Loughborough University

Department Of Computer Science

Intelligent Computational Techniques and Virtual Environment for Understanding Cerebral Visual Impairment Patients

By

Mr Yahya Al-Fadhili

Submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy of Loughborough University

Supervised by

Prof Paul Chung
Dr Baihua Li
Dr Richard Bowman

January 2018

ABSTRACT

Cerebral Visual Impairment (CVI) is a medical area that concerns the study of the effect of brain damages on the visual field (VF). People with CVI are not able to construct a perfect 3-Dimensional view of what they see through their eyes in their brain. Therefore, they have difficulties in their mobility and behaviours that others find hard to understand due to their visual impairment.

A branch of Artificial Intelligence (AI) is the simulation of behaviour by building computational models that help to explain how people solve problems or why they behave in a certain way. This project describes a novel intelligent system that simulates the navigation problems faced by people with CVI. This will help relatives, friends, and ophthalmologists of CVI patients understand more about their difficulties in navigating their everyday environment.

The navigation simulation system is implemented using the Unity3D game engine. Virtual scenes of different living environments are also created using the Unity modelling software. The vision of the avatar in the virtual environment is implemented using a camera provided by the 3D game engine. Given a visual field chart of a CVI patient with visual impairment, the system automatically creates a filter (mask) that mimics a visual defect and places it in front of the visual field of the avatar. The filters are created by extracting, classifying and converting the symbols of the defected areas in the visual field chart to numerical values and then converted to textures to mask the vision. Each numeric value represents a level of transparency and opacity according to the severity of the visual defect in that region. The filters represent the vision masks.

Unity3D supports physical properties to facilitate the representation of the VF defects into a form of structures of rays. The length of each ray depends on the VF defect's numeric value. Such that, the greater values (means a greater percentage of opacity) represented by short rays in length. While the smaller values (means a greater percentage of transparency) represented by longer rays. The lengths of all rays are representing the vision map (how far the patient can see).

Algorithms for navigation based on the generated rays have been developed to enable the avatar to move around in given virtual environments. The avatar depends on the generated vision map and will exhibit different behaviours to simulate the navigation problem of real patients. The avatar's behaviour of navigation differs from patient to another according to their different defects.

An experiment of navigating virtual environments (scenes) using the HTC Oculus Vive Headset was conducted using different scenarios. The scenarios are designed to use different VF defects within different scenes. The experiment simulates the patient's navigation in virtual environments with static objects (rooms) and in virtual environments with moving objects. The behaviours of the experiment participants' actions (avoid/bump) match the avatar's using the same scenario.

This project has created a system that enables the CVI patient's parents and relatives to aid the understanding what the CVI patient encounter. Besides, it aids the specialists and educators to take into account all the difficulties that the patients experience. Then, is to design and develop appropriate educational programs that can help each individual patient.

Keywords: Artificial Intelligent Based Modeling, Vision Impairment Simulation, Cerebral Visual Impairment, Imitation and Navigation.

List of Published Paper

- Al-Fadhili Y.Q.I., Chung P.W.H., Li B. and Bowman R. (2018). 3D Simulation of Navigation Problem of People with Cerebral Visual Impairment. *Advances in Computational Intelligence Systems 650, Springer.* 265-275. Cardiff, UK.

DEDICATIONS

Every challenging work needs self-efforts as well as the support of those who were very close to our heart.

I dedicate my effort to my loving wife and children

Nadia, Mustafa, Yousif and Maryam

Their affection, love and prays make me able to get such success and honour

And to my sweet

Siblings and friends

Acknowledgements

My greatest appreciation to the Iraqi Government and the Ministry of Higher Education and Scientific Research (MOHSR) for the scholarship I've been given to gain my PhD degree in computer science from Loughborough University, UK.

I would like to express my sincere gratitude for my supervisor Professor Paul Chung for his guidance, great support and kind advice through my PhD research studies. It was a real privilege and an honour for me to share of his exceptional scientific knowledge. I also would like to thank my co-supervisor Dr Baihua Li for her support and efforts to get this thesis to high standard level. Special thanks to my external supervisor Dr Richard Bowman (Great Ormond Street Hospital) for his support and providing details about the cerebral visual impairment.

I'd like also to thank all of my friends and colleagues for their unparalleled support through their endless encouragements during our meetings and discussions.

Many thanks to everyone give supports to get this thesis to light.

For all of them, I say… ***THANK YOU***.

Yahya

Contents

Abstract	i
List of Published Paper	iii
Dedications	Iv
Acknowledgements	V
Contents	vi
List of Figures	X
List of Tables	xiv
List of Abbreviations	xvi
CHAPTER 1: Introduction	**1**
1.1 Introduction	1
1.2 Motivations	3
1.3 Overview of Project	3
1.4 Aim and Objectives	4
1.5 Contributions	5
1.6 Structure of Thesis	6
CHAPTER 2: Visual Impairment	**8**
2.1 Cerebral/Cortical Visual Impairment CVI	8
2.1.1 Overview of CVI	8
2.1.2 Visual System Pathways	11
2.1.3 Special Educational Treatment Required	13
2.2 Brain Damage Detection Tools (MRI scan, CT scan)	14
2.2.1 Magnetic Resonance Imaging (MRI) Scans	15
2.2.2 Computerised Tomography (CT) Scan	16
2.2.3 MRI versus CT in Exploring the Brain	17
2.3 Voxelation	18
2.4 Visual Field Test	19
2.5 Summary and Conclusions	22
CHAPTER 3: Computer Applications In Visual Impact	**23**
3.1 Inventory for Assessment and Classification of CVI	23
3.2 Relationship between Eye Movements and Visual Field	24
3.3 Safe Simulation Environments for Visual Field Tests	26

3.3.1 Driving Hazard Perception 26
3.3.2 Visual Field Distortion During Reading Task 27
3.4 Screen Readers for the Visually Impaired 28
3.5 Wearable Computer Technology 30
3.5.1 OpenGlass Project 30
3.5.2 Badge3D Project 32
3.5.3 Artificial Intelligent Powered Vision 33
3.6 Summary and Conclusions 34
CHAPTER 4: Project Methodology 36
4.1 Introduction 36
4.2 Visual Field Symbols Processing 38
4.3 Vision Masks Design 39
4.4 Projection and Simulation 40
4.5 The Emulation Tool for Projecting the Virtual Scene for Simulation 42
4.6 Summary 45
CHAPTER 5: Visual Field Symbols Processing 47
5.1 Visual Field Chart Processing 47
5.2 Identification of Region of Interest 48
5.3 Visual Symbols Detection 50
5.4 Visual Symbols Extraction 52
5.5 Visual Symbols Classification 54
5.6 Visual Symbols Representation 58
5.7 Summary and Conclusions 61
CHAPTER 6: Mask Design 62
6.1 Mask Design for the Transparency and Blurring Effects 62
6.2 Transparency Effects Applied to the Masks 64
6.3 Summary and Conclusions 66
CHAPTER 7: 3D Visual Field Projection 67
7.1 Vision Mask 67
7.2 Vision Depth 69
7.3 Summary and Conclusions 73

CHAPTER 8: Navigation Modelling Based-On Vision Rays **75**
8.1 Introduction 75
8.2 Vision-Based Situation Assessment 75
8.3 Navigation Decision Making 81
8.4 Scenario 1: Navigation Simulation with Static Objects (Room) 83
8.5 Scenario 2: Navigation with Stairs 86
8.5.1 Upstairs Navigation Simulation 86
8.5.2 Downstairs Navigation Simulation 88
8.6 Scenario 3: Navigation Simulation with Moving Objects 91
8.7 Summary 94
8.8 Conclusions 95
CHAPTER 9: Simulation Results On Navigation Behaviour **97**
9.1 Introduction 97
9.2 Simulation of Normal Vision 98
9.3 Simulation of Right-Side VF Defects 99
9.4 Simulation of Left-Side VF Defects 101
9.5 Simulation of Superior VF Defects 102
9.6 Simulation of Inferior VF Defects 104
9.7 Simulation of the Most Defected VF 105
9.8 Summary and Conclusions 107
CHAPTER 10: Experiment Implementation and Results Discussions **108**
10.1 Introduction 108
10.2 Overview of Experiment 108
10.3 Experiment Design 109
10.3.1 Virtual Scenes 109
10.3.2 Participants 111
10.3.3 Participant in the Scene 112
10.4 Experiment Tasks 113
10.5 Experiments and Data Collection 114
10.6 Experiments Results and Discussions 115
10.7 Summary and Conclusions 126

CHAPTER 11: Summary, Contributions and Future Works **128**
11.1 Summary 128
11.2 Future Works 129
11.2.1 Real CVI Patients (or their Parents/Carers) to Validate the System 129
11.2.2 Using Colour Contrast Along with the Visual Field Defects 130
11.2.3 Speeding up the Algorithm Processing 131
11.2.4 Application for VF projecting 131
REFERENCES **133**

List of Figures

Figure 2.1: The visual system includes the lateral geniculate nucleus, the optic radiation and part of the primary visual cortex 9

Figure 2.2: The main lobes of the brain 11

Figure 2.3: Visual processing centres in the brain. The dorsal and ventral streams 12

Figure 2.4: Dorsal stream damage in MRI scan 12

Figure 2.5: MRI brain scan 15

Figure 2.6: CT brain scan 17

Figure 2.7: Visual field measurements for central, nasal and lateral fields 19

Figure 2.8: Visual field chart 21

Figure 3.1: Driving Hazard Perception Test. (a) Normal visual field, (b) binocular visual field moved according to the person's gaze position (red dot) in the superior VF, (c) inferior VF. 27

Figure 3.2: Visual field distortion during reading task 28

Figure 3.3: Screen reader with Braille display 29

Figure 3.4: Optical Character Recognition (OCR) 30

Figure 3.5: Google Glass 31

Figure 3.6: Horus: AI powered vision 33

Figure 4.1: The implementation workflow 37

Figure 4.2: (a) visual field symbolic pattern and (b) the extracted and classified numerals 38

Figure 4.3: Expanding the 8×8 grid to 15×15 grid to smooth transitions between each two adjacent numbers 39

Figure 4.4: Steps required for designing a vision mask 40

Figure 4.5: Unity3D game engine interface (version 4.0) 43

Figure 5.1: Chart of the Humphrey's Perimetry Visual Field test: (a) Pattern deviation probability map, (b) Meaning of probability symbols 47

Figure 5.2: The required steps to extract numeral values from the DPM 49

Figure 5.3: B/W image conversion to delete the Cartesian coordinates 51

Figure 5.4: Image dilation, holes filling and image opened 52

Figure 5.5: Symbols extraction and symbols' identification 53

Figure 5.6: (a) Pixels scales for the five symbols, (b) Pixels scales 3D graph 54

Figure 5.7: Processes to gain the bag of visual words required for SVM classification 55

Figure 5.8: Training code implementation sequence 57

Figure 5.9: Representing the symbol pattern into 2D array of numerals 58

Figure 5.10: Transparency levels corresponding to the visual scales 59

Figure 5.11: The procedure to expand the array of the visual values 59

Figure 5.12: Adjusting the edges between scales to smooth transitions 60

Figure 5.13: The whole visual field before and after adjustment 60

Figure 6.1: Two layers image to create a mask 63

Figure 6.2: Opacity and Fill percentage for both layers of the mask 63

Figure 6.3: Setting texture's transparency in Unity3D 65

Figure 6.4: Figure 6.4: Assigning transparent shader to objects having transparent textures 66

