic rendition of the condition which the patient is facing, and explain the effects of specific visual impairments. We present EmpathEyes, an immersive virtual reality simulation system for vision impairments that provides the following benefits and is a major improvement over current practice:

- A Google cardboard based VR solution that only requires commodity smart phones and hence can be widely used (EmpathEyes can also run on Oculus CV1, Desktop PC, Android and iOS phones).
- Simulate a comprehensive set of vision impairments (listed in Table 1.1 with the ability to simulate the progression of the impairments).

- Provision for an expert (Ophthalmologist or optometrician) to direct the simulation and modify the parameters of the impairment in real-time by using a second device (smart phone or a laptop) as a controller. The simulation content is also mirrored on the control smart phone to allow the expert to understand the point of view of the user.
- Ability to independently control the impairment level in each eye, including the commonly occurring case of normal vision in one eye.
- Requires no infrastructure and minimum training for using the system
- Designed with inputs from and validated by experienced ophthalmologists for accuracy of the simulation
- Novel ways of simulating visual impairments which are more optimised and a closer approximation of the actual perceptual symptoms of an impairment than the traditional algorithms
- Ours is the first virtual reality simulation of retinal detachment and retinitis pigmentosa
- The entire code of the system is available in open source.

Impairment	Symptoms
Diplopia	Simultaneous perception of two images of a single object that may be displaced horizontally, vertically, diagonally
Cataract	Vision appears as if looking through ground glass
Colour Vision Deficiency	Decreased ability to see color or differences in color
Glaucoma	Symptoms include loss of peripheral vision characterised by reduction in field of view or complete loss of vision.
Age related Macular Degeneration	Symptoms include the foveal vision becoming blurred and distorted, colours appear less vivid
Retinitis Pigmentosa	Reduced field of view and reduced ability to resolve fine details
Myopia	Objects farther away are blurry
Hypermetropia	Difficulty seeing objects close to the eye
Astigmatism	Vertical or horizontal lines appear blurred
Retinal Detachment	Flashes, shower of floaters and loss of peripheral vision

Table 1.1: Simulated Vision Impairments

1.2 Outline of report

We are not aware of any other solution that meets all or even most of the criterias mentioned in Section 1.1. We present a survey of existing solutions in Section 1.3. We used an iterative design process to drive the development of EmpathEyes that alternated between implementation and testing and validation by experts and end users and this is described in Section 2.1. We then describe the system architecture and the implementation framework in Section 3.1. A detailed description of each of the impairments and their implementation is given in Section 4.2. We present the results of user studies with EmpathEyes in Section 5.1. We briefly list the limitations of our current work in Section 6.1 and conclude with pointers to next steps of this research.

Early user trials involving more than 150 users with the simulations mentioned in table 1.1 demonstrated the potential of EmpathEyes to be a powerful tool for educating and for creating empathy.

1.3 Related Work

Different approaches have been taken in the past to simulate vision impairments and these can be broadly divided into four categories:

1.3.1 Simulation goggles, lenses and funnels

Early work to simulate cataract vision was done using petroleum jelly on a F-5 lens system. Photos using such a lens demonstrated relationship between image contrast and resolution as a function of the percent of cataract present [28].

Simulator goggles have lenses overlaid with vinyls or paints or damaged to simulate different vision impairments or levels of vision impairment [2]. Zagar et al. [16] have developed simulator goggles for glaucoma, cataracts, macular degeneration, diabetic retinopathy and retinitis pigmentosa. One hundred pharmacy students were enrolled to study the accuracy of the simulations by wearing these goggles. Final results show that the simulations were reliably reproducing characteristics of simulated vision impairments. In another study [9] S. Aball et al. show that simulator spectacles are feasible for evaluating different levels of vision impairments.

Though simulator spectacles are a cheap way to simulate vision impairments with high accuracy, the variation in severity is limited as each level of impairment will require a lens with different level of damage or overlay, further it requires a setup time for changing between these intensities.

1.3.2 Personal computers

Vision impairments are usually simulated on computer screens as an aid to developers and designers for making accessible designs and finding lapses in accessible design. Goodman et al.[13] developed a software for demonstrating the effects of common vision and hearing problems so that designers can empathise with the visually impaired and take an empathetic design approach. Lewis et al. [15] explore feasibility of using a game engine for simulating visual impairments by creating a virtual model of their university cafeteria for increasing empathy for the visually impaired. They validated their simulations with opticians and a visually impaired consultants and tested it on twenty one students for assessing their simulator’s impact on empathy and awareness. Banks and McCrindle [11] developed a vision impairment simulator to simulate customised common vision impairments like macular degeneration, glaucoma, cataracts, diabetic retinopathy and colour blindness. Web browser extensions like Spectrum (available on Chrome store) simulate colour vision defects (CVD) help web designers in finding accessibility lapses in their design. While, on the other end of the spectrum simulators like aDesigner [5] help web designers make websites which are accessible to blind users. More extensive desktop simulators simulate a wider range of visual impairments and are applied to the desktop screen [8]. While another software called EASE [18], simulates interaction experience of users with low vision and motor impairments. EASE did not intend to give an exact reproduction of vision impairments but gave an idea of software accessibility problems.

These simulators are accessible to the general public since personal computers are common unlike simulator goggles or spectacles which have to be bought separately. They also give control over the intensity of the simulations and allow for making customised simulations by user input, a flexibility which simulator goggles or spectacles do not possess. However, these simulators can not provide immersion comparable to simulator goggles. The overall objective of these simulators is quite different from our end goals. We provide these earlier works for completeness.

1.3.3 Virtual reality solutions

Virtual reality provides a synthetic 3D environment and is more immersive than the above two methods. One of the earliest such work utilises the immersive aspect of virtual reality to generate empathy amongst caregivers of stroke survivors[19]by adding motion blur and reducing the visual field in addition to simulating other associated motor impairments. Jin et al. [14] simulate colour blindness, age related macular degeneration and diabetic retinopathy and a create a 3D model of the eye for using it as an educational tool.

Väyrynen et al. [24] developed a virtual city model in their simulator and simulated macular degeneration, cataract, glaucoma and myopia. They give users way finding and target location task for evaluation. Unlike [15] they did not validate the correctness of their simulator by consulting experts but focused on creating visual impairment experiences for evaluating accessibility lapses in architectural designs. Thus, they can not be reliably used in educating the general public about the symptoms or used as a consultation aid in ophthalmology.

1.3.4 Mediated reality

Mediated reality is experienced when the sensory perception of everyday reality is modified. One of the earliest examples of mediated reality was created by Stratton [23] by inverting his vision by 180 degrees about the line of sight by using convex lenses placed in a tube. Today, mediated reality is typically experienced using a wearable computer or a smart phone.

Users can experience vision impairments in a familiar environment. Smartphone applications like Vision Sim developed by Braille institute, Synsimulator developed by Pixelwerk AS and ViaOpta Simulator by Novartis Pharmaceuticals Corporation simulate visual impairments on real time camera feed. These applications are available on Apple Appstore or Android Play store, One of the recent works simulate personalised colour vision loss [17] for building empathy for people with colour vision defects. Despite being commonly available, smartphone based mediated reality visual impairment simulations are not immersive as most smartphones have one RGB camera which is insufficient for giving perception of depth. Ates et al. [10] did a qualitative study showing a HMD coupled with head mounted stereo camera visual impairment simulator is more immersive and has higher potential for detecting accessibility problems. Werfel et al. [26] used a similar system as [10] and simulated visual and hearing impairments for increasing empathy.

Experiencing inaccurate visual impairment simulations can distort its understanding. Such simulators would not be a suitable tool for education and awareness.

Chapter 2

Iterative design of EmpathEyes

EmpathEyes developed over a period of 9 months. Three different prototypes were developed. With each prototype the performance and accuracy were improved. We took help of experts which included ophthalmologists, optometrists, engineers who work with patients having visual impairment between the development of these prototypes. We briefly mention about our validation procedure at two hospitals.

2.1 EmaphathEyes: Evolution, iterative design and development

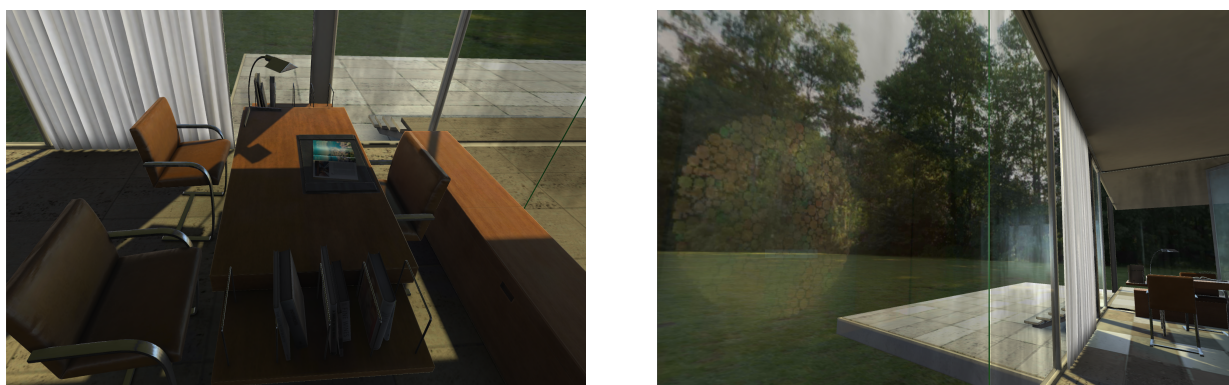
We embarked on a project of using immersive virtual reality technologies to enhance the quality of life of visually impaired persons (VIPs). We didn't have first or even second hand experience of any major visual impairment. Hence we started building EmpathEyes, an immersive virtual reality experience as a means for us to understand and possibly empathise with VIPs, as a first step in building any technology solutions to help improve the quality of their lives. In the rest of this paper we describe the progress we have made so far with EmpathEyes.

2.1.1 The simulated world

The user is placed in a synthetic scene of a modern house. This is a modified version of an architectural visualisation scene [1]. In addition to furnishing, there are Sloan optotype vision test chart, Ishiara test plates placed on the walls, and a low lit area in the house to appreciate the severity of visual impairments. To make this scene run with smoothly on smart phones, the textures of this scene were compressed, all light sources and reflections were pre-baked. Users can move around but cannot pass through furnitures and walls and will receive haptic feedback (where available) on colliding with objects in this virtual house. The rich set of furnishings

and artifacts as well as the standard vision test entities were used so that both subjective and objective aspects of vision impairments could be experienced. Figure 2.1a shows a close up view of some artifacts and figure 2.1b shows a panoramic view of looking through the window into some greenery outside.

EmpathEyes evolved over a 9 month period alternating between design, implementation and validation. We conducted two kinds of validation, the first from domain experts and second from lay end users. We present the progress chronologically including the learnings from each stage and the resulting evolution of EmpathEyes.



(a) A table chair arrangement in our virtual house (b) View through a glass wall of our virtual house

Figure 2.1: Screen shots from our virtual house

2.1.2 Phase 1: Desktop based tethered HMD solution

Our initial simulations were based on photos of how a VIP sees the world on National Eye Institute’s [3] website. In addition, literature about cause of each of the simulated eye defects and their associated symptoms, and literature on earlier software simulations [18, 14, 13, 25, 15, 10, 24, 11] were consulted. This version was built using Unity Engine and used the Oculus Rift as the user device, requiring a high-end PC to drive the experience. The initial simulations were presented to Domain Expert 1(DE1)¹

The test set up was as follows: The user was seated in a swiveling chair wearing the Oculus Rift. The monocular view was shown in the PC monitor and the simulations were controlled by us. The user had control of the navigation in the virtual world and was able to look around and move around using an Oculus remote controller. This allowed us to understand the specific comments of the user in context. The specific impairments and the severity of the impairment

¹Padmamalini Mahendradas, head of Department of Uveitis and Ocular Immunology at a leading Eye hospital at Narayana Nethralaya, Bangalore and a member of International uveitis study group (IUSG) and International Ocular Inflammation Society (IOIS), and executive committee member of Uveitis Society of India.

was set by us. The testing methodology was to show DE1 the simulations of each of the impairments, tell the name of the visual impairment simulation and ask her view on the fidelity of the simulation. Correctly represented aspects, incorrectly simulated aspects, and missing features were pointed out by the expert along with suggestions for improvement.

After incorporating all the inputs and validating it again with the DE1, we presented the system in an informal demo setting and the results are discussed in Section [5.1](#)

2.1.3 Phase 2: Smart phone with Google cardboard

Encouraged by the positive response from the general audience, we considered ways to make the experience more widely accessible and hence embarked on porting the solution to a Cardboard-based VR experience. Unity being a cross platform engine provided an easy migration path. Our desktop based simulator was ported to a smartphone and retinitis pigmentosa and hypermetropia simulation were added to this version. The ported simulations were colour blindness, cataract, diplopia, macular degeneration, retinitis pigmentosa, glaucoma, myopia and hypermetropia. However, the performance gap between a Oculus compatible PC and even a high-end smartphone required us to make many optimisations. After porting the simulations, an Android version was demonstrated again to DE1 who verified the accuracy of the simulations.

The next public demo was done at Open Day at CSA, IISc. 60 people tried out EmpathEyes. This exercise unearthed a few design problems in our solution using a single device for simulating visual impairments in virtual reality. Some users reported that they did not catch what the voice prompt said because the names of visual impairments are not used commonly in English. They wanted a written handout of the simulation they were looking at before experiencing the simulation. Some users wandered out of the virtual house into the woods, and since there were no everyday objects or colour variations in the woods they were unable to appreciate visual impairments. This was hard to find out as we did not know what the user was looking at.

2.1.4 Phase 3: Dual device solution

The design problems of our one device solution were mostly rectified by developing a two device solution. In the two-device solution, the simulation was unchanged from the previous version. The control of the experience was taken over by either a laptop or a second smart phone which also mirrored the display. This dual device setup was extensively validated and tried out at the LV Prasad Eye Institute in Hyderabad, India and Narayana Netralaya in Bangalore India.

2.1.4.1 LV Prasad Eye Institute

The process used for the validation as well as user studies that followed at LVPEI is as follows:

1. The controls for moving in the virtual world are first explained. The user wears the Cardboard with the primary smart phone mounted.
2. The user familiarizes with the scene and with moving around the virtual world with no visual impairments introduced. The user spends around 5 minutes in this step
3. The visual impairments are introduced from the control device. The change in the impairment is announced with a beep. The name of the impairment is not announced. The user is just told that the impairment starts at the onset stage and progresses through to an advanced stage.
4. The user spends 2-10 minutes in each simulation. The user is asked to identify the impairment and point out which visual features suggest the impairment.
5. The simulation was stopped between each impairment to minimize possibility of VR sickness.

Diplopia, cataract, colour vision disability (tritanopia, protanopia, deuteranopia), early stage of age related macular degeneration, late stage of age related macular degeneration, glaucoma, retinitis pigmentosa, myopia, hypermetropia, astigmatism (horizontal and vertical focus) simulations were the impairments simulated in this study.

We obtained the generous help of our second domain expert (DE2)¹ in validating the simulations. The positive outcome of the trial was that DE2 was able to correctly identify ALL of the impairments simulated by just going through the simulations with no other cues or hints. This was an excellent validation of the broad correctness of the simulations. In addition DE2 provided detailed inputs for improving the fidelity of most of the simulations.

Inputs on improving simulation from DE2

- Snellen chart should have a reference point with user distance. This could be replaced by a logMAR chart using Sloan optotypes which calibrates based on user's position.
- Patients are diagnosed for diplopia by testing them for visual confusion. It involves closing one eye and looking through the other. If by doing so, the patient can escape double vision they have binocular diplopia. The simulation as shown, simulates monocular diplopia which does not have visual confusion.

¹Dr Premnandhini Satgunam, MS and PhD in Vision Science from the Ohio State University, and was a postdoctoral fellow in the field of vision rehabilitation at Schepens Eye Research Institute, Harvard Medical School, Boston, USA.

- She briefed on other colour vision tests like Hardy Rand, D15, FM100U test and added that for simulating perfect colour vision test in our simulators we will have to do hetrochromatic flicker photometry. In our current simulation she pointed that the virtual house environment is nice for appreciating colour vision disability (CVD) but more coloured objects like flowers should be added to the lawn .
- Based on a user comment about retinal detachment flashes not being simulated correctly in section 2.1.2 she explained how retinal detachment flashes can be understood by closing and pushing against one of our eyes. The light perceived is due to pressure phosphine which happens because retina was stimulated and brain processed it as light. This flash is similar to what retinal detachment patients experience.
- Macular degeneration patients are evaluated with Humphrey’s visual field testing. In the results positions marked as black do not mean that the patient sees black at that point. It indicates that there will be distortion and we need to use a low pass filter to simulate the effect.

Four other experts contributed their time to try out EmpathEyes and gave feedback on the impairments that they specialise in.

E1 is an expert in Epidemiology and Ophthalmology. She has been teaching ophthalmology for the last 25 years and is currently a professor at LV Prasad eye institute.

E2 is an Ophthalmologist at LVPEI and CEO of a hospital chain. She has Retinitis Pigmentosa.

E3 is an ophthalmologist with 15 years of experience. She is the head of Centre for Glaucoma Care at LV Prasad Eye Institute.

E4 is a Consultant Optometrist, currently pursuing his PhD from School of Optometry and Vision Science, University of New South Wales, Australia. His research interests lie in measuring impact of low vision on individual’s participation and quality of life, and various contextual factors that may affect participation. The encouraging result from E4 was that he was ready to deploy EmpathEyes in his clinic to educate patients and their families.

All of them were able to correctly identify a majority of the 9 presented simulations. In the cases where they did not, it was due to close similarities between impairments on some of the features. In addition, many inputs for improving the simulations were suggested by these experts.

2.1.4.2 Narayana Netralaya

The inputs from LVPEI evaluation were incorporated into EmpathEyes and then presented to 8 first year graduate students of ophthalmology at Naraya Netrayalaya. Visual impairments simulated were macular degeneration, cataract, monocular diplopia, retinitis pigmentosa, retinal detachment and glaucoma. The same test process described in Section 2.1.4.1 was used. In addition, if a user mis-identifies an impairment we ask questions about the visual features that prompted the decision. We then reveal the identity of the impairment and ask the participant's view as to what aspect of the simulation caused the mis-identification.

Subjects used their knowledge in ophthalmology and past experience of treating patients while identifying simulated visual impairments. Seven out of eight students identified all or five out of the six simulated impairments correctly. The primary reason for not identifying an impairment is because some perceptual symptoms are common to more than one impairment. In our case it was distinction between glaucoma and retinitis pigmentosa. This requires experience and patient history to distinguish between them.

The outcome of these extensive trials and validation by domain experts is that we are now confident that EmpathEyes portrays the effects of the diverse set of impairments with sufficient fidelity to be used for educating patients and their families. We also have a wealth of inputs on enhancing the repertoire of EmpathEyes as well as many alternate features to highlight specific aspects of the impairments. For instance, standard visual tools employed by ophthalmologists in testing patients could be introduced into the visual field. We discuss some of these enhancements in Section 6.2

Chapter 3

System Architecture

We describe the our modification to the rendering pipeline, input methods, network connection and data synchronisation between devices in our two screen solution over the network in this chapter

3.1 High level design

We describe the architecture of EmpathEyes. The complete project along with documentation can be found in the source code repository.¹

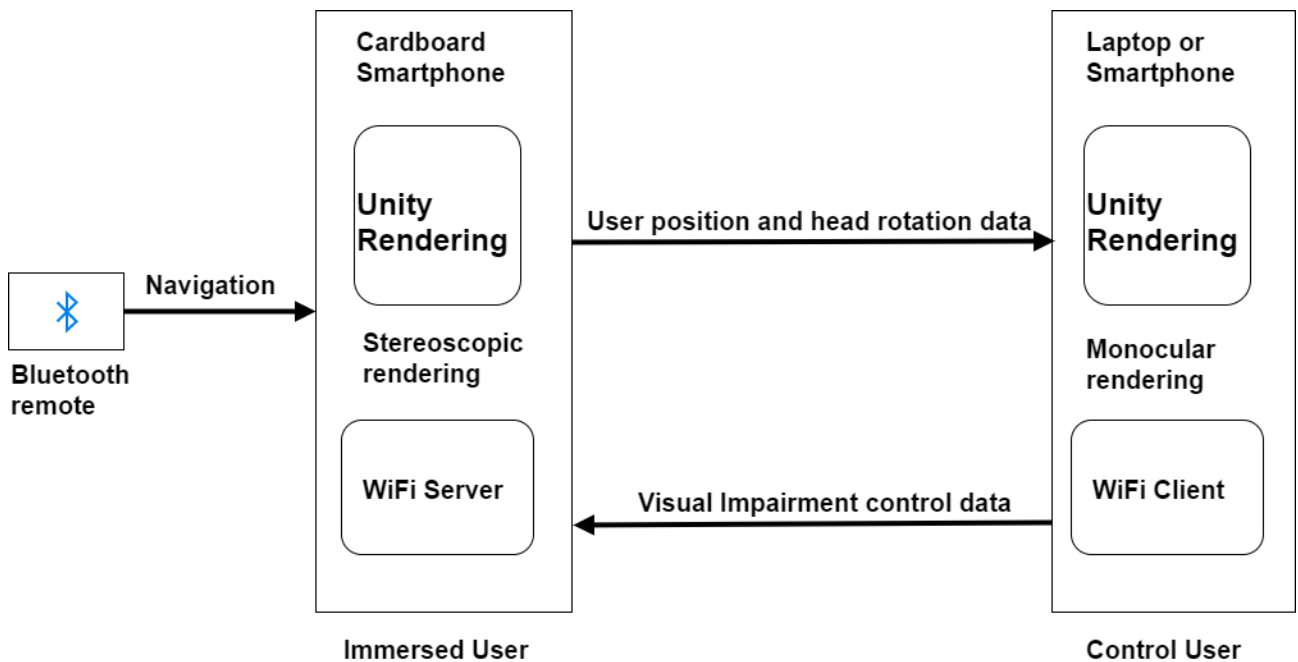


Figure 3.1: High level system design

¹<https://tinyurl.com/mlkqsfr>

Figure 3.1 shows the overall system architecture. The user experiencing the immersive simulation uses the Cardboard smartphone. The navigation within the world is controlled by a bluetooth remote. EmpathEyes bundles a wifi server and a local client, the device choosing to be the virtual reality experience device starts the device. The two devices connect to each other over a local wifi network. Within each device native rendering is done. This rendering is stereoscopic for the immersed user's device and monoscopic for the controller user's device. This rendering renders the scene and the visual impairment natively. The established network connection ensures the same view is rendered on both the devices.