Figure 7.1: (a) Room design and view point of FPS character (avatar), (b) 2D-array of planes of vision mask with the veil attached in front of the main camera 68

Figure 7.2: VF pattern projection. (a) Projected patterns into Unity, (b) Original patterns 69

Figure 7.3: Vision rays for normal visibility 70

Figure 7.4: Vision rays for deteriorated visibility (back view). (a) Most of the VF are defected, (b) Right side VF defect 71

Figure 7.5: Algorithm for ray instantiation attached to each of the ray's planes 72

Figure 8.1: The hierarchy of the designed project 76

Figure 8.2: Pseudo code for Awake function 77

Figure 8.3: Pseudo code for Start function 78

Figure 8.4: The preparation and calculation algorithm before each updated frame 80

Figure 8.5: Navigation scenarios diagram 83

Figure 8.6: Room scene with static furniture and objects 83

Figure 8.7: Navigation algorithm according to the central and lateral visual 86

field
Figure 8.8: Going upstairs (a) Stairs is seen; (b) Stairs is not seen 87
Figure 8.9: Algorithm for going upstairs 88
Figure 8.10: The proper length of the ray to the platform 89
Figure 8.11: The preprocessing before and at each updated frame for downstairs detection 89
Figure 8.12: Going downstairs (a) Stairs is seen, the inferior rays are greater than the down-sight limit; (b) Stairs is not seen, the inferior rays are not hit and less than the down-sight limit 90
Figure 8.13: Algorithm for getting downstairs 91
Figure 8.14: Algorithm for navigation the central and lateral VF avoiding moving objects 94
Figure 9.1: The views of the avatar of the same room with same furniture arrangement but with different VF patterns. (a) Normal vision, (b) Right-half side defects, (c) Left-half side defects, (d) Superior defects, (e) Inferior defects and (f) Most of the VF defects 97
Figure 9.2: The path and the behaviour of the avatar with clear visual field 98
Figure 9.3: The behaviour of the avatar with clear visual field going up and down the stairs without problem 98
Figure 9.4: The behaviour of the avatar with clear visual field navigating a room scene among moving objects. (a and d) Cube is seen and stop action has been taken, (b and c) capsule is seen and stop action has been taken 99
Figure 9.5: The behaviour of the avatar with right-side VF defects 99
Figure 9.6: The behaviour of the avatar with right-side VF defects going up and down the stairs without problem 100
Figure 9.7: The behaviour of the avatar with right-side VF defects navigating room scene among moving objects. (a) Cube is seen and stop action has been taken, (b) Capsule is seen and stop action has been taken, (c) Capsule is seen and running action has been taken and (d and e) Capsule and cube are not seen and bumped with 100
Figure 9.8: The behaviour of the avatar with VF defects in the left-side 101
Figure 9.9: The behaviour of the avatar with VF defects in the left-side going up and down the stairs without problem 101

Figure 9.10: The behaviour of the avatar with VF defects in the left-side navigating room scene among moving objects. (a) Cube is seen and stop action has been taken, (b) Capsule is seen and stop action has been taken, (c and d) Capsule and cube are not seen and bumped with as they approaching from the unseen left-side 102

Figure 9.11: The behaviour of the avatar with superior VF defects 103

Figure 9.12: The behaviour of the avatar with superior VF defects going up and down the stairs without problem 103

Figure 9.13: The behaviour of the avatar with superior VF defects navigating room scene among moving objects. (a, b, c and d) Cube and capsule are seen and stop action has been taken and (e) Cube is seen and running action has been taken 104

Figure 9.14: The behaviour of the avatar with inferior VF defects 104

Figure 9.15: The behaviour of the avatar with inferior VF defects going up and down the stairs 105

Figure 9.16: The behaviour of the avatar with inferior VF defects navigating room scene among moving objects. (a and d) Cube is not seen and bumped with and (b and c) Capsule is seen and stop action has been taken 105

Figure 9.17: The behaviour of the avatar with most defected VF 106

Figure 9.18: The behaviour of the avatar with most defected VF going up and down the stairs 106

Figure 9.19: The behaviour of the avatar with most defected VF navigating room scene among moving objects. (a and e) Cube is not seen and bumped with (b) Capsule is seen and stop action has been taken, but (c) Capsule is no longer seen and bumped with and (d) Capsule is seen and running action has been taken 107

Figure 10.1: The scenes used in the 3D environment with static and moving objects 110

Figure 10.2: Pseudo code for initializing and adjusting the virtual body of the participants 113

Figure 11.1: The proposed cell phone application. (a) Entering the visual symbols according to the visual field chart and (b) the projection of the visual field onto the main cell phone camera 132

List of Tables

Table 5.1: Computer Vision Toolbox functions supported by MatLab 2015b 56

Table 6.1: The opacity and fill percentage for the masks 64

Table 10.1: The age and sex of the participants 111

Table 10.2: The projected VF patterns onto scenes 114

Table 10.3: Results of Most defected VF simulation in Room1 (Static Objects). Experiment index number in Table 10.2 is 16. The percentage of matching is 99.2% 116

Table 10.4: Results of Normal VF simulation in Room1 (Static Objects). Experiment index number in Table 10.2 is 17. The percentage of matching is 100% 117

Table 10.5: Results of Left VF defects simulation in Room2 (Static Objects). Experiment index number in Table 10.2 is 18. The percentage of matching is 83.3% 118

Table 10.6: Results of Superior VF defects simulation in Room2 (Static Objects). Experiment index number in Table 10.2 is 19. The percentage of matching is 100% 119

Table 10.7: Results of Right VF defects simulation in Room3 (Static Objects). Experiment index number in Table 10.2 is 20. The percentage of matching is 83.3% 120

Table 10.8: Results of Inferior VF defects simulation in Room4 (Static Objects). Experiment index number in Table 10.2 is 21. The percentage of matching is 100% 121

Table 10.9: Results of Inferior VF defects simulation into all rooms (Moving Objects). Experiments index numbers in Table 10.2 are 1, 6 and 11. The percentage of matching is 100% 122

Table 10.10: Results of Left VF defects simulation into all rooms (Moving Objects). Experiments index numbers in Table 10.2 are 2, 7 and 12. The percentage of matching is 100% 122

Table 10.11: Results of Most defected VF simulation into all rooms (Moving Objects). Experiments index numbers in Table 10.2 are 3, 8 and 13. The percentage of matching is 85% 123

Table 10.12: Results of Right VF defects simulation into all rooms (Moving Objects). Experiments index numbers in Table 10.2 are 4, 9 and 14. The percentage of matching is 100% 124

Table 10.13: Results of Superior VF defects simulation into all rooms (Moving Objects). Experiments index numbers in Table 10.2 are 5, 10 and 15. The percentage of matching is 100% 125

Table 10.14: The matching percentage for scenarios with static/moving objects 126

List of Abbreviations

AI	Artificial Intelligent
CP	Cerebral Palsy
CT	Computerised Tomography
CVI	Cerebral Visual Impairment
EEG	Electro-Encephalogram
FPS	First Person Shooter
GPS	Global Positioning System
GPU	Graphics Processing Unit
HPT	Hazard Perception Test
MRI	Magnetic Resonance Image
OCR	Optical Character Recognition
PDM	Probability Deviation Map
PDS	Probability Deviation Symbols
PS	Photo Shop
PVL	Periventricular Leukomalacia
ROI	Region Of Interest
SVM	Support Vector Machine
VBM	Voxel-Based Morphometry
VEP	Visual Evoked Potential
VF	Visual Field
VI	Visual Impairment
VM	Vision Map
VR	Virtual Reality

CHAPTER 1: INTRODUCTION

1.1 Introduction

Visual perception is a function of the eyes and brain together. Therefore, it is the way that human knows and understands the world around them by what they see through the eyes and processed by the brain (Swift, et al., 2008). Sensory information from the eyes is manipulated by the brain to recognise objects, colours, and brightness and the distance of how close or far an object is (Isak, William, 2008).

Cerebral Visual Impairment (CVI) is medically defined as a neurological disorder caused by damage to the occipital lobes and/or to the visual pathways. It is associated with disturbed visual sense because of brain deterioration rather than eye damage (Bernas-Pierce, et al., 1998; Huo, et al., 1999; Dutton, et al., 2004).

CVI affects a person's ability to see and recognise obstacles in their surrounding area. Thus, it affects their ability to move around in their everyday environment. Therefore, patients with CVI often appear to be clumsy because of their inability to see, even though their eyes appear to be perfectly normal. There is the need to demonstrate the difficulties that CVI patients encountered so that their relatives and friends have a better understanding of the problems that they face in their everyday living. If the CVI patient is a child then the better understanding will also help teachers in schools to accommodate for their needs due to their vision deficiency.

Specialist hospitals and clinics use special devices for measuring, locating and evaluating the damages in the brain and the loss in the visual field of CVI patients.

Interventions in the form of educational programs can then be provided to help these patients.

For CVI diagnosis, patients must be subjected to two tests, a brain scan usually using MRI and a visual field test. The MRI scan shows the damage areas in the brain and accordingly the ophthalmologists decide to subject the patient to the visual field (VF) test. This test requires the patients to be sat in front of a device to measure their VF by perceiving flashing lights within their visual field while they are staring at a central fixation point. As a result of this test, a full layout contains a number of numerical and symbolical patterns to clarify the clear and the defected visual areas. The ophthalmologists depend on this test for diagnosing and classifying the degree of the visual damage and thus the educators decide to which educational program the patient will be subjected to.

A major branch of AI is the simulation of behaviour (AISB, 1964). Computational models are built to simulate and explain people's behaviour. This project describes the navigation problem that is faced by people with CVI and develops intelligent algorithms for simulating the behaviour of CVI patients to demonstrate their problems. Furthermore, a virtual environment has also been developed for people to 'experience' the difficulties that CVI patients faced in different scenarios. In the simulation system, an autonomous avatar moves around in a 3D environment mimicking the behaviour of different individuals according to their level of visual impairment. Therefore, given the vision field chart results of an individual, the avatar will move around a given 3D scene avoiding or bumping into objects in the scene as that particular individual would. This serves to highlight the specific problems that individual faces and how the environment may be adapted to suit their needs.

Parents, educators and other people can put on an HTC Vive (virtual reality headset) to experience what it is like to move around in an environment with the same visual impairment (defected VF patterns) as a particular patient.

1.2 Motivations

There is a lack of understanding of the behaviour of CVI patients as they show unexpected ‘clumsiness’ as they move around in an environment. Even though their eyes look perfectly normal and they are facing directly at an object they would still bump into it or trip over it. People might blame them for not paying enough attention to the surroundings, whereas the cause of their unexpected clumsiness is because of the limitations of their vision.

The motivation behind this project is to build a novel 3D simulation system that will aid understanding the navigation behaviour of people with CVI by imitating their movements.

1.3 Overview of Project

This study is to explore how to visualise what CVI patients can and cannot see based on their visual field chart results. Two systems are presented in the work to illustrate the behaviour of navigation of people with CVI.

The first system is designed using 3D modelling software, Unity3D. The VF patterns are projected onto 3D virtual environments to explore the visual view of the patients. The projection shows the clear areas in the VF that the patients can see through them to depend on for their daily actions and what areas are deteriorated and cannot be used to do these actions. The simulation of the behaviour of navigation is done using an avatar within Unity3D. The avatar mimics an individual patient. Intelligent algorithms are applied to manipulate the movement of the avatar. Different scenarios were set to be navigated by the avatar with different VF patterns. These scenarios represent the common places and situations that a patient experiences every day. The simulation can aid the understanding of how a patient makes decisions for the way he navigates.