3.2 Rendering pipeline

We used the game engine Unity and shading language CG and ShaderLab along with Oculus Rift SDK to develop our first version of the simulator. The Google Cardboard SDK was used to build our smartphone based simulator. We use a driver script which sends the full screen texture (called Render Texture by Unity) to a shader program which processes each pixel on the GPU in real-time. By utilizing a script to pass the Render Texture to a shader, we create a flexible system to create simulations for each visual impairment. Figure 3.2 shows the rendering pipeline. The part of the pipeline within the blue box is where we are modifying the scenes to create visual impairment simulations.

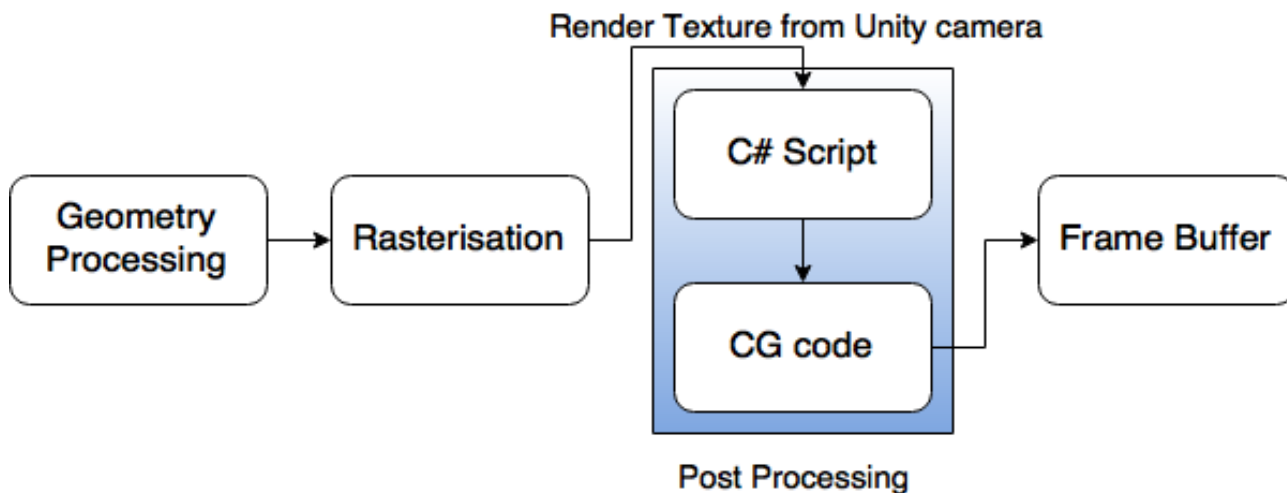


Figure 3.2: Rendering pipeline

3.3 Network connection

This device starts broadcasting a message containing a key and the version of the game on to the local wifi network using UDP broadcast feature. Any other device can start as a client, will

listen for broadcast messages will connect to our broadcasting server if it has the same version and key as received in the broadcast message. This establishes a two-way connection with the controller device, this could be another smartphone or a laptop. The expert can control the simulation by selecting the impairment and configuring the parameters (like scotoma size, vision acuity, visual field, glare, contrast, floaters etc.) either by using the keyboard (in the case of a laptop) or using the touchscreen of the control smartphone. This data is sent to the cardboard smartphone. The position and orientation of the user in the virtual world are sent to the controller device through the wifi link. This enables the expert to view the world as seen by the immersed user. The Unity rendering using the shader program runs natively on both the devices.

It is also possible to use the simulation in a single device mode where the device is configured to either cycle through a number of impairments or allow for extended experience of a single impairment. These interactions can be controlled by actions suitably mapped either to a bluetooth remote, or the switches on the cardboard.

3.4 Data synchronisation

Each device runs has two players. One of which is the player representative of the immersed user and the other representing the controller user both of whose states are synchronised over the devices. Each device will deactivate the other player in their scene. This is important so that each device doesn't contest over the state values. To distinguish the devices we check the number of connections it has, the immersed user device acts as a local server to its local client and as a remote server to the controller device which is a remote client for this device. Once we've distinguished we can keep separate instructions in the same class. Next, we need to distinguish between the representative player of each device. For this we tag each player differently. We need the position and rotation values from the immersed user device to controller user device and visual impairment data from the controller user to the immersed user device.

Chapter 4

Simulated visual impairments

We explain the algorithms, optimisations and the symptoms associated with each visual impairment.

4.1 Optimisations

To achieve our goal of running EmpathEyes on mobile GPUs we had to optimise our algorithms. We enlist and various optimisation techniques used in computer graphics which were harmonious in developing our simulations.

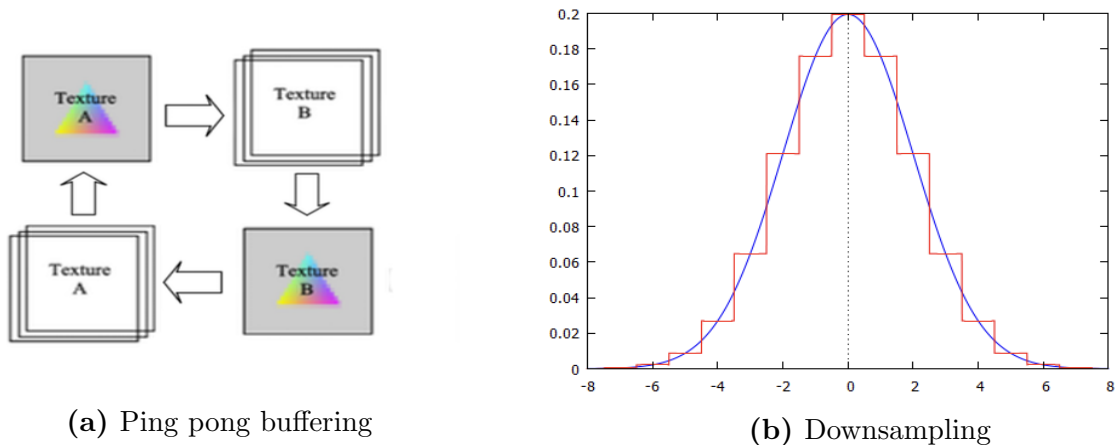


Figure 4.1: Optimisation techniques

- **Ping Pong buffering** Keep two frame buffer objects and keep alternating between them as inputs and outputs in the post processing stages. This technique is useful when the same texture needs to be post processed on again as in a gaussian blur. Figure 4.1a shows the ping pong technique.

- **Downsampling** Texture size is reduced by a user defined downscaling factor. This texture size is increased to its actual size again but the texture colour values are extrapolated based on the downscaled texture. This technique is used in producing heavy blurs, a 4 times downscaled image with a post processing of gaussian blur of 10 pixel width would approximate the results produced by a gaussian blur having 40 pixel blur width. Figure 4.1b shows a graphical representation of downsampling, the step graph approximates the values of the curve.
- **Bilinear filtering** This is a fixed function hardware optimisation. Bilinear filtering interpolates the colour value at a position based on four nearest texel values, since this is a fixed function hardware operation if we don't fetch at texel center positions of our texture then we can get information about multiple pixels. We use it as explained in an optimised blur algorithm [22] for linear sampling of textures thereby reducing the number of texture fetches.
- **Pre-baking global illumination** Global illumination is a system which computes how light is bounced off of different surfaces in the scene (indirect lighting) in addition to how its bounced off from a light source to a surface. Computing lighting at runtime increases the scene load time and reduces frame rates. We use an inbuilt Unity feature which not only computes the behaviour of lighting on static objects but also pre-computes all possible light bounces. In case a non static object is introduced in the scene which in our case will be the player, we have to add light probes [7], "Light probes are positions in the scene where the light is measured or probed during the bake or precomputation, and then at runtime the indirect light that hits non-static objects is approximated using the values from the probes that the object is closest to at any given moment". Therefore, even though the indirect lighting contribution of this source is not accounted for but it can still pick up bounced lighting from different direct and indirect sources.

4.2 Simulated vision impairments

The list of visual impairments that we have simulated are given in table 1.1. In this section we briefly explain the cause and perceptual symptoms of each visual impairment and the output from our system. We also give detailed notes on the implementation of each effect in order to facilitate further development of impairments not covered by our current work and for the creation of simulations customised to individuals.

4.2.1 Diplopia

Cause of Diplopia: It is commonly known as double vision. It can happen due to abnormal curvature of the cornea or misalignment of the eyes or impaired functioning of extra ocular muscles.

Perceptual Symptoms: Simultaneous perception of two images of a single object that may be displaced horizontally, vertically, diagonally (i.e., both vertically and horizontally). There can be two forms of Diplopia: Monocular and Binocular diplopia. Monocular diplopia is double vision in one eye. This doesn't go away when the other eye is closed. Binocular diplopia is caused by misalignment of the images of the two eyes and hence goes away when one eye is covered.

Developing the simulation: Algorithm 1 & 2 show the algorithms for Binocular Diplopia and Monocular diplopia. Figure 4.2 shows diplopia simulated using algorithm 2.

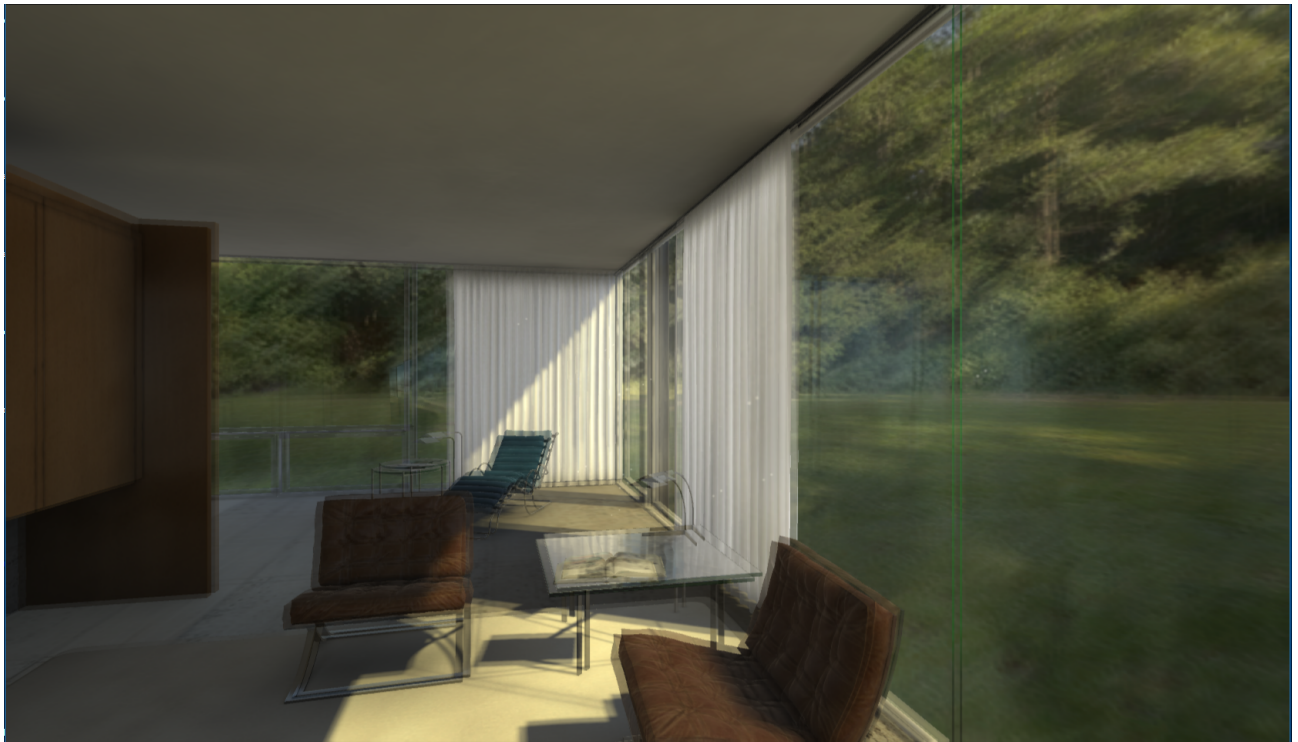


Figure 4.2: Monocular diplopia simulation screenshot

Algorithm 1 Binocular Diplopia

procedure BINOCULARDIPLOPIA(c, a, d, f) ▷ c stores the render texture of one of the cameras of the stereo camera, a decides the angle of displacement of pixels, d is the amount in pixels by which pixels are displaced & f is the flag deciding between left or right camera

$i \leftarrow c.uv.xy$ ▷ Pixel position being processed in the shader

if f is 1 **then** ▷ If the pixel being processed is from the render texture of left camera

$r \leftarrow i - d \times a$ ▷ Displace the current pixel by set displacement and specified angle

else ▷ If the pixel being processed is from the render texture of right camera

$r \leftarrow i + d \times a$ ▷ Displace the current pixel by set displacement and specified angle

end if

return r ▷ The displaced texture to be rendered

end procedure

Algorithm 2 Monocular Diplopia

procedure MONOCULARDIPLOPIA(c, a, d) ▷ c stores the render texture of the in game camera, a decides the angle of displacement of pixels & d is the amount in pixels by which pixels are displaced

$i \leftarrow c.uv$ ▷ Pixel information of the pixel being processed in the shader

$r \leftarrow (i.xy - d \times a) \times 0.70$

$r \leftarrow r + (i.xy + d \times a) \times 0.30$ ▷ Adding two texture lookups on either side of the displaced pixel in a skewed ratio

return r ▷ The displaced texture to be rendered

end procedure

4.2.2 Cataract

Cause of Cataract: Cataract happens due to protein build up in the lens. This can happen due to old age, diabetes or injury to the eye. Air pollution also increases the risk of getting cataract.

Perceptual Symptoms: Monocular diplopia, vision feels like looking through ground glass, glare from bright lights and during daytime. Need to see objects up close to resolve details.

Developing the simulation: Monocular diplopia is implemented as explained above. For ground glass effect a gaussian blur shader is used. A naive 2D gaussian blur algorithm is computationally expensive. This arose the need for an optimised gaussian blur algorithm.

We use a modified version of this[22] shader. The modified snippet is shown in algorithm 3. Additionally downsampling of the rendered texture and ping pong buffering is done for generating heavy blurs on our smart phone GPU. The glare and contrast difference associated with cataract is shown in algorithm 4. Figure 4.3 shows cataract simulated using this algorithm.

Algorithm 3 Cataract Blur

procedure CATARACTBLUR(c, s, d) ▷ c stores the render texture of one of the cameras of the stereo camera, s is an alpha parameter for deciding the strength of blur to give for depth, d is depth information of the current pixel

$i \leftarrow c.uv$ ▷ Pixel information of the pixel being processed in the shader

$l \leftarrow \text{Linear01Depth}(d)$ ▷ Scaling the depth value to be in 0-1 range

$b \leftarrow l^d$ ▷ Sets the blur width to increase exponentially with distance

$\text{GaussianBlur}(i, b)$ ▷ Send the blur width and current pixel information to gaussian blur function

end procedure

Algorithm 4 Cataract Glare & Contrast

procedure CATARACTCOLOUR(c, g, w) ▷ c stores the render texture of one of the cameras of the stereo camera, g is the colour representing yellowish tint and glare & w is alpha parameter for linear interpolation

$i \leftarrow c.rgba$ ▷ Colour information of the pixel being processed

$w.rgba \leftarrow \text{fixed4}(0, 0, 0, 0)$ ▷ Declaring variable w to white colour

$o \leftarrow \text{lerp}(\text{fixed4}(w.r, w.g, w.b, w.a), \text{fixed4}(c.r \times g.r, c.g \times g.g, c.b \times g.b, c.a), w)$ ▷ Linear interpolation for changing contrast and introducing glare

return o ▷ Pixel to be rendered post processing

end procedure



Figure 4.3: Cataract simulation screenshot

4.2.3 Colour Vision Deficiency (CVD)

Cause of CVD: Normal vision is trichromatic. It is initiated by the absorption of photons in three classes of cones, peak sensitivities of which lie in the long, medium and short wavelength regions of the spectrum. Reduced forms of colour vision arise from the effective absence of one of the retinal photopigments of the L type in protanopes, the M type in deuteranopes, and the S type in tritanopes. People coming under these three categories are called dichromats. Another rare form of CVD is rod monochromatism (Achromatopsia).

Perceptual Symptoms: Difficulty in distinguishing colours. Protonopes and deuteranopes have confusion identifying shades of green, red and yellows. Protonopes are insensitive to red light. While, deuteranopes are insensitive to green lights. Tritanopes are insensitive to blue light and have confusion identifying shades of green and blue. Rod monochromats see things in different shades of grey.

Developing the simulation: Dichromatic CVD is simulated using [20]. For Monochromatic CVD we convert each pixel of render texture to gray-scale using a colour matrix [12]. Figure 4.4 shows a stereoscopic screenshot with left eye having protanopia and right having normal vision. The result is that the immersed user's right eye compensates for CVD in the

left eye, making effect of CVD subtle.

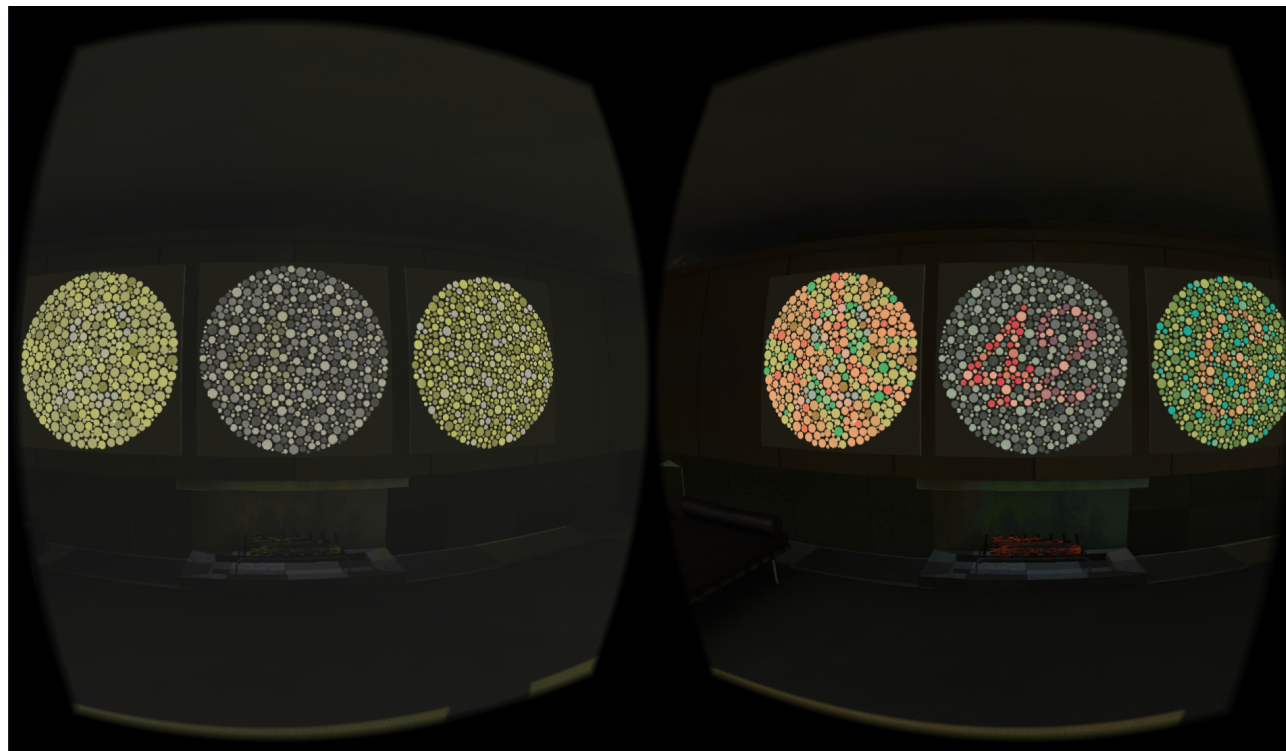


Figure 4.4: A stereoscopic screenshot with red-green CVD in the left eye and no CVD in the right eye

4.2.4 Glaucoma

Cause of Glaucoma: Humor maintains the shape and size of the eye, of which vitreous humor is also responsible for maintaining the intra-ocular pressure. Over production of Aqueous humor or inadequate drainage of it leads to increase in intra-ocular pressure which leads to Glaucoma.