The second system is built on the previous system but is designed to work with the HTC Oculus Vive headset. The same 3D virtual environments used in the simulation are set to be navigated by real persons. The VF patterns are projected onto the headset to mask the visual field to mimic the visual defects. Then, a healthy person (participant) wears the vive and navigates the virtual environments to feel like the impaired patient and to experience what he is really seeing.

An evaluation study has been carried out to validate the behaviour of the avatar navigation. The behaviour of the person during the navigation will be compared to the avatar navigation for validating the proposed algorithms. Percentage of matching of the behaviour of both the avatar and participant is found to clarify how much the used intelligent algorithms are correct.

1.4 Aim and Objectives

The aim of this research is to aid understanding the behaviour of navigation of people with CVI according to their visual impairment by designing two virtual reality systems. The first is to use an avatar to imitate the CVI patients and demonstrates the visual problems, particularly in navigation. The second is to let the individuals (with healthy vision) can explore what it is like to have similar visual impairment. It enables people with normal vision to experience what these patients encounter during their navigation and why they behave in a certain way.

The objectives of the study are as follows:

1. Convert the visual field chart into a representation that can be used to mimic the CVI patient's visual defects.
2. Design a system that uses 3D modelling software to projecting the visual field representation onto different 3D virtual environments.
3. Use a tool that converts visual defects into a representation of how far the CVI patient can see and explore what he can see and what he cannot see.
4. Write intelligent algorithms to simulating the behaviour of a CVI patient navigation.
5. Design another system that uses HTC Oculus Vive headset to navigate the 3D virtual environments by a real person to experience what CVI patient encounters and then use this simulation to evaluate the results from the first system.

1.5 Contributions

The aim of the study is to build an intelligent system that extracts the visual field symbols from the visual field test chart, and then projects the impaired visual fields onto virtual environments to simulate the navigation problem that is faced by people with CVI. This will help relatives, friends, and ophthalmologists of CVI patients understand more about their difficulties in navigating their everyday environment. The contributions are:

- The study proposed a framework using 3D virtual environments to demonstrate the visual problems that people with CVI face, particularly in navigation.

- A virtual reality system had been proposed as well to let individuals can explore what it is like to have similar visual impairment, and to validate the previous framework to show the consistent between the two systems.

1.6 Structure of Thesis

The rest of thesis structured as follows:

- Chapter two is a review of the human visual system. It also illustrates what the visual system consists of, and focuses on defining and explaining the CVI causes, influences and diagnostic tests.
- Chapter three describes related computer applications and previous works that aim to help and aid people with impaired vision or are blind.
- Chapter four describes the proposed methodology to achieve the project objectives. The methodology consists of three main steps:
  1) Pre-processing the VF symbolic pattern to extract the vision probability symbols and converting them into numerals.
  2) Project the pattern onto visualised 3D world using a vision mask.
  3) Use the vision mask for simulating the behaviour of navigation.
- Chapter five explains the processes for extracting, classifying and categorizing the vision probability symbols in the VF chart and converting them into numeric values.
- Chapter six describes the steps required to convert the deterioration levels of the visual symbols to masks. The masks are designed using textures with specific transparency and opacity to represent which parts of the visual field a patient can or cannot see. This chapter also explains the main 3D engine for projection and simulation.

- Chapter seven explains the projection of the VF masks onto a 3D virtual world. It also illustrates how the VF numeral values are converted into rays and how calculations are carried out for VF simulation.
- Chapter eight explains the use of the instantiated rays in navigation. The calculations to assess the situation of the visual perception are discussed in details.
- Chapter nine provides examples of avatar simulation. The 3D scenarios for testing the avatar's navigation behaviour are described. Details of different visual patterns of deficiencies that are used to represent different visual impairments are given.
- Chapter ten describes the experimental setup for validating the autonomous avatar's behaviour. The behaviours of the participants and the avatar are compared and the results are presented and analysed.
- Chapter eleven states the overall conclusions and gives suggestions for future work.

CHAPTER 2: VISUAL IMPAIRMENT

"Due to improved medical care, children with severe brain insults have increasingly begun to survive over the last 30 years. The vast majority have tended to have severe multiple disabilities, including a variety of learning difficulties. A minority of them also has a permanent visual loss, but normal or minimally abnormal eye examinations.

Traditionally, educators for the visually impaired assisted only those whose eye conditions were associated with visual loss (reduced acuity). Now it has become necessary to offer services for those whose visual loss is due to brain damage. Thus, the definition of CVI was born."

Dr James E. Jan, 1993

2.1 Cerebral/Cortical Visual Impairment CVI

2.1.1 Overview of CVI

Cerebral visual impairment (CVI) is medically defined as a neurological disorder caused by damage to the occipital lobes and/or to the visual pathways. CVI is associated with disturbed visual sense because of the brain deterioration rather than eye damage (Bernas-Pierce, et al., 1998; Huo, et al., 1999; Dutton, et al., 2004; Swift, et al., 2008). In another word, with CVI the eyes look perfectly fine but the brain does not interpret what the eyes see. Therefore, it may be misdiagnosed by ophthalmologists as the structure of the eye is normal. Yet, there are regular behavioural features which can be identified using good observation of behaviour and testing, and a structured history taking, which will all lead to the diagnosis (Dutton, et al., 2004). The process of vision –from the classic view– takes place as a result of the visual system. This includes the anterior visual pathways, lateral geniculate bodies (a pair of neural structure that serves as a processing station in the major pathway from the retina to the cerebral cortex), the optic radiations, and the occipital cortices where the picture is 'seen' (see Figure 2.1). Therefore, any slight damage to those components will lead to cause visual deterioration like reduction of visual acuity and the loss of visual field (Dutton, et al., 2006; Dutton, 2009; Macintyre-Beon, et al., 2010).

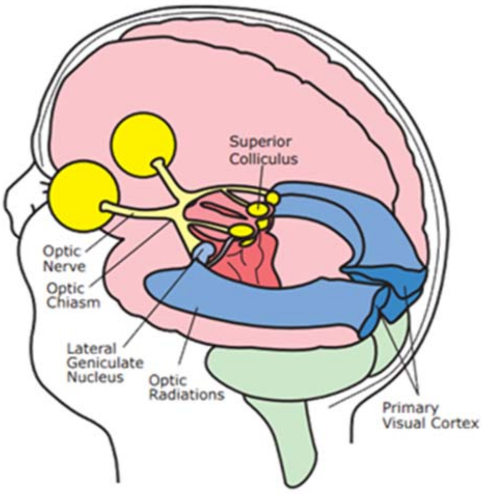


Figure 2.1: The visual system includes the lateral geniculate nucleus, the optic radiation and part of the primary visual cortex

(Souse, 2001)

CVI is common in children with cerebral palsy (CP) (Good, et al., 1994). According to Huo et al. (1999), perinatal hypoxia, cerebral vascular accident and meningitis are can also cause CVI.

The impaired vision can range from mild visual impairment to blindness. It depends on the onset time, as well as the strength and location of the injury. Therefore, CVI can simply be defined as a condition that the visual systems of the brain are not able to understand or interpret what the eyes see (Bernas-Pierce, et al., 1998).

The visual system is poor in new-born children and gradually develops to finer levels during the early years after childbirth. *Visual acuity* limits the number of the details that can be seen. *Contrast sensitivity* limits how objects can be faint before these objects become invisible. *Visual fields* limit the extent of the region that a person can see when looking straight ahead. *The speed of visual processing* limits the speed of a moving object that can be seen (Dutton, 2008).

Many studies that concern with measuring visual development indicate that the contrast sensitivity and visual acuity are mature by the age of 5 to 6 years. This was measured psychophysically.

The visual acuity test is a routine eye test called the "Snellen". It involves reading letters from a chart and these letters become gradually smaller. Snellen outcome consists of two numbers. The first number indicates how far the person being tested can see. The second number represents how far a person with normal vision can read the chart.

The contrast sensitivity is an important measurement for CVI, particularly when there is a reduction in contrast between objects and their background, in situations like low light, fog or bright light. The patient may have 20/20 visual acuity but may have reduced contrast sensitivity so cannot see well at all. Pelli-Robson contrast sensitivity chart is the most widely used for testing contrast sensitivity. It consists of capital letters with horizontal lines. The contrast between lines for the different letters is different. The high contrast letters represent low spatial frequencies and letters with low contrast represent high spatial frequencies.

The visual field test can be done using several different tests. One way is for the person being tested look straight ahead at a device that has lights flashing in different parts of the vision area. The person presses a button whenever they see a light flashing. This shows the extent of their visual field. Another way is for the person being tested to follow a moving object across their field. The person indicates the onset of seeing the object and when it no longer can be seen (Segre, 2014; Lusby, 2013).

2.1.2 Visual System Pathways

Basically, the visual system consists of two functional pathways: the perceptual and action pathways. The perceptual pathway is the recognition of route, object and person. It passes from the occipital lobes to the temporal lobes and is called the *ventral stream*. The action pathway is for guiding actions. It passes from occipital lobes to the posterior parietal lobes and is called the *dorsal stream*, as illustrated in Figure 2.2 (Ungerleider, et al., 1982; Goodale, et al., 1992).

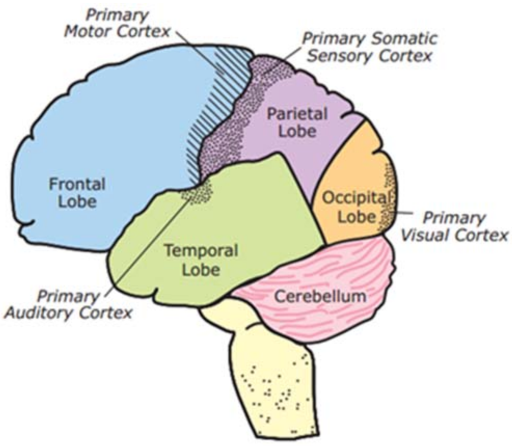


Figure 2.2: The main lobes of the brain

(Palmer, 1999)

The ventral stream is also called the 'what' pathway where the 'image libraries' are stored. The recognition is achieved by matching what is seen with what is stored in the library. It is essential to perceive the identity and the properties of the objects such as colour, shape, texture and orientation. The dorsal stream is described as the 'where' pathway which assesses the whole visual scene and facilitates the immediate visual guidance of movement through the scene (see Figure 2.3). The dorsal stream is unconscious, automatic and immediate, which is not memory based. Damages to this pathway impair visual guidance (Macintyre-Beon, et al., 2010; Dutton, 2009; Goodale, et al., 1992).

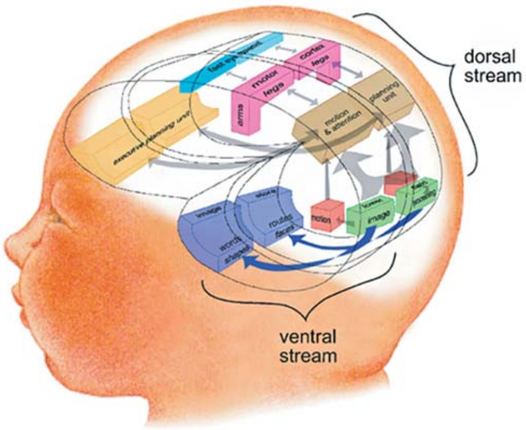


Figure 2.3: Visual processing centres in the brain. The dorsal and ventral streams
(Dutton, 2003)

Dorsal stream dysfunction is common. Impaired visual guidance of movement may be understood as 'clumsiness' (Dutton, 2008).