Perceptual Symptoms: Symptoms include loss of peripheral vision characterised by reduction in field of view or complete loss of vision. Other symptoms include blurred Vision and appearance of halo around Lights.

Developing the simulation: Texture masks were created, with areas of vision loss represented by black colour having a soft gradient at the edges. Linear interpolation is done between rendered texture and the product of mask texture and rendered texture by a user defined amount. This amount increases or decreases the field of view, this is shown in algorithm 5. Higher values are used to simulate severe Glaucoma. Figure 4.5 shows glaucoma simulated using this algorithm.

Image is blurred using a two pass separable Gaussian filter. This image is then processed using Unity's post processing effect *Bloom Optimized*[6]. Bloom threshold is reduced to simulate severe Glaucoma. Halos are light objects in Unity. They were added to light sources in our scene. Their size is increased for simulating severe Glaucoma.

Algorithm 5 Field of vision loss

```

procedure VISIONMASK( $c, m, a, f$ )  ▷  $c$  stores the render texture of one of the cameras of
the stereo camera,  $m$  stores the mask texture &  $a$  store the overlay strength
   $i \leftarrow c.uv.rgb$                 ▷ Colour information of the pixel being processed
   $b \leftarrow m.uv.rgb$               ▷ Colour information of the mask at current pixel position
   $blended \leftarrow i \times b$       ▷ Overlaying colour information of mask over the image to be rendered
   $o \leftarrow lerp(i, blended, a)$ 
  return  $o$                         ▷ Pixel to be rendered post-processing
end procedure

```



Figure 4.5: Glaucoma simulation screenshot

4.2.5 Age related Macular Degeneration (AMD)

Cause of AMD: The macula is the centre part of the retina. This disease is generally age related and experienced in individual after the age of 60 years and hence known as Age related Macular Degeneration (AMD). This disease affects the central vision.

Perceptual Symptoms: Initially foveal vision becomes blurred and distorted, this area is called a scotoma. In later stages the size of the scotoma increases affecting the peripheral vision.

Developing the simulation: Since we do not have a gaze contingent solution our basic assumption is that the user is looking at the center of the screen. Therefore, the foveal vision would be a circle with center at the center of the rendered texture of each in game camera. We simulate an area of confusion, the algorithm for simulating which is shown in algorithm 6, it should be also noted that scotoma radius is only increased till 30 visual field and this corresponds to late stage of AMD. This is done as AMD is a self limiting visual impairment. Also, in later stages foveal vision becomes hazy and distorted. The field graph of the effect on pixels is shown in figure 4.6. For heavier distortion we write a distortion shader, with distortion being maximum in the central vision and reducing exponentially towards the periphery, the algorithm for which is shown in algorithm 7. In addition to this colours are reduced by giving a low weightage to the pixel during texture lookup. Figure 4.7 shows AMD simulated using this algorithm.

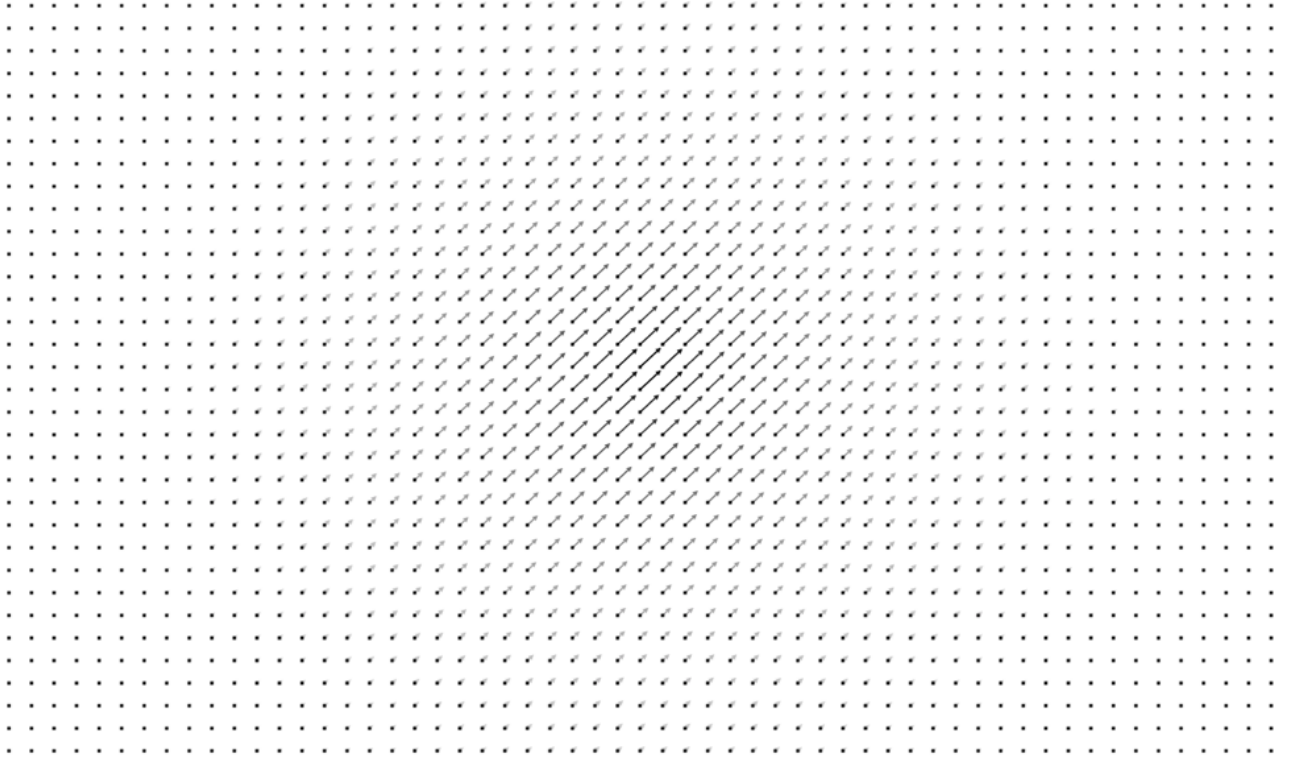


Figure 4.6: Field graph showing pixel displacement using algorithm 6

Algorithm 6 AMD scotoma

```

procedure AMD SCOTOMA( $c, a, d, n$ )      ▷  $c$  stores the render texture of the cameras,  $a$ 
specifies the range of scotoma,  $d$  is blur direction,  $n$  is the number blur iterations
   $i \leftarrow c.uv.xy$                     ▷ Pixel position being processed in the shader
   $offset \leftarrow e^{a \times \sqrt{(i.x-0.5)^2 + (i.y-0.5)^2}}$   ▷ Higher offset values for pixels in the center of the
screen
   $p \leftarrow i.xy - d \times offset$       ▷  $p$  and  $q$  store a copy of displaced pixel on either direction
   $q \leftarrow i.xy + d \times offset$ 
   $step \leftarrow (p - q)/n$               ▷ The number of copies of pixel to create between  $p$  and  $q$ 
  for  $i = 0; i < n; i++$  do
     $i = i + step$ 
     $o = i$ 
  end for
  return  $o/n$       ▷ The output is divided by number of iteration to remove for colour leak
end procedure

```

Algorithm 7 AMD Distortion

procedure DISTORTION(c, d) \triangleright c stores the render texture of one of the cameras of the stereo camera, & d sets the distortion strength

$i \leftarrow c.uv$ \triangleright Pixel information of the pixel being processed

$offset \leftarrow e^{k_1 \times \sqrt{(i.x-0.5)^2 + (i.y-0.5)^2}}$ \triangleright Higher offset values for pixels in the center of the screen

$a \leftarrow k_2 + k_3 * \sin(i.x * k_4)$ \triangleright a & b changes the position of the current pixel based on trigonometric functions

$b \leftarrow k_2 + k_3 * \cos(i.y * k_4)$

$distortion \leftarrow d \times float2(a + \sin(b), b + \cos(a))$

$o \leftarrow distortion \times (0.5 - offset)$ \triangleright Reducing from 0.5 to ensure offset at centre is maximum

return o \triangleright Pixel to be rendered post processing

end procedure



Figure 4.7: Age related macular degeneration simulation screenshot

4.2.6 Retinitis Pigmentosa

Cause of Retinitis Pigmentosa: It is a group of rare, genetic disorders that involve a breakdown and loss of cells in the retina. In early stages rods die which eventually leads to a breakdown causing loss of cones.

Perceptual symptoms: There is a progressive loss of visual field eventually leading to tunnel vision. In some cases, colour vision is also affected.

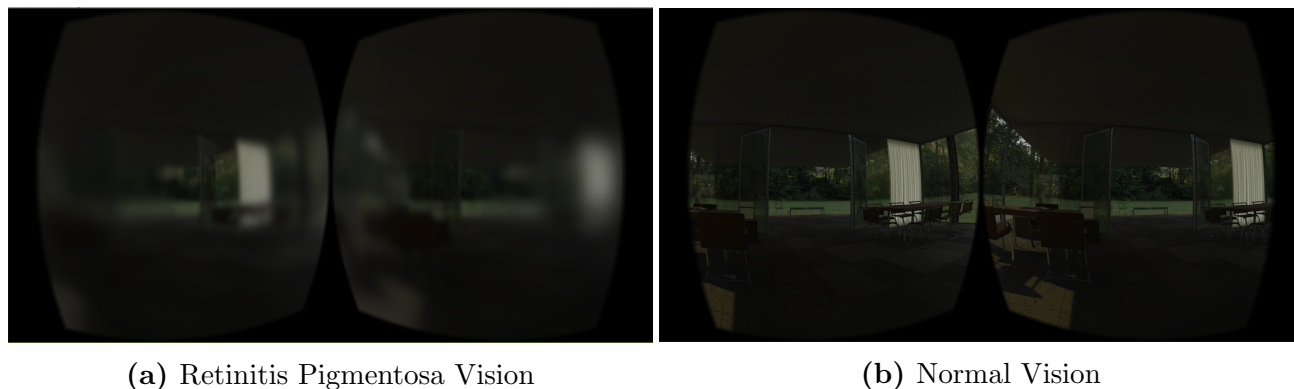


Figure 4.8: 4.8a shows Retinitis Pigmentosa vision and 4.8b shows the same view without any visual impairment in low light

Developing the simulation: A modified version of the blur algorithm mentioned in 4.2.2 is used. Instead of changing blur width by distance from the game camera, we increase the blur width as a sigmoid function with low blur width towards the center and higher towards the periphery. Also colour was reduced by linearly interpolating between grayscale and rendered texture image. The alpha parameter of this linear interpolation is reduced linearly as a distance measure from the pixel being rendered to the center of the screen. Figure 4.8 shows a comparison between normal vision figure 4.8b and our simulated retinitis pigmentosa vision figure 4.8a in a dimly lit environment.

4.2.7 Refractive Errors

We have simulated myopia, hypermetropia and astigmatism, which are some of the more common kinds of refractive errors.

Cause of Refractive Errors: Refractive errors happen when light rays do not focus directly on the retina due to the shape of the eye, lens, or cornea resulting in a blurred image.

Perceptual Symptoms: Distant objects appear blurred in case of Myopia while in Hypermetropia, closer objects become blurred. In case of Astigmatism horizontal or vertical lines appear blurred.

Developing the simulations: Blur algorithm from 4.2.2 is used. Blur width is increased linearly with distance for Myopia and reduced linearly from 40 cm distances to the camera.

For Astigmatism we first set whether the simulation should be for horizontal or vertical axis blur. Texture lookups are done at $\pm (3, 2, 1 \text{ and } 0)$ in the axis of blur specified. Each of the seven lookups are given a weighting factor, the lookups further from the actual position of the fragment are given a lower weighting factor. The total sum of weighting factors is insured to be 1.

4.2.8 Retinal Detachment

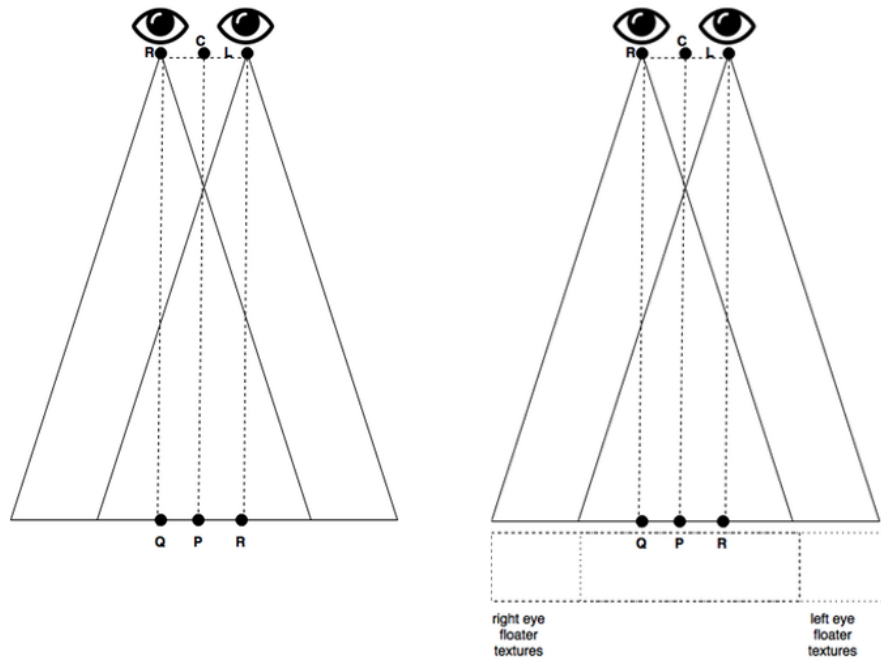
Cause of Retinal Detachment: If there is a strong adhesion between the retina and the vitreous, the posterior vitreous detachment can cause a fracture in the retina, if the retina itself is fragile at that point this can lead to a retinal tear, which will allow intra ocular fluids to go through and detach the retina, in the large majority of cases, retinal detachment happens abruptly and its unexpected.

Physical Symptoms: Includes flashes of light, shower of floaters and loss of peripheral vision.

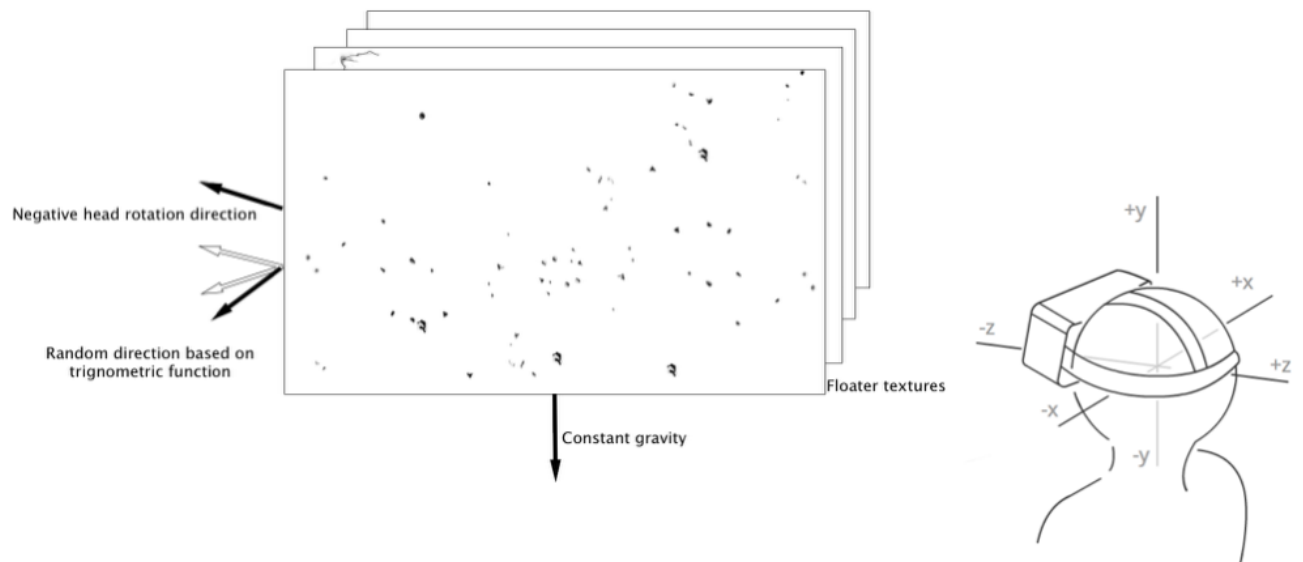
Developing the simulations: Light Flashes happen at random instances and the flashes fill a quadrant of the visual field. To simulate this a black mask texture is created with the quadrant to be filled with flash coloured white having a soft gradient at the edges. This texture is loaded along with our main texture. We check for the colour values, if it's not complete black we render the main texture. Else, we linearly interpolate the colour of the pixel being processed between it's actual colour given by the main texture and colour of the pixel at corresponding point in our mask texture. This behavior is triggered using a random variable.

Different floater textures are created and loaded in each of the in-game cameras since, loading the same texture would allow the HMD wearer to triangulate the position of floater texture and it would see a plane of floaters. This is shown in figure 4.9a. Next, each floater texture is given a random speed and trajectory which varies over time using trigonometric functions. A perturbation vector is calculated based on head motion of the HMD wearer and is added to the initial trajectory of each floater texture, as shown in 4.9b. The algorithm for floater rendering and movement is shown in algorithm 8.

Peripheral vision loss is simulated using the technique described in 4.2.4.



(a) Left image shows how eye triangulates objects, while right image is our modification for creating an illusion of depth



(b) Head rotation and affect on floater movement

Figure 4.9: Floater rendering and movement

Figure 4.10 shows retinal detachment simulated using this algorithm.

Algorithm 8 Floater movement

```
procedure FLOATERSMOVEMENT( $C, f, h, rand, a$ )       $\triangleright$   $C$  stores the render texture of
each of the cameras of the stereo camera,  $f$  store a floater texture,  $h$  stores the head rotation
data,  $rand$  store a random number between -1 to 1  $a$  store the opacity of the floater texture
  for each  $c \in \mathcal{C}$  do
     $i \leftarrow c.uv.xy$                              $\triangleright$  position information of the pixel being processed
     $moved \leftarrow i \times \alpha \times h \times \sin(rand)$    $\triangleright$   $\alpha$  is a constant, different textures are loaded per
camera and will have different constants and trigonometric function for perturbation
     $b \leftarrow f.uv$        $\triangleright$  Corresponding pixel information of the floater texture to the moved
render texture
     $b \leftarrow f.rgb \times i.rgb$ 
  end for
   $o \leftarrow lerp(i, b, a)$ 
  return  $o$                                         $\triangleright$  Pixel to be rendered post processing
end procedure
```



Figure 4.10: A stereoscopic screenshot of retinal detachment screenshot with left eye having retinal detachment and no visual impairment in the right eye

Chapter 5

Results

EmpathEyes went through thorough expert validation and user study. In this section we present result of our user study. Expert validation methodology and results were previously explained in chapter 2.

5.1 User Studies

Our first study was conducted using the Oculus Rift version to adults, in the age group of 24-70. All of them were technology savvy but had no specific ophthalmology domain expertise. Similar set up as used for DE1 was used: we were in control of the visual impairment being simulated and its severity and the users were free to move in the virtual house environment. Each user tried this demo for 4 to 7 minutes. Though a formal study was not done, some interesting comments were recorded. Some of the comments from the public were: “I already have floaters, never thought it can be an indication of a visual impairment, would get it checked soon”. Another user said, “I had retinal detachment, the simulated flashes aren’t correct. Flashes I experienced seem to come from the back of my eye”. Another user volunteered the information that he was color blind and was curious about how we have simulated it. We did not have time to pursue many possible insights we could obtain by getting further trials and feedback from the user. However, this clearly suggested to us the vast possibilities for this system in studies with users with visual impairments. In general, if a user knew someone who had a visual impairment, they were more interested in experiencing that specific visual impairment. This informally substantiated our goal of providing empathy. A user after experiencing Cataract simulation commented “So this is why my grandfather has to hold written text too close to his eyes for reading”. Another user who was recently diagnosed with very early stages of a degenerative disease (macular dystrophy) volunteered that information and wanted to know

more. Unfortunately, we did not have the simulations.

The second user study of EmpathEyes was at CSA Open Day. The following study was done informally to get a sense of how appealing or otherwise EmpathEyes can be. A rigorous follow up study is needed to validate the insights presented below. 60 people from a diverse age group tried the demo with each user trying it for about 7 to 8 minutes. Each participant filled paper questionnaire before and after trying out the simulations.

Each user started with normal vision and was in control of sequentially trying out various impairments. Since this was a one device solution we did not know what the user was looking at. Before each simulation the name of the visual impairment was given out as a voice prompt. Simulations were shown in a cyclic order, with users deciding when to go on to the next simulation. Visual impairment simulations shown had a fixed preset for onset stage and later stage. Onset stage was shown prior to later stage.

The responses to the questionnaires were recorded on a five point Likret scale with the meanings, 1 - No understanding, 2 - Vague understanding, 3 - Good understanding, 4 - Fairly good understanding and 5 - Detailed understanding.

We present the results of three questions from the survey. For the question on the self-reported level of awareness of visual impairments among the participants ranging from 1 (no understanding) to 5 (detailed understanding), the pre-experience average was 2.18 (Standard Deviation 0.76) and the post-experience average was 3.93 (Standard Deviation 0.74), the bar chart is shown in figure 5.1. To the question about the level of awareness of day to day difficulties encountered by people with visual impairments. The pre-EmpathEyes average was 2.49 (Standard Deviation 0.80) and the post-EmpathEyes average increased to 3.76 (Standard Deviation 0.87), the bar chart is shown in figure 5.2. Finally, we show the post-EmpathEyes trend in figure 5.3 on the question of whether, they'll recommend or take a close kin for vision consultation if they've been reporting problems in their vision.