Both visual processing pathways at a higher level have their limitations. Dorsal stream function limits the ease of seeing something in a visually crowded environment and limits the accuracy of visually guided movement. In the other hand, the ventral stream function limits the recognition of what person can see (people, objects, environment during route finding). Figure 2.4 shows an MRI scan for dorsal stream damage.

Figure 2.4: Dorsal stream damage in MRI scan
(Dutton, 2008)

When navigating through the 3D world, the scene of what a person can see is processed by the occipital lobes at the back of the brain. The process is then broken down into two components; vision for recognition and vision for action (see Figure 2.3). The vision for recognition takes place in the temporal lobes at the bottom of the brain and represents the ventral stream. The vision for action takes place in the parietal lobes at the top of the brain and represents the dorsal stream in which the scene of the visual world is passed to the part of the brain responsible for moving the body through this scene (Dutton, 2008).

People with visual impairment have problems to make accurate movements through 3D visual space. This is because the visual pathways which pass the details of the scene to the part in charge of motion, the dorsal stream, have been damaged. Dorsal stream sends the information to the posterior parietal visual cortex where the whole scene is processed and a part of the scene is chosen.

Safe, soft and quiet areas with few people around can provide a stimulating environment for people with vision problem to learn to move through 3D space without injury. This provides a chance to learn skills and develop self-confidence (Dutton, 2008).

2.1.3 Special Educational Treatment Required

CVI is considered as educational and training recovery disease. The increasing number of CVI patients prompted the need to provide special education in the field of vision. Educational intervention programs have been designed and developed in specialist institutions, hospitals and clinics. These intervention programs present visual information in highly controlled environments.

CVI patients have individual differences in terms of the degree of impairment due to the amount of damage in the brain and the amount of damage in the visual system. Thus, each patient with CVI must have their own functional assessment. This assessment will guide a tailored treatment for each patient. A single approach will not be for all.

CVI may recover gradually and may take months to years. In some cases, a partial return of vision has been documented. The pattern of recovery typically follows the improvement of light, colour and form perception, and the improvement in visual acuity (Smith, 2007).

Very few patients recover completely after being diagnosed with CVI but most of them show some recovery. Usually, the most dramatic improvement happens in the first two years after the diagnosis. Improvement is related to a patient's brain plasticity, development and degree of neurological damage (Bernas-pierce, et al., 1998).

In summary, by realising and understanding the impact of CVI has on a patient's behaviour, educators and carers can find suitable ways to interact and communicate with or to arrange their environment to make them safe.

2.2 Brain Damage Detection Tools (MRI scan, CT scan)

In healthcare, imaging has become an essential tool. Investigators demand higher sensitivity and high-resolution images for analysis Ultrasonography, Computerised Tomography (CT scan) or Magnetic Resonance Imaging (MRI) and apply them to different medical areas.

CVI detection can be carried out in several ways. A form of computerised Electro-Encephalogram (EEG) called Visual Evoked Potential (VEP) can detect the electrical activity, or lack of activity, in the brain. VEP provides information related to the functioning of the visual occipital cortex when the person is exposed to visual stimuli. Imaging techniques, such as Ultrasonography, CT scan or MRI scan can identify areas of affected brain tissue (Groenveld, 2004; Harrell, 2004).

2.2.1 Magnetic Resonance Imaging (MRI) Scans

MRI uses magnetic field supported with radio waves to create images of the body or the brain.

The mechanism of MRI is to create a strong magnetic field by passing an electric current through the wire loops. This will make the protons in the body to be aligned. Meanwhile, the coils in the magnet send and receive radio waves. These radio waves are observed by the aligned protons and make them spin. The energy is released as signals and picked up by the coil. This information is then sent to a computer which processes and converts all the signals into an image (see Figure 2.5). The final representation of the area being examined is a 3-D image (McRobbie, 2007; Bello, 2013; Kalapurayil, 2013).

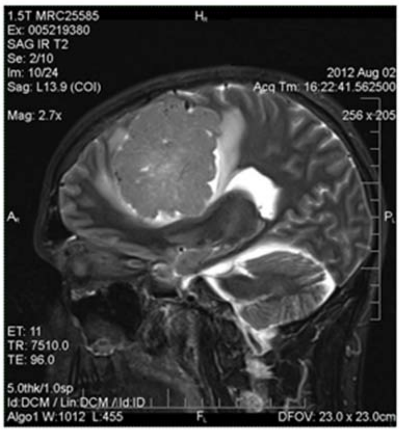


Figure 2.5: MRI brain scan

Peter Ganza's Blog (20 March 2013)

Brain MRI scans can identify brain lesions associated with CVI. For example, they can be interpreted according to the severe lesions of the optic radiations and the lesions of the visual cortex. The visual acuity can be statically correlated with MRI grading. Cioni et al. (1996) found that the damage at the optic radiations is better predictable. In addition, the early detection of abnormal MRI findings at the optic radiations may indicate which infants should receive visual assessment, specific rehabilitation and educational management.

MRIs can show the presence of periventricular leukomalacia (PVL) (a form of white-matter brain injury) indicate the impossibility for the full visual recovery. MRIs with normal visual cortex and absence of PVL indicate better visual outcome (Casteels, et al., 1997).

2.2.2 Computerised Tomography (CT) Scans

Computerised tomography (CT) scanner is a useful tool that assists medical diagnosis which uses a computer that takes images of the structures or organs inside the human body. Every image represents a single cross-section (slice) through a 3D object. The CT scanner generates a 3D image of the body by using the digital geometry processing. This 3D image is constructed by taking several 2D X-ray images from many angles of the same area.

The mechanism of the CT scan is to release a group of beams of X-ray from a tube rotate around the body. An X-ray detector inside the CT receives the rays on the opposite side of the body and produces an image of the scan using a computer. CT scan gives clear images of muscles, bones, organs and blood vessels. It is commonly used for the brain to determine the cause of a stroke and head injuries (see Figure 2.6) (Nordqvist, 2014; Cancer Research UK, 2013; Herman, 2009).

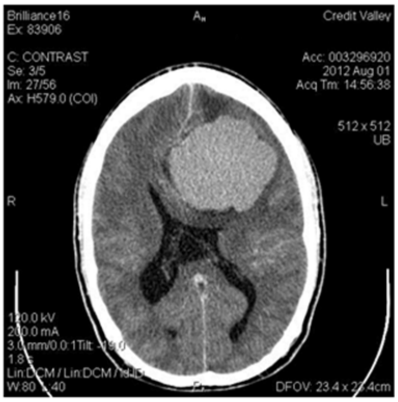


Figure 2.6: CT brain scan

Peter Ganza's Blog (20 March 2013)

Brian CT scan is very effective for diagnosing lesions that are related to CVI. According to Schenk-Rootlieb et al. (1994), lesions in the visual areas are found using CT scans of cerebral palsy (CP) patients with CVI. The study found the abnormalities of white matter near to the lateral ventricles and under the visual cortex.

2.2.3 MRI versus CT in Exploring the Brain

MRI and CT are complementary imaging technologies; each has their own advantages and limitations for different applications. In general, CT is more widely used than MRI because it is faster, cheaper, less sensitive to patient motion during the examination and suitable for emergency rooms. CT is more accurate for detecting body cancer, metal bodies and calcification. It is good for imaging skeletal system, bone structure (neck, shoulders and spine), the head (brain and its vessels, eyes, inner ear, and sinuses) and chest (heart and lungs).On the other hand, repeated CT scan can be harmful and even cause cancer because it uses radiation.

The advantage of preferring MRI over CT is that it does not use ionising radiation, thus it is safer for patients that require multiple imaging examinations. MRI is suitable for examining soft tissues such as spinal cord and brain tumours because it gives more details and differentiates between different kinds of soft tissues (Sanjeev, 2012; Hess, 2012).

2.3 Voxelation

The high-resolution of the 3-dimensional images obtained by MRI and CT scan leads to use a new approach for diagnosing many diseases and impairments, including CVI. This new approach uses the principle of volume pixel instead of normal pixel.

Two-dimensional images are defined by pixels (picture element) using x and y coordinates. Three-dimensional images use the z coordinate for depth. The smallest distinguishable box-shaped part in the three-dimensional space is called voxel (volume pixel/volume element).

3-D software and images defined by voxels are now used by researchers and medical physicians to analyse and view X-Rays, CT scans, and MRI scans to explore the inside of the human body (Rouse, 2007; Kaufman, et al., 1993).

Voxelation permits high throughput of multiple volumetric images of the brain. These images are similar to the images of the biomedical imaging systems using the analysis of registered spatial voxels (cubes).

Medical scientists began to build voxel models that represent computational models of human anatomy using CT and MRI high resolution cross-sectional digital images of internal organs and tissues by using voxel-based morphometry (Caon, 2004; Ashburner J., 2000).

Voxel-based morphometry (VBM) is an approach for revealing the differences in the density or volume of brain tissue. Pathological changes in the brain would result in brain tissue cell loss, or atrophy, which can be recognised by structural MRI (Jennifer, 2009; Watkins, et al., 2001).

2.4 Visual Field Test

The visual field is the space or the area in which objects can be seen by the peripheral vision (side) while staring ahead at a fixed central point. For each eye, the visual field (monocular) includes 30 degrees of the inner vision and central fixation, and extends to 95 to 100 degrees for lateral/peripheral outward vision away from the nose (temporally), and 60 degrees for the medial vision toward the nose (nasally), and 60 and 75 degrees for upward (superiorly) and downward (inferiorly) vision respectively (see Figure 2.7) (Walker, et al., 1990; Lusby, 2013).

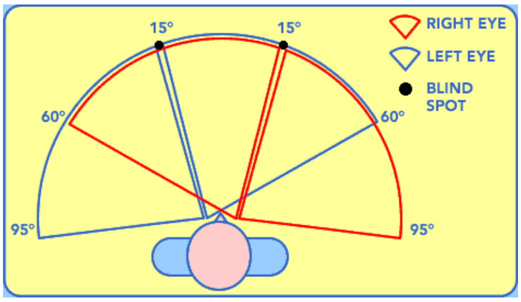


Figure 2.7: Visual field measurements for central, nasal and lateral fields

Image adapted from www.testvision.org

The visual field test is a dysfunction detection of the central and peripheral vision. Visual field impairment may be caused by brain tumours, stroke, glaucoma or neurological deficits. The test can be done clinically by keeping the eye gazing at a fixed point while presenting objects at different locations within the visual field.

A perimeter is an instrument often used for the visual field test. There are two kinds of Perimetry, kinetic and static. The kinetic perimeter is where the examiner moves light points from the lateral side toward the nasal side in the field of vision. The test is repeated for different light brightness. It is suitable for cognitive impairment patient. The static perimeter uses fixed light points that flash on a white screen. The brightness of the lights gradually increased until the threshold is set. In both methods, the patient responds by pressing a button to indicate that they have a light. Visual field measurement is one of the most important steps for CVI and other optical diseases and disorders diagnosis. A perimeter will be used for the visual field test after a brain scan had been done for lesions and injuries detection.

The visual test results appeared as a chart with basic information. The computer checks the accuracy of the test according to the percentage of error. Then the patient’s sensitivity numerical results (patient numerical pattern) are compared with normal results of non-patient in the same age group (normal numerical pattern) and statistics are calculated. Finally, the computer represents each numerical pattern for each location of the visual field by a symbol. The darkness of the symbol at a location depends on the difference the patient’s results and normal results. All patterns are of a particular shape of "diamond" as the VF in the four corners are not seen and therefore are not subjected to the test (see Figure 2.8).