There was high level of enthusiasm for experiencing the simulation. Some participants waited in line as much as 45 minutes to get a chance to try out. Not expecting the large turnout we had not readied two setups and hence had to turn away another two dozen more since the event time ran out. Though the study was not done very rigorously, the results from the questionnaires and our subjective experience of the reaction from the participants to the simulations demonstrated to us that EmpathEyes could be a good tool for educating people about visual impairments.

The extensive validation provided by domain experts over the multiple iterations imply that EmpathEyes is accurately portraying the visual experience of people with the simulated impairments. The results of the user studies, though informal, conducted with more than 150

users strongly indicate that EmmpathEyes can be a powerful tool for education of the impact of visual impairments. However we need to do further research and studies to deploy EmpathEyes at scale. In the next chapter we list some of the current limitations and our plans for future research next.

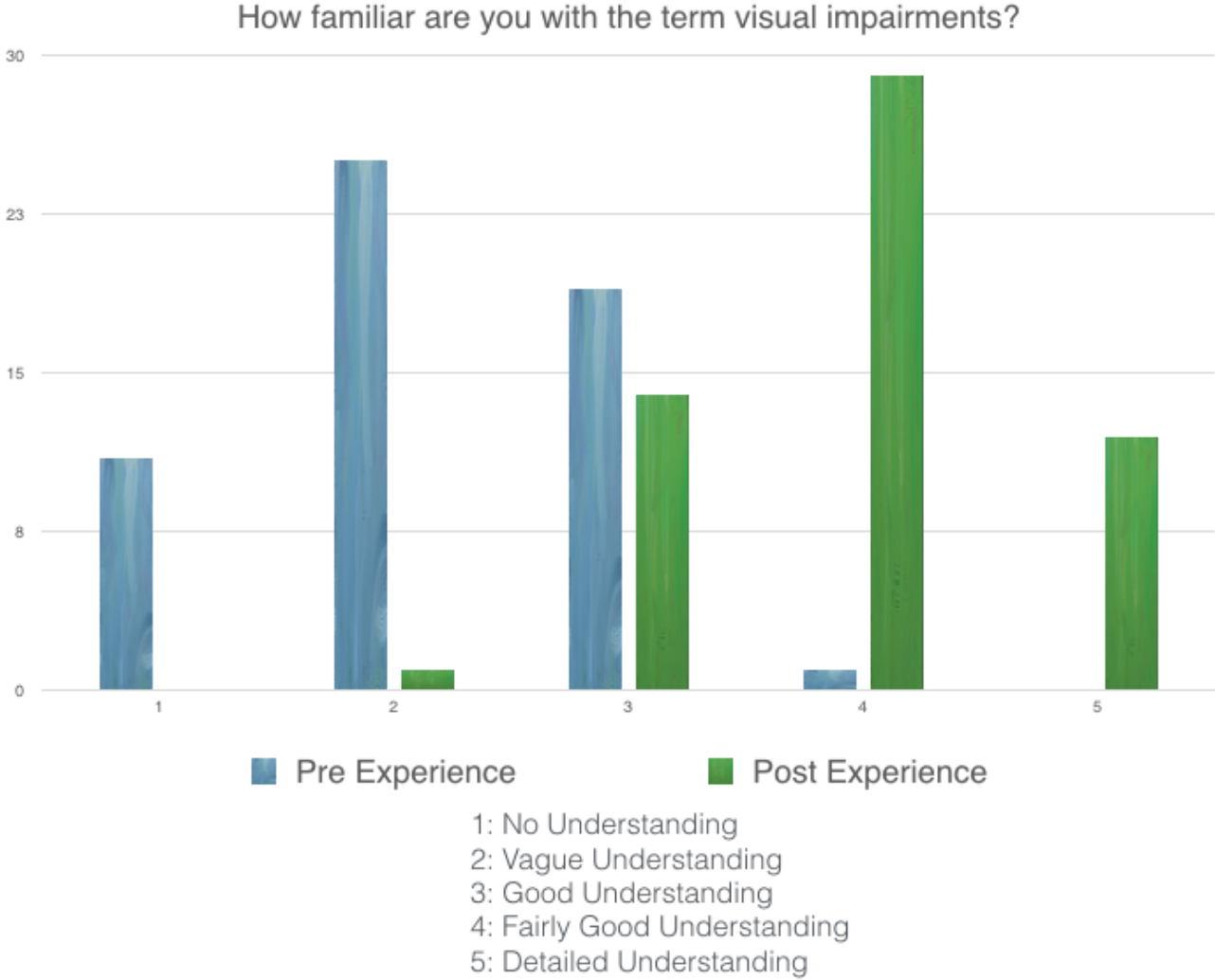


Figure 5.1: Pre and post EmpathEyes experience trend on awareness

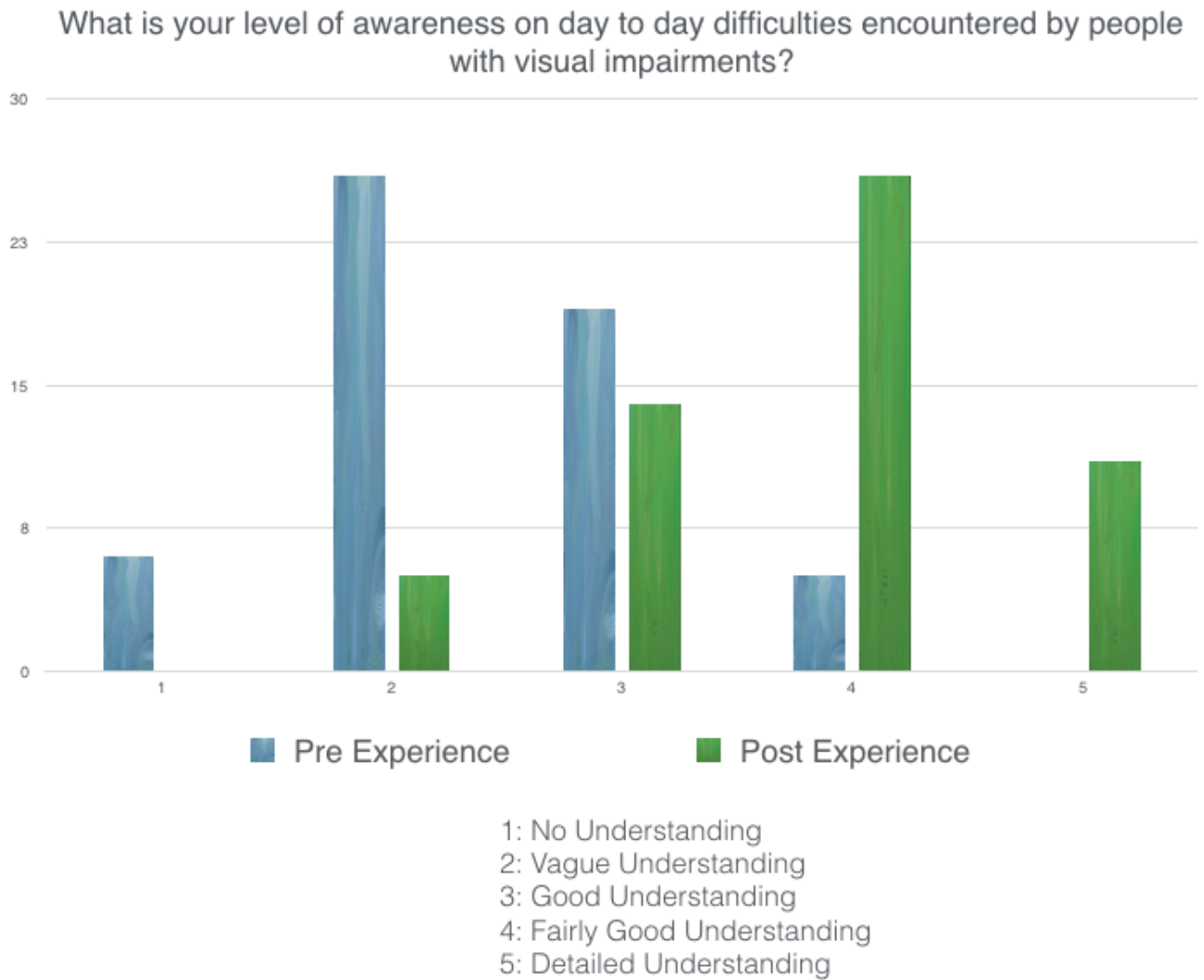


Figure 5.2: Pre and post EmpathEyes experience trend on empathy

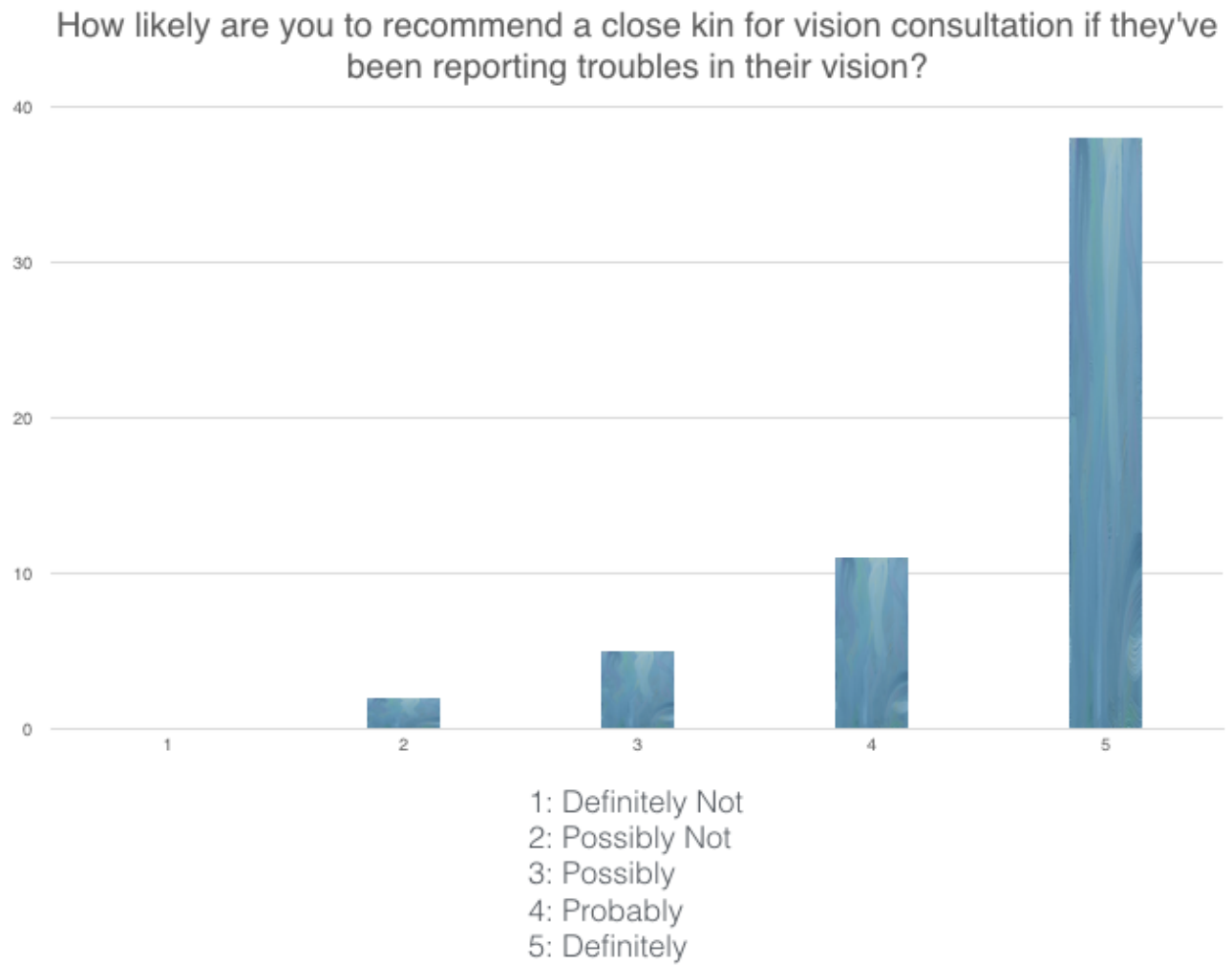


Figure 5.3: Pre and post EmpathEyes experience trend on risk comprehension

Chapter 6

Conclusion

We have presented EmpathEyes, an immersive smartphone based visual impairments simulation system. At this point, many of the common impairments have been simulated and validated for accuracy with domain experts. Informal user experiments demonstrate the potential for use of the system for education of people suffering from impairments, as a tool by consulting optometrists to educate families of people with impairments, as well as a learning tool for students of ophthalmology and optometry.

6.1 Limitations of our Approach

There are many limitations of the current version of EmpathEyes and the validation and user testing process. We list them in no particular order.

1. The use of Google cardboard, though attractive in terms of cost introduces major limitations on the simulated experience: for instance, unlike an Oculus Rift or a HTC Vive, the cardboard based VR only allows 4 degrees of freedom. The user only moves virtually by using some controls.
2. None of these devices can track the user gaze. Hence the user can look around scotoma in their foveal vision or look at the scotoma in their peripheral vision in our simulations. This takes away from the realism and frustration associated with visual impairments involving macula.
3. We can modify the level of impairments that are being simulated. However, we have not yet tried to match an individual's impairment in the simulation. For instance MacAlpine

et.al [17] demonstrate how color vision deficiencies could be dynamically customised for a given individual.

4. These simulations by restricting the experience to synthesized worlds, do not convey the difficulty that VIPs face in the real world.
5. All our user studies have been conducted informally with no rigorous testing processes.

6.2 Future Research

We have released the source code of EmpathEyes to enable a range of research directions to be explored, some of which we outline below.

- Our immediate next step is to undertake systematic user studies to understand the effectiveness and practical feasibility of using EmpathEyes as a tool for creating empathy among care givers of visually impaired. In addition, encouraged by the feedback of an expert optometrist, we will explore the use of EmpathEyes in a clinical counseling set up to educate patients about their specific impairments.
- We are yet to conduct studies of the use of EmpathEyes by people with impairments and evaluate if our simulations match their experience.
- Another follow up study is to utilize the ability of our solution to simulate impairments of varying intensity on a single eye. Users with impairments in one eye can provide feedback by comparing our simulations of impairments viewed with their unaffected eye with the actual experience of their affected eye.
- A related direction is to use data from users with having impairments in one eye so that we can match their vision using our dynamic visual impairment creator. This would help us in creating more realistic visual impairment simulations.
- Another goal of building EmpathEyes is to educate people in the early stages of impairments like glaucoma and their families about the potential ill-effects of non treatment by allowing them to experience the simulated progression of the untreated glaucoma. Similarly, educating people about the debilitating impact of cataract, combined with the information that cataract is treatable can help reduce the number of preventable blindness cases in India. We intend to pursue this direction with support from the two eye care institutions that have supported our work.

- Simulation of the impairments in tandem with real-time eye gaze tracking will substantially increase the fidelity of the simulated experience. There has been substantial work on gaze contingency paradigm and related work on foveated imaging to provide much more realistic experience of visual impairments. Real-time gaze tracking hardware at the rates required to detect saccadic motion of the eye which has been recently demonstrated [4]

In the long term, such immersive simulation experiences could lead to the design of technologies and solutions that help mitigate the impact of the impairments. For instance, people with reduced field of view may be given visual cues on what is beyond their limited field of vision, using techniques like the one used by Xiao and Benko [27] for enhancing the field of view of HMDs with limited field of view.

Bibliography

- [1] Archvizpro vol4 on unity asset store. <https://www.assetstore.unity3d.com/en/#\protect\kern-.1667em\relax/content/73674>. Accessed: 2017-04-30. 7
- [2] Cambridge simulation goggles. <http://www.inclusivedesigntoolkit.com/csg/csg.html>. Accessed: 2017-04-30. 4
- [3] National eye institue. <https://nei.nih.gov/health>. Accessed: 2017-05-03. 8
- [4] Smi eye tracking and foveated rendering. <https://uploadvr.com/smi-hands-on-250hz-eye-tracking/>. Accessed: 2017-05-04. 38
- [5] Actf adesigner. <https://www.eclipse.org/actf/downloads/tools/aDesigner/>. Accessed: 2017-05-04. 5
- [6] Bloom optimized unity documentation. <https://docs.unity3d.com/550/Documentation/Manual/script-BloomOptimized.html>. Accessed: 2017-05-03. 23
- [7] Pre computed realtime gi and light probes. <https://docs.unity3d.com/Manual/GIIntro.html>. Accessed: 2017-05-03. 17
- [8] Visual impairment simulator for microsoft windows. <http://vis.cita.uiuc.edu/downld.php>. Accessed: 2017-05-03. 5
- [9] S. Aballéa and A. Tsuchiya. Seeing and doing: Feasibility study towards valuing visual impairment using simulation spectacles. 2004. 4
- [10] Halim Cagri Ates, Alexander Fiannaca, and Eelke Folmer. Immersive simulation of visual impairments using a wearable see-through display. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '15, pages 225–228, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3305-4. doi: 10.1145/2677199.2680551. URL <http://doi.acm.org/10.1145/2677199.2680551>. 6, 8

BIBLIOGRAPHY

- [11] D Banks and R J Mccrindle. Visual eye disease simulator. 5, 8
- [12] Brandon Cannaday. Convert a coloured image to grayscale. <http://web.archive.org/web/20150430080920/http://tech.pro/tutorial/660/csharp-tutorial-convert-a-color-image-to-grayscale>. Accessed: 2017-05-04. 21
- [13] Joy Goodman-Deane, Patrick M. Langdon, P. John Clarkson, Nicholas HM Caldwell, and Ahmed M. Sarhan. Equipping designers by simulating the effects of visual and hearing impairments. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '07, pages 241–242. ACM, 2007. 5, 8
- [14] Bei Jin, Zhuming Ai, and M. Rasmussen. Simulation of eye disease in virtual reality. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 5128–5131, 2005. 5, 8
- [15] J. Lewis, D. Brown, W. Cranton, and R. Mason. Simulating visual impairments using the unreal engine 3 game engine. In *2011 IEEE 1st International Conference on Serious Games and Applications for Health (SeGAH)*, pages 1–8, Nov 2011. doi: 10.1109/SeGAH.2011.6165430. 5, 6, 8
- [16] Zagar M. and Baggarly S. Low vision simulator goggles in pharmacy education. *American Journal of Pharmaceutical Education*, 74. ISSN 0002-9459. 4
- [17] Rhouri MacAlpine and David R. Flatla. Real-time mobile personalized simulations of impaired colour vision. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '16, pages 181–189, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4124-0. doi: 10.1145/2982142.2982170. URL <http://doi.acm.org/10.1145/2982142.2982170>. 6, 37
- [18] J. Mankoff, H. Fait, and R. Juang. Evaluating accessibility by simulating the experiences of users with vision or motor impairments. *IBM Systems Journal*, 44(3):505–517, 2005. 5, 8
- [19] M Maxhall, A Backman, K Holmlund, L Hedman, B Sondell, and G Bucht. Participants responses to a stroke training simulator, 2004. 5
- [20] John D Mollon, Françoise Viénot, and Hans Brettel. Digital video colourmaps for checking the legibility of displays by dichromats. *Color: Research and applications*, 24(4):243–252, 1999. 21

BIBLIOGRAPHY

- [21] Donatella Pascolini and Silvio Paolo Mariotti. Global estimates of visual impairment: 2010. *British Journal of Ophthalmology*, pages bjophthalmol–2011, 2011. 1
- [22] Daniel Rákos. Optimised gaussian blur. <http://rastergrid.com/blog/2010/09/efficient-gaussian-blur-with-linear-sampling/>. Accessed: 2017-04-30. 17, 20
- [23] GEORGE M. STRATTON. Vision without inversion of the retinal. 2004. 6
- [24] Jani Väyrynen, Ashley Colley, and Jonna Häkkinen. Head mounted display design tool for simulating visual disabilities. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia*, MUM '16, pages 69–73, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4860-7. doi: 10.1145/3012709.3012714. URL <http://doi.acm.org/10.1145/3012709.3012714>. 6, 8
- [25] Margarita Vinnikov, Robert S. Allison, and Dominik Swierad. Real-time simulation of visual defects with gaze-contingent display. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ETRA '08, pages 127–130, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-982-1. doi: 10.1145/1344471.1344504. URL <http://doi.acm.org/10.1145/1344471.1344504>. 8
- [26] Fabian Werfel, Roman Wiche, Jochen Feitsch, and Christian Geiger. Empathizing audiovisual sense impairments: Interactive real-time illustration of diminished sense perception. In *Proceedings of the 7th Augmented Human International Conference 2016*, AH '16, pages 15:1–15:8, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3680-2. doi: 10.1145/2875194.2875226. URL <http://doi.acm.org/10.1145/2875194.2875226>. 6
- [27] Robert Xiao and Hrvoje Benko. Augmenting the field-of-view of head-mounted displays with sparse peripheral displays. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1221–1232. ACM, 2016. 38
- [28] JOEL L. ZUCKERMAN, DAVID MILLER, WILLIAM DYES, and MICHAEL KELLER. Degradation of vision through a simulated cataract. *Investigative Ophthalmology & Visual Science*, 12(3):213, 1973. URL [+http://dx.doi.org/](http://dx.doi.org/). 4