According to section 2.1.3, the VF can be improved by special educational treatments by presenting the visual information in highly controlled environments and after a long time of treatment there may be some improvements. These improvements can be tested and seen by subjecting the patient to VF tests. These tests can be carried out regularly so that the changes in the visually impaired areas can be identified.

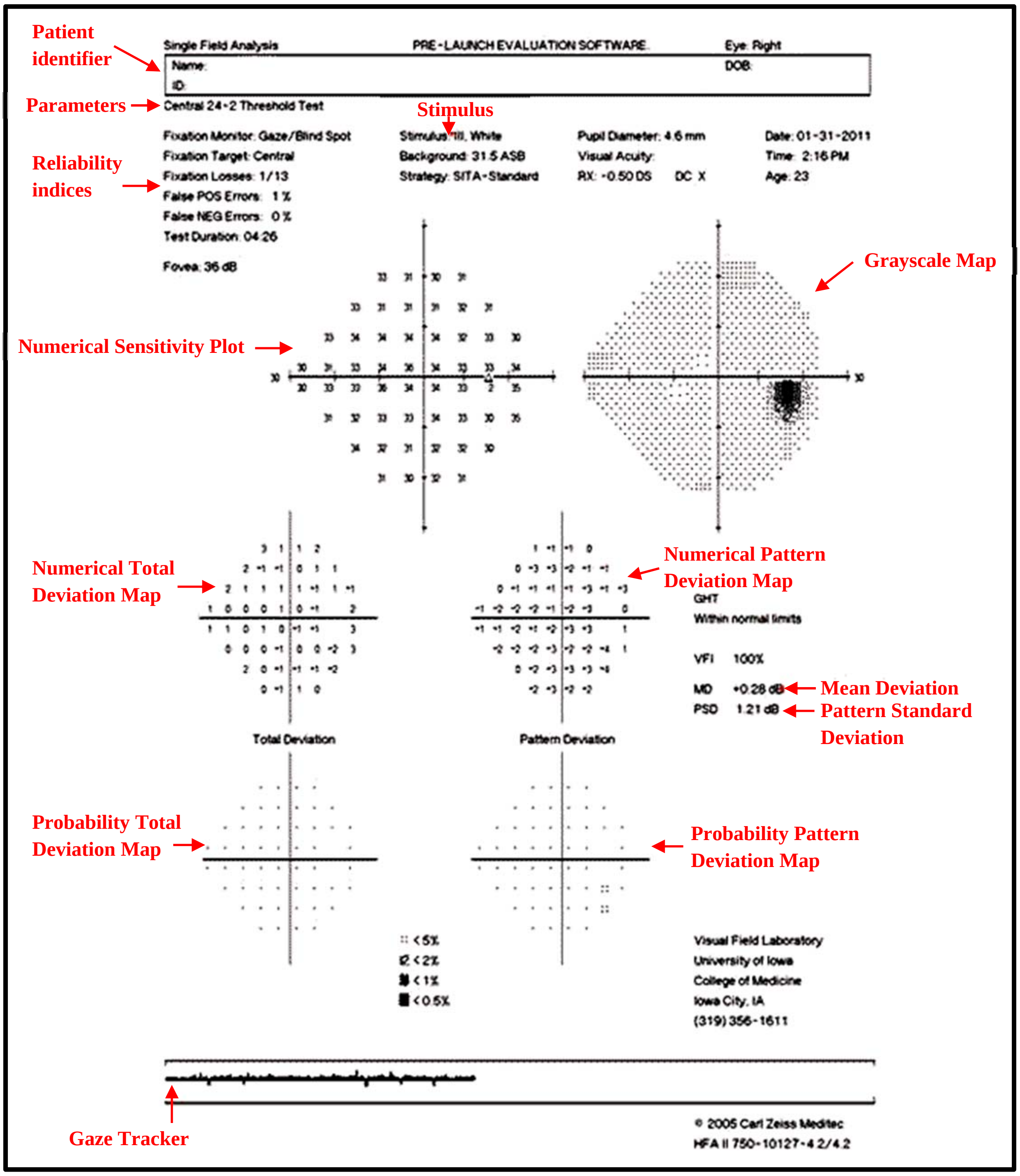


Figure 2.8: Visual field chart

(Carroll, et al., 2013)

2.5 Summary and Conclusions

Visual impairment is when a person lost their sight which cannot be corrected using glasses. With CVI, the eyes and the optic nerves may be perfectly normal but it is the processing of the information by the brain that is at fault.

Brain scans using CT or MRI will show where the damaged brain tissues are. Taking measurements of the visual field using a perimeter and plotting the results using a visual field chart will show which parts of the visual field are affected.

CVI patients need educational treatment rather than medical treatment to help them gain confidence in their living environment.

However, CVI patients are often misunderstood as their eyes look perfectly normal and others do not appreciate the difficulty they have. Therefore, this project aims to develop tools that will help people understand more about CVI and why CVI patients often appear to be clumsy.

The next chapter reviews literature that describes computer applications that help people with visual impairments.

CHAPTER 3: COMPUTER APPLICATIONS IN VISUAL IMPACT

"Our visual field as an island of vision or hill of vision surrounded by a sea of blindness"

(Grzybowski, 2009)

Since the diagnosis of CVI had become known, ophthalmologists, specialists, researchers, and many institutions have given attention and concern for its assessments and treatments. Several computer programs have been suggested and implemented to measure and evaluate the degree of the patient’s disability. There are also programs that were built to help patients face their disability and rely on themselves to perform their daily tasks as much as possible. Some of these programs are described below.

3.1 Inventory for Assessment and Classification of CVI

Several studies have provided evidence that CVI is possibly a contributor to cognitive and intellectual dysfunction in prematurely born and neonatal children.

Specialists in the Great Ormond Street Hospital and the School of Health and Life Science, Glasgow University, have been making significant progress in the field of assessment and classification of children with CVI who attend mainstream schools. The team worked to characterise children behaviours and classify impaired children according to the degree and type of impairment. The studies relied on eye examination (visual acuity and visual field tests), hypotheses, and statistical analysis methods to perform this classification.

One of their computerised studies that consisted of two groups (study and control) is based on a structured clinical history using questionnaire. The questions were grouped into 7 groups according to the kind of evidence of VF impairment. All patients also go through four visual perception tests. These tests can be explained as: *Visual closer test* that finds the ability to percept the whole picture when given a partial picture, *Global form test* that measures the ability to recognize and grab an image within noise, *Global motion test* which measures the ability to realize and grab a moving image from noise, and *Face processing test* that assesses the ability of matching a photograph of a face with other faces of Sterling coins. All these tests had been done by monitoring images on a computer display.

The inventory questionnaire was answered by the parents on behalf of the children. The questions asked about the frequency of encountered difficulties. The responses are based on a scale of 5 psychometric items: *never*, *seldom*, *sometimes*, *often* and *always*. Two clusters were created as a binary score with never and seldom being negative and the other three items being positive. The study used statistical analyses to identify the differences between these clusters and the control group.

The outcome of the study emphasised that the inventory provides a possible way to identify and characterise CVI conditions.

Simple strategies were suggested to the parents at home and for teachers in schools to address the dysfunctionality for impaired children. A program was built in Microsoft Excel to create the list of suggested strategies automatically based on the answers to the inventory (Macintyre-Beon, et al, 2010).

3.2 Relationship between Eye Movements and Visual Field

The applied vision centre at City University in London has studied glaucoma patients. The patient is visually stimulated by images displayed on a computer screen. The eye tracker then monitors and records the saccadic eye movements of the patient. A study was carried out to find the relationship between impaired visual search and eye movements. The patient would search for specific objects among a series of target objects in photographs of real-world scenes displayed on a computer screen. The eye tracker records all saccades that the patient makes during their search. The study found that the average number of saccades in patients was significantly lower than the control group (Smith, et al, 2012).

Another study focused on saccadic eye movements and face recognition performance. The study built on the Cambridge Face Memory Test which assesses difficulties in recognizing faces (Pinola, 2011). The patient searches for a specific person's face among many faces in a photograph displayed on an eye tracker computer screen. The study found that patients with visual field disorders made larger saccades associated to their performance of face recognition within the 10 degrees of their central visual field (Glen, et al, 2013).

Smith et al (2014) assessed the reading performance among patients with glaucomatous and those with asymmetrical visual field loss. The study examined the reading duration and saccadic eye movements when reading short texts using an eye tracker system. Patients' worse eyes are distinguished from their better eyes according to their Humphrey VF test and one eye was randomly selected. The patients were seated with a headrest in front of the computer screen and asked to read 50 different texts as quickly and accurately as possible, one at a time. The study determined the exact points when the reading began and finished. The system classified and recorded three types of saccades: saccades from left to right (forward), from right to left (backwards/ regression), or between lines (line change). The system concluded that patients with worse eye deficiencies take longer reading duration than their better eye. Significantly patients' backwards saccades and unknown eye movements were recorded.

3.3 Safe Simulation Environments for Visual Field Tests

CVI patients have many limitations going about their daily activities because of their visual defects. Therefore, researchers try to simulate some of the everyday environment for CVI patients to try in a safe way and monitor their performance.

3.3.1 Driving Hazard Perception

Car accidents are related to the degree of the binocular visual field loss. A study employed a novel gaze-contingent display is used for Hazard Perception Test (HPT). A set of films of driving scenarios were used to measure the rate of responses for detecting hazards using a computer-based test. For the study, an eye tracker was used. The displayed image is distorted to reflect a visual field defect. The distortion was done for the superior and inferior visual field respectively. Figure 3.1a shows the normal visual field. Figures 3.1b and 3.1c show the distortion of superior and inferior visual fields respectively.

The normal drivers keep looking ahead as they fixate their sight at the centre of the view. Meanwhile, the perception of the peripheral visual field helps the drivers in both superior/inferior and lateral vision. As the test focused on superior and inferior deteriorated visual field, the study concluded that inferior visual field defects were less harmful for hazard detection than superior defects (Glen, et al, 2014).

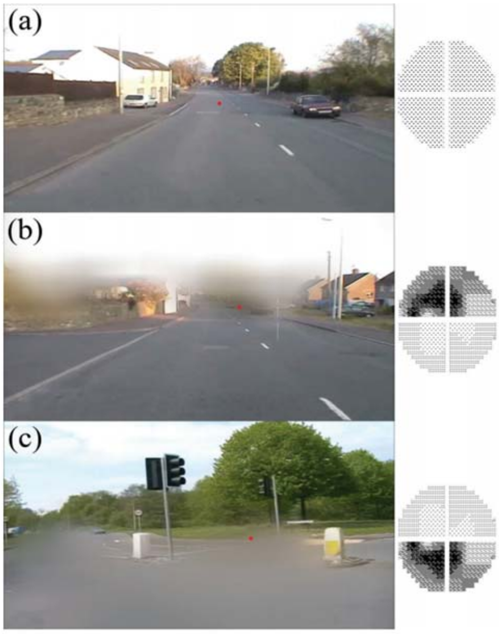


Figure 3.1: Driving Hazard Perception Test. (a) Normal visual field, (b) binocular visual field moved according to the person's gaze position (red dot) in the superior VF, (c) inferior VF.

(Glen, et al, 2014)

3.3.2 Visual Field Distortion during Reading Task

A daily task performed by CVI patients is reading. The patients tend to use their central visual field in order to gaze better at the line and the word being read. A study was carried out to measure and simulate the reading performance of glaucoma patients. This work was done by the same group that carried out the driving hazard perception test described above (Glen, et al., 2016). The eye tracker was used to position distortions in patient's visual field as he reading a book. The eye movements were recorded as a video file. The gazing position was indicated by a red point over each word being read.

The video shows the gazing position moving from word to word as the reader moves their head from left to right. It also shows the eye moving from the end of one line to the beginning of the next line with the head movement as well. The blur of the peripheral areas in the scene indicates the corresponding defects in the visual field areas (see Figure 3.2). The study indicates how the patients depended and focused on their central visual field during reading.

The study ignored eye saccades as it was concerned with central field gazing. Therefore, only a red dot was used to show the patient’s gaze position.

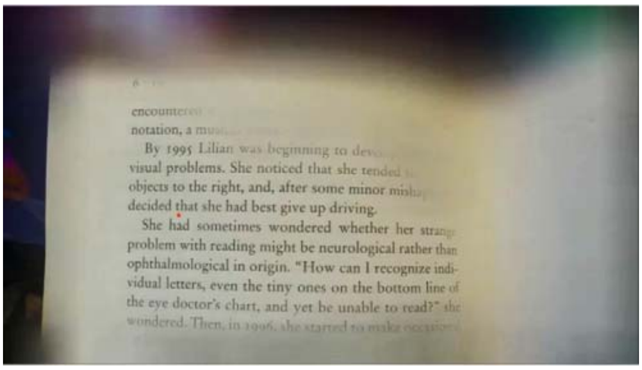


Figure 3.2: Visual field distortion during reading task

(Crabb D., 2014)

3.4 Screen Readers for the Visually Impaired

Visually impaired people – include low vision and blindness – do use computers. Therefore computer output devices need to be adapted to their needs. The blind and visually impaired require the computer output systems to be adapted rather than the input systems as they have no problem using the keyboard or giving voice commands.

Access to the output of a computer by the visually impaired (VI) and blind is normally by using a screen reader, speech synthesiser or braille display. Fonts and images are usually enlarged using screen magnification programs to help the visually impaired.

A screen reader in addition to speaking aloud what is on the monitor screen, it also speaks out each key being pressed to provide an auditory cue for application navigation. A braille display contains a row of cells and each cell forms a Braille character with a series of movable pins that rise or fall depending on the character being represented (see Figure 3.3) (Rouse M., 2015).

Figure 3.3: Screen reader with Braille display

(Greco L., 2015)

Some screen readers combine screen reading, screen magnification and braille output in one application with support for braille display. Other kinds of support include image caption reading and options for web page navigation. Example of such screen reader is *COBRA* and *SuperNova* (Watson L., 2005).

People with visual impairment may read documents that are not stored digitally on a computer by using Optical Character Recognition (OCR) technology. Some OCR software requires a scanner to capture the text and convert it into a digital form. Other OCR software uses a camera that is connected to the computer for the scanning process. Popular OCR software is Kurzweil 1000 which supports both options (see Figure 3.4) (Leibs A., 2015).

Figure 3.4: Optical Character Recognition (OCR)

(Kurzweil, 2017)

3.5 Wearable Computer Technology

It is estimated that there are over 30 million blind or partially sighted people in Europe (EBU, 2010). Due to the business of the urban environments, traditional aids such as white canes and guide dogs are not very effective for people with visual impairment. On the other hand, advancement in technology has the potential to help people with visual impairment to live independently.

Wearable computer technology is one that has received considerable attention.

3.5.1 OpenGlass Project

A couple of computer science PhD students (Brandyn White and Andrew Miller) who worked at the Dapper Vision Company pioneered a project that use Google Glass (see Figure 3.5) and crowdsourcing to develop applications that help to increase the perception and understanding of the environments for the blind and visually impaired users. Google Glass is a wearable computer that presents information on an optical display mounted on a pair of glasses. The initial purpose for it was to interact with the Internet via voice commands and to display the requested information on the optical device. It is like a hands-free smartphone. The project turned it into "an information vehicle for the visually impaired" (Owano N., 2013; Rouse M., 2013). The increased environmental perception for the visually impaired is by automatically identifying the surrounding objects and feeding back the information to the wearer (Duffy M., 2013).

A visually impaired user can give a voice command to the glass to take a picture of what is in front of them and then ask it to find a specific object in the picture or ask what is in the picture. The glass processes the image and searches for the specific object and indicates whether it exists or not and gives its location if it exists.

Figure 3.5: Google Glass

(Stevens T., 2013)

Another application is called Memento. It stores pictures with descriptions of each picture given by sighted people. These pictures could be of a particular living environment, e.g. a room. The visually impaired person wearing the glasses could ask for a description of the room they are in and the glass would retrieve the matching picture from the database and provide the information associated with the stored picture. See video in (http://phys.org/news/2013-08-openglass-apps-visually-impaired-video.html). Thus, Memento can warn the users to avoid dangerous situations, harmful equipment and hazards. Carers can provide pictures and annotations of a whole house or a route to a certain place so that a visually impaired person can navigate around the house or go along the route with the help provided by the glass.

3.5.2 Badge3D Project

Sensor-based technologies have been developed to replace the use of a white cane. Most of the technologies are based on ultrasound sensors, laser or light-emitting beacons to measure distance and angle. This will give the user an idea of the whereabouts are the surrounding obstacles.

On the other hand, vision-based systems use image processing techniques to recognise objects and identify obstacles and hazards. Vision-based technology combined with GPS and also be used for path planning and inform the user the destination is reached.

Scientists from Messina University suggested a system that is based on a single camera and visual tags or markers. Therefore, it can be classified as a tag detection system. Object recognition is performed by 'reading' visual tags attached to objects in the environment. This system requires prior placement of tags on selected objects. The system is designed to work in real time to detect and track tags accurately. All the identifiable objects are tagged with barcodes that can be read by a wearable computer system. Each barcode is surrounded by a rectangular black shape to make it easier to identify the tag in a captured image. The four corners of the marker determine the position and orientation of the tag relative to the camera calibration. The user can interact with the system by asking for an object present in the scene using spoken commands, and receive information through the speech-based interface and then guided to the target. In addition, unknown obstacles are avoided by an ultrasound-based alarm device mounted on the user’s belt.

During navigation, the system receives both sensor signals and a video stream. The video stream is captured by a camera mounted on the user’s head. The video stream is processed frame by frame to detect and decode any identifiable tags (Iannizzotto G., et al., 2005).

3.5.3 Artificial Intelligent Powered Vision

Horus is a wearable device that is designed to bring more independence to people with visual impairment. The device is a headband with two cameras attached to gather information about the surrounding environment (see Figure 3.6). Horus utilises a Nvidia Tegara GPU for computer vision together with sensors and deep learning algorithms. It can be used to describe what objects are in the direction that the cameras are facing. It can learn to recognise the shape and appearance of an object by rotating the object in front of the cameras and then Horus can recognise this or similar objects later from different angles.

Figure 3.6: Horus: AI powered vision

(Beckett J., 2016)

Horus can also be used to build a list of contacts using face detection and recognition technologies. It can notify the user of a known person when they are in view and can learn to recognise new faces. Another use of Horus is to read books and road signs. However, the main application of its 3D imaging software is to give people with visual impairment the ability and independence to navigate by alerting them of obstacles and giving them directions (Vilvestre J. 2016; Beckett J., 2016).

For mobility assistance, by generating 3D sounds with different intensity, Horus can alert the user of the presence of obstacles in the surrounding area and can be used both indoor and outdoor (HORUS, 2017).

3.6 Summary and Conclusions

With the increase in the number of diagnosis of CVI patients, researchers are investigating the cause, assessment and support relating to this condition. The CVI disorder has no medical treatments, but educational and practical interventions can be of benefit. Research and development projects have been carried out to provide tools to the visually impaired or blind to cope with their everyday lives and to help their carers understand more about the conditions.

The simulation of the visual fields of glaucomatous patients during reading attempts to show how patients used their central visual field to focus on the words being read. In addition, a study to simulate the visual field defects for the patients' road perception during the driving hazards test.

Electronic devices to assist VI patients and blinds come in different forms to replace the traditional white cane or guide dog.

Early systems were built using ultrasound sensors, laser or light-emitting beacons. Their main function is object detection and avoidance. However, with an obstacle avoidance system people with visual impairment are still unaware of what is in their surrounding environment. Therefore, recent research has gone beyond obstacle avoidance to recognise surrounding objects to provide even more help to the visually impaired.

Wearable computers and glasses with head-mounted cameras were developed for the above purpose. For example, *Google Glass* is based on object detection and recognition using image processing. Patients are aided by this system through interacting with the system using spoken commands to send instructions and receive information and assistance through a speech synthesiser or braille display.

On the other hand, the *Badge3D Project*, for instance, is based on barcode detection and decoding for object recognition and using an ultrasound component for obstacle avoidance. Barcode detection and decoding are not demanding computationally so the system can be used in real-time but requires objects to be tagged.

Horus is another device that uses a combination of technologies. It uses learning technologies for object and face recognition, and use 3D sound with different intensity to signal the proximity of objects nearby.

So far visual field defects had not been used to present realistically how CVI patients see the real world so that carers and specialists can gain a better understanding of the difficulties of the patients faced in their daily tasks.

Visualisation techniques to assist such understanding are required. Therefore, a simulation that projects the defected visual field onto 3D virtual world and imitates the patient's navigation in this virtual world is proposed to aid the understanding of the patient's carers by letting them see and experience what it is like to have similar visual impairment. This is the focus of this thesis.

Chapter four will explain the methodology of the proposed computational model for the projection and simulation.

CHAPTER 4: PROJECT METHODOLOGY

4.1 Introduction

The aim of this project is to enable people to visualise and experience what CVI patients can and cannot see in real-world situations using VR technology so that there will be an improved understanding of the difficulties that the patients face.

The project went through three stages. Each stage made use of outputs from the previous one, and software is developed for each stage for processing. The three stages are:

- Determining and extracting the CVI symbolic pattern from a visual field chart.
- Next, is designing vision masks to simulate the visually impaired areas of the VF to represent the normal and defected areas of the patient's sight.
- Then, is projecting the vision mask onto 3D virtual world scenes; and simulating the behaviour of CVI navigation, then allowing the user to explore and navigate around the scenes as if they were visually impaired with the same level of severity according to the visual field chart.

This chapter gives a general overview of these three stages and will expand in detailed. Chapter 5 explaining the VF symbols extraction from the VF chart, then Chapter 6 demonstrates the designing steps for the vision masks. Chapter 7 shows the projection process of the VF onto 3D virtual scenes, while the used algorithms for the navigation are explained in details in Chapter 8; and their simulations in Chapter 9. The proposed system to let the user exploring what the real CVI encounter during their navigation is described in Chapter 10.

Figure 4.1 shows the workflow for the main steps of creating the system.

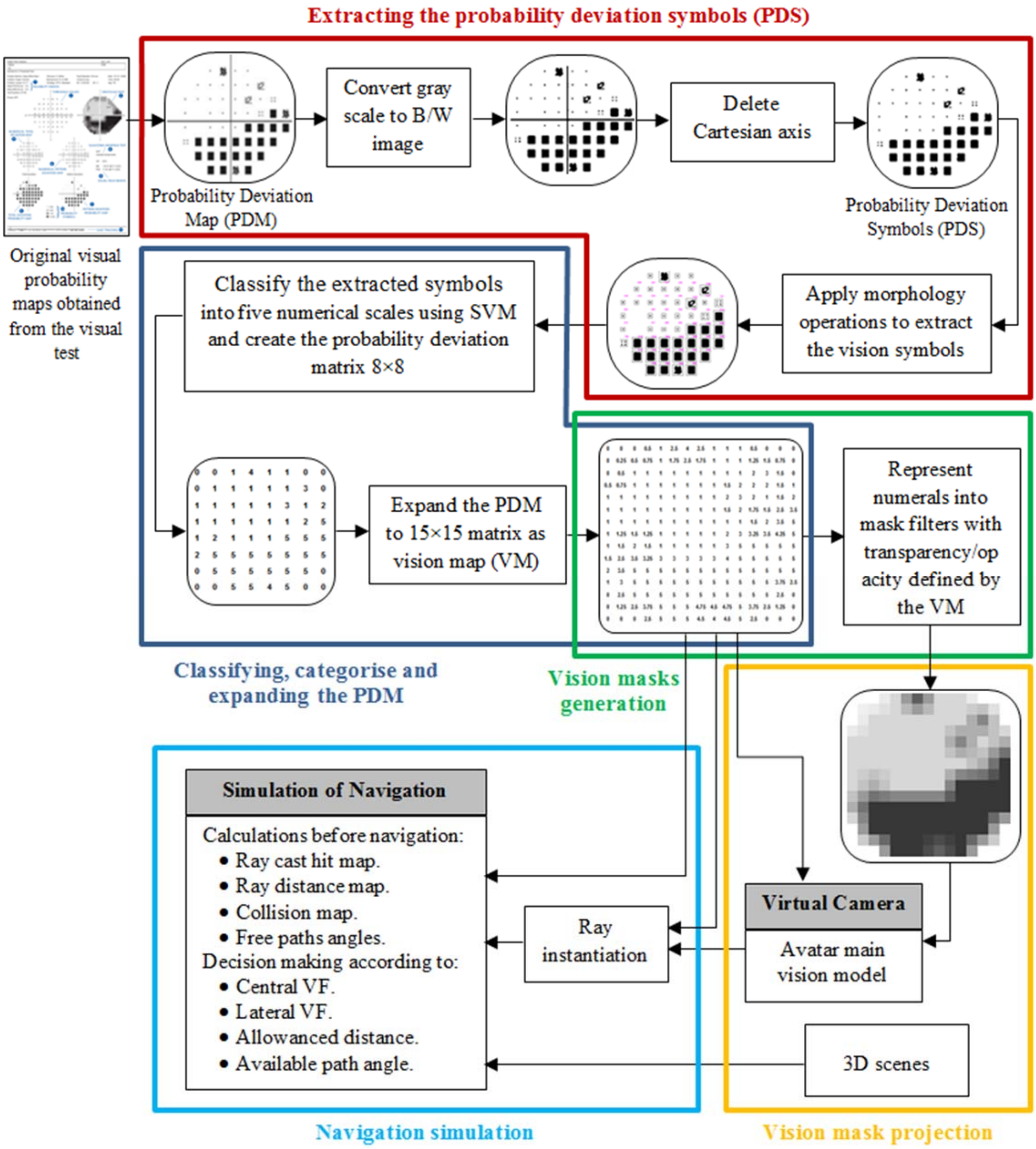
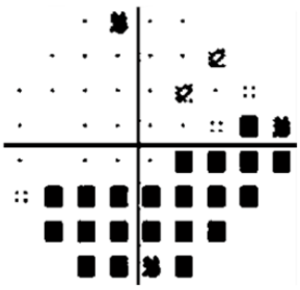


Figure 4.1: The implementation workflow

4.2 Visual Field Symbols Processing

The VF chart is divided into many numerical and symbolic patterns. Figure 4.2(a) shows the region of interest (ROI) which is the Pattern Deviation Probability Map. It consists of five symbolic shapes. The single blobs define the normal (healthy) visual areas while the dark rectangular shapes represent the most defected visual areas. The aim is to convert the VF chart into numeric values for later representation. Therefore, the symbols are extracted using image processing algorithms and morphological operations then categorised into a scale of 1 to 5 using a classifier. Accordingly, the support vector machine (SVM) algorithm, which is a supervised machine learning algorithm, is used for this classification purpose. Figure 4.2(b) shows the symbolic pattern in numerical form.



(a)

0	0	1	4	1	1	0	0
0	1	1	1	1	1	3	0
1	1	1	1	1	3	1	2
1		1	1	1	1	2	5
1		1	1	1	5	5	5
2	5	5	5	5	5	5	5
0	5	5	5	5	5	5	0
0	0	5	5	4	5	0	0

(b)

Figure 4.2: (a) visual field symbolic pattern and (b) the extracted and classified numerals

The numeric values are stored in an 8×8 array to represent the whole of the VF. The values are discrete and the difference between each value and its adjacent value is significant. Thus, the average values of each two adjacent values are inserted. The insertion is done in the whole array horizontally and vertically. Thus, the array is expanded to double size (15×15) to smooth these transitions between the values (see Figure 4.3). MatLab R2015a is used for this processing. Chapter 5 clarifies the whole procedure to extract the visual symbols and converting them into 2D numerical array.

0	0	1	4	1	1	0	0
0	1	1	1	1	1	3	0
1	1	1	1	1	3	1	2
1	1	1	1	1	1	2	5
1	2	1	1	1	5	5	5
2	5	5	5	5	5	5	5
0	5	5	5	5	5	5	0
0	0	5	5	4	5	0	0

0	0	0	0.5	1	2.5	4	2.5	1	1	1	0.5	0	0	0
0	0.25	0.5	0.75	1	1.75	2.5	1.75	1	1	1	1.25	1.5	0.75	0
0	0.5	1	1	1	1	1	1	1	1	1	2	3	1.5	0
0.5	0.75	1	1	1	1	1	1	1	1.5	2	2	2	1.5	1
1	1	1	1	1	1	1	1	1	2	3	2	1	1.5	2
1	1	1	1	1	1	1	1	1	1.5	2	1.75	1.5	2.5	3.5
1	1	1	1	1	1	1	1	1	1	1	1.5	2	3.5	5
1	1.25	1.5	1.25	1	1	1	1	1	2	3	3.25	3.5	4.25	5
1	1.5	2	1.5	1	1	1	1	1	3	5	5	5	5	5
1.5	2.5	3.5	3.25	3	3	3	3	3	4	5	5	5	5	5
2	3.5	5	5	5	5	5	5	5	5	5	5	5	5	5
1	3	5	5	5	5	5	5	5	5	5	5	5	3.75	2.5
0	2.5	5	5	5	5	5	5	5	5	5	5	5	2.5	0
0	1.25	2.5	3.75	5	5	5	4.75	4.5	4.75	5	3.75	2.5	1.25	0
0	0	0	2.5	5	5	5	4.5	4	4.5	5	2.5	0	0	0

Figure 4.3: Expanding 8×8 grid to 15×15 grid to smooth transitions between each two adjacent numbers

4.3 Vision Masks Design

The numeric values represented the level of vision defects in the different areas of the VF. The aim is to represent these numeric values into textures (masks). Then, all masks are gathered and aligned into 2D form to represent the vision masks.

The textures are designed using Photoshop software and each one has its degree of transparency according to the degree of deficiency. Textures are designed from merging two layers onto a single image. The first layer is a white image and the second layer is a transparent image. Both must have the same value of “opacity” and “fills” properties to represent the degree of deficiency. Such that, a 100% of opacity and fill properties produce a blocked black mask, while a 0% of the properties produce a transparent clear mask. The final mask is saved in PNG image format. Figure 4.4 shows the process of designing a vision mask.

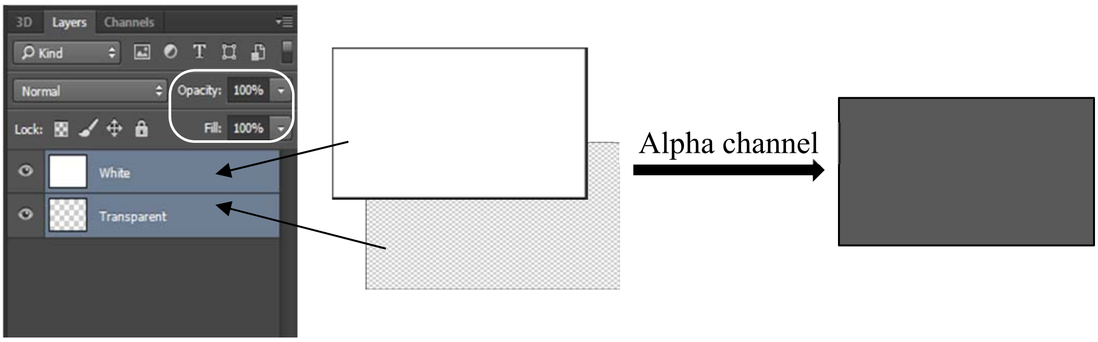


Figure 4.4: Steps required for designing a vision mask

The RGB channels in each texture are expanded by adding a fourth channel for transparency. The transparent Alpha channel is added using the facilities of the texture asset which is supported by the engine of Unity3D software. Finally, the vision masks are projected onto the 3D virtual scene to represent the normal and the defected visual areas as shown by the VF chart. Chapter 6 explains the step of designing the vision masks.

4.4 Projection and Simulation

Using the Unity3D game engine, a 2D array is used to hold the transparency values for all textures. Each location in the array is matched by a plane (Unity3D primitive 3D object) that holds the corresponding texture. Accordingly, the 2D visual masks are uploaded to 2D of planes to be placed in front of the avatar's main camera to blur the scene and to represent the VF impairment as experienced by the patient. The avatar in the 3D virtual environment represents the patient. The camera attached to the top of the avatar is considered the patient's eyes and its view represents the patient's visual field. The camera view is displayed onto the game view window of Unity3D. Therefore, the user can see the simulated view of what the patient would be seeing.

Different VF patterns can be used and projected onto different scenes to explore the type of deficiencies in each pattern. The healthy and clear areas will clearly show the corresponding projected parts of the scene, while the blurred areas will conceal partially or totally the projected parts. The clear and concealed parts of the scene reflect what the CVI patient really sees. These scenes are parts of different scenarios designed for navigation, where the movement of the avatar and its attached camera exposes a different scene at each step from different angles.

Thereafter, each plane that hold a vision mask will be considered as the basis for generating the vision rays according to the corresponding values of transparency. The numerals can be used to specify the position, length and direction of vision rays. The calculated lengths represent the distances of how far the patient can see. These lengths are presenting the vision depth of a patient and stored into a 2D array to create the vision map. This map is dependable to create other required maps for the simulation of navigation. The length and the rotation angles about x and y-axes are calculated at the beginning of the demonstration.

The rays are formed as beams using the physics ray-casting facility of Unity3D. This ray-casting will cover the whole visual field to represent the regular visual space of an individual view. They have the ability to penetrate the objects in the scene and let the avatar goes between objects.

Depending on these rays, the avatar can detect the surrounding objects within the available visual field. Such that when a ray hit occurred, means that the object that had been hit is seen. While the object is not hit, means that it is not seen and a collision with the avatar may occur. The large values of defects are represented by short/non-ray beams, whereas the low and zero values of defects are represented by long beams by which the patient can detect the surrounding objects in his local environment. The ray length is calculated as follows:

$Ratio = View_{depth} / 5$ (maximum scale of deficiencies) .... (1)

$Ray_{length} = View_{depth} – Defect_{level} * Ratio$ .... (2)

For example, suppose the following:

view depth = 15 meters	view depth = 15 meters
deficiency = 5	deficiency = 1
Then:	Then:
ratio = 15 / 5 = 3	ratio = 15 / 5 = 3
ray length = 15 – 5 * 3 = 0 meters	ray length = 15 – 1 * 3 = 12 meters
(no ray – blind area)	(a little-blurred area)

By using the characteristics of the created rays give the opportunity to do many calculations, such as creating various numerical and logical maps, and computing many measurements that can be used to navigate any designed scenario.

Different scenarios were used to reflect different tasks that a patient can make in his daily life. These scenarios include how a CVI patient can navigate his private room, going upstairs, getting downstairs, and avoiding small obstacles like toys and detecting moving objects nearby or cross in front of his sight. The whole navigation depends only on the vision rays and the required calculations.

Chapters 7, 8 and 9 explain in details the process of VF projection and the representation of the rays and the simulation of behaviour of navigation.

4.5 The Emulation Tool for Projecting the Virtual Scene for Simulation

There are a lot of software that can deal with 3D modelling and simulation. Most of them are used for 3D modelling applications for designing structures, buildings, aircraft and 3D environment. Such software includes 3D-Max, Maya and Blender.

On the other hand, software that uses the pre-designed models and builds 3D space environments to model the 3D real world is defined as 3D engines. One of the most powerful software used to build a 3D virtual world that depicts and visualises the real world is Unity3D.

Unity3D is a powerful cross-platform 3D game engine. It has basic functions and features that provide modelling, texturing, graphics, animation, artificial intelligence path finding and 3D physics. Many interactive and attractive recent games were designed using Unity, such as "Call of Duty", "Battle of Fields", "Medal of Honour", "GTA" and others. Unity3D is also used in the cinema industry for the cinematic tricks. It’s easy to learn for the beginner and powerful to develop complex applications for the expert (ZAMOJC, I. 2012). It is free for academic use on Windows and Mac (http://unity3d.com/unity/download/archive).

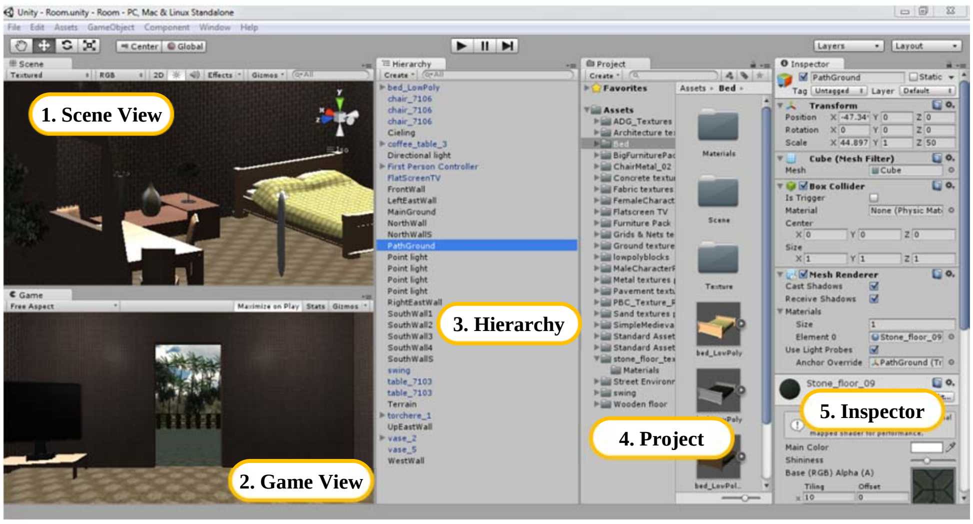


Figure 4.5: Unity3D game engine interface (version 4.0)
(http://unity3d.com/learn/tutorials/modules/beginner/editor/interface-overview)

Unity3D has a huge library of assets which considered the resources for the game. These like 3D models, textures, audio, materials, scripts and many others. Besides that, it has many basic simple (primitive) objects or assets such as a cube, sphere, plane, cylinder, terrain, camera, various lights and many others which fulfil the requirements of the project design. It has an empty object asset, which it can be used to gather many objects and consider them as one object to facilitate applying functions on them as one. Moreover, it has the ability to import and use of predesigned assets which created externally by 3D modelling application and painting tools. Therefore, it has the ability to import assets of prefabs (prefabricates) created by MAX, Maya, Blender, Cinema4D, Modo, Lightwave and Cheetah3D models. Also, it has the flexibility to be developed by writing scripts function using C# language and Javascript.

The First Person Controller (FPS/First Person Shooter) is used as a character of a patient (avatar) having CVI. During the project running time, the avatar simulates the patient’s navigation and moving. As a result of the main camera of the project is connected to the top of the avatar, it stands for the patient’s eyes and his vision dimensions are projected by the Unity supported game view window (Figure 4.5).

Unity has physics which can be used, for example, as the avatar body’s collider and its rigid body. These two physics are used for simulating the patient bumping with other unseen objects and enforce him to take an action such as turning away from the hit obstacles. Moreover, the most useful physics function is the ray-casting. By which the programmer can generate rays to consider them as the vision rays which the patient can see using them.

For any written algorithm using the Unity3D engine, there are priorities for functions/actions execution. Such that, the very start-up actions, values and variables should be written to be executed within *Awake* event function, but the initial actions and variables are implemented within the *Start* event function. While the actions need to be performed on each frame of the running game/simulation should be written and executed within *Update* event function. This order of implementations facilitates the separation of instructions and tasks according to their importance and priorities.

4.6 Summary

The visual symbols can be extracted from the pattern deviation probability map, which is obtained from VF test's chart. The symbols are located, segmented and classified using image processing techniques and morphological operators. The symbols are classified into numerical values on the scale of 1 to 5 using machine learning.

These numbers are stored into an 8×8 2D array to represent the visual field deficiency. The numbers are discrete, and the difference between each number with its adjacent numbers is significant. Therefore, the array is changed from 8×8 dimension into 15×15 dimension by inserting median values between each two values to smooth the transitions between them. Thus, the size of the array is doubled.

Texture masks were designed with different levels of transparency according to the classified defect values to represent what can be seen in the visual field areas. The Unity3D game engine is used to project the vision masks onto a 2D formation of planes and replaced in front of the avatar's camera to reflect the real patient's VF.

Some physics properties of Unity3D are used for rays instantiation to represent vision rays. The patient's imitation avatar can detect the surrounding objects using these rays. The big defected values are represented by short or no rays, but areas that have normal vision are represented by long rays. The rays cover the whole visual field to represent the vision map (how far the patient can see). The avatar's simulation of navigation depends on the vision rays' map, and on other calculated maps. So, the ray hit means that the object that had been hit is seen and the object is not hit by any ray, means that it is not seen and a collision with the avatar may happen.

The best software that simulates the 3D world is Unity3D. It has the ability to design desired scenes and platforms using its assets or imported assets predesigned by other 3D modelling software. The existence of the predesigned first-person character facilitates the simulation of a patient, and the main camera in the scene is set on the top of the character to simulate the patient’s eyes. This will ease the projection of the visual areas onto a 3D scene, and clarify what this patient is seeing. Unity3D is also used to design 3D scenarios. These scenarios represent the normal living environment, which includes rooms with furniture, stairs and moving objects.

The next chapter will explain in details the first part of the methodology which is the processing of the VF symbols.

CHAPTER 5: VISUAL FIELD SYMBOLS PROCESSING

5.1 Visual Field Chart Processing

Visual field test produces a chart with numbers and symbols that describe a person's vision, i.e. which part of the person's vision is normal, which part is defective and to what extent (see Figure 5.1). Without going into details, the representation that is of most interest to this project is the pattern deviation probability map (DPM) (Figure 5.1a). Each symbol on the map represents a probability range (Figure 5.1b). For example '<5%' means less defected areas and '0.5%' means highest defected areas. Therefore, the region with a small percentage value indicates poor vision and a larger percentage value indicates good vision.

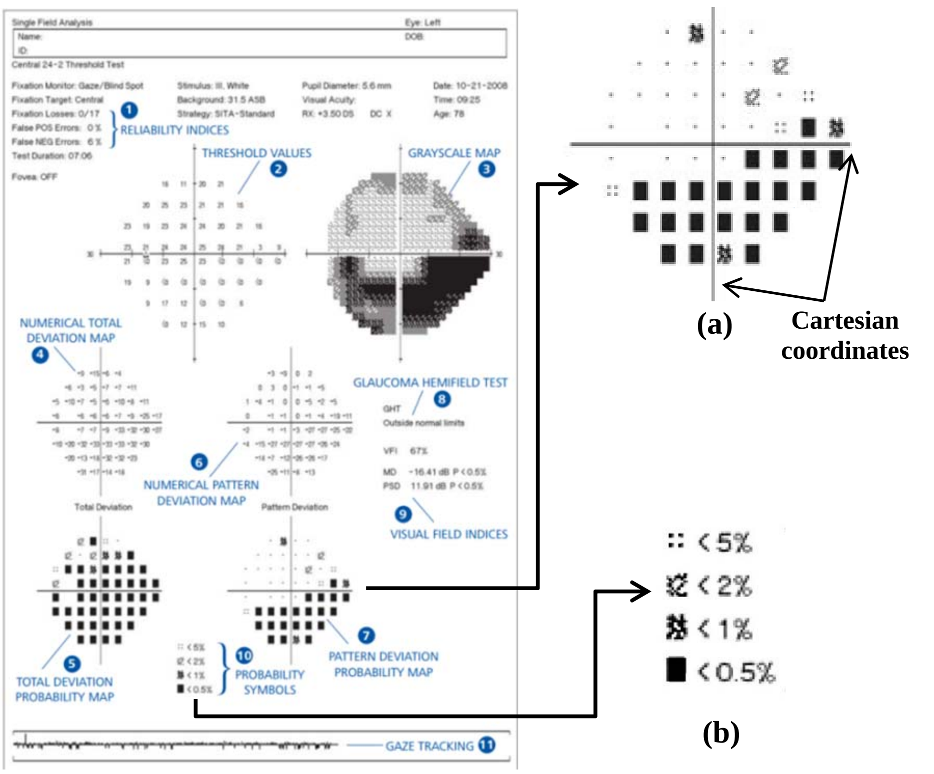


Figure 5.1: Chart of the Humphrey's Perimetry Visual Field test: (a) Pattern deviation probability map, (b) Meaning of probability symbols

(Carl Zeiss Meditec, 2013)

One of the objectives of the project is to convert the symbols in the visual field chart into a representation that can be used to mimic the CVI patient's visual defects (recognise the symbols from the chart and convert them to numeric values representing the level of deficiency). The rest of this chapter will describe how this is achieved; see Figure 5.2 for the required steps.

5.2 Identification of Region of Interest

The process of locating the pattern deviation probability map is performed by using an appropriate cropping tool to crop the region of interest (ROI) from the chart. MatLab has the facility to manage the location manually and then handle the cropped image for further processes.

The majority of visual field charts has a title in the left upper corner of the pattern. All texts around the map need to be removed and image processing will only focus on the map. The VF chart was scanned to form a digital image. This image probably has a low contrast resulted from the poor printing, scanning or due to the low-quality low-resolution of the source chart image (e.g. faded symbols). The image could be saved as a one colour channel (grey image) or as three colour channels (RGB image: .jpg, .png). For these reasons, a colour based segmentation processing was used for both single colour space images and for any three colour spaces RGB images (Andy Thé, 2014).

With the RGB image, the colour channels are separated into three planes. Each plane represents one of the corresponding R, G, and B image. These planes are converted to B/W image individually then all of the planes are summed into a single B/W image. Object detection is carried out using the B/W image by extracting the symbolic shapes using image processing methods of segmentation and morphology operations.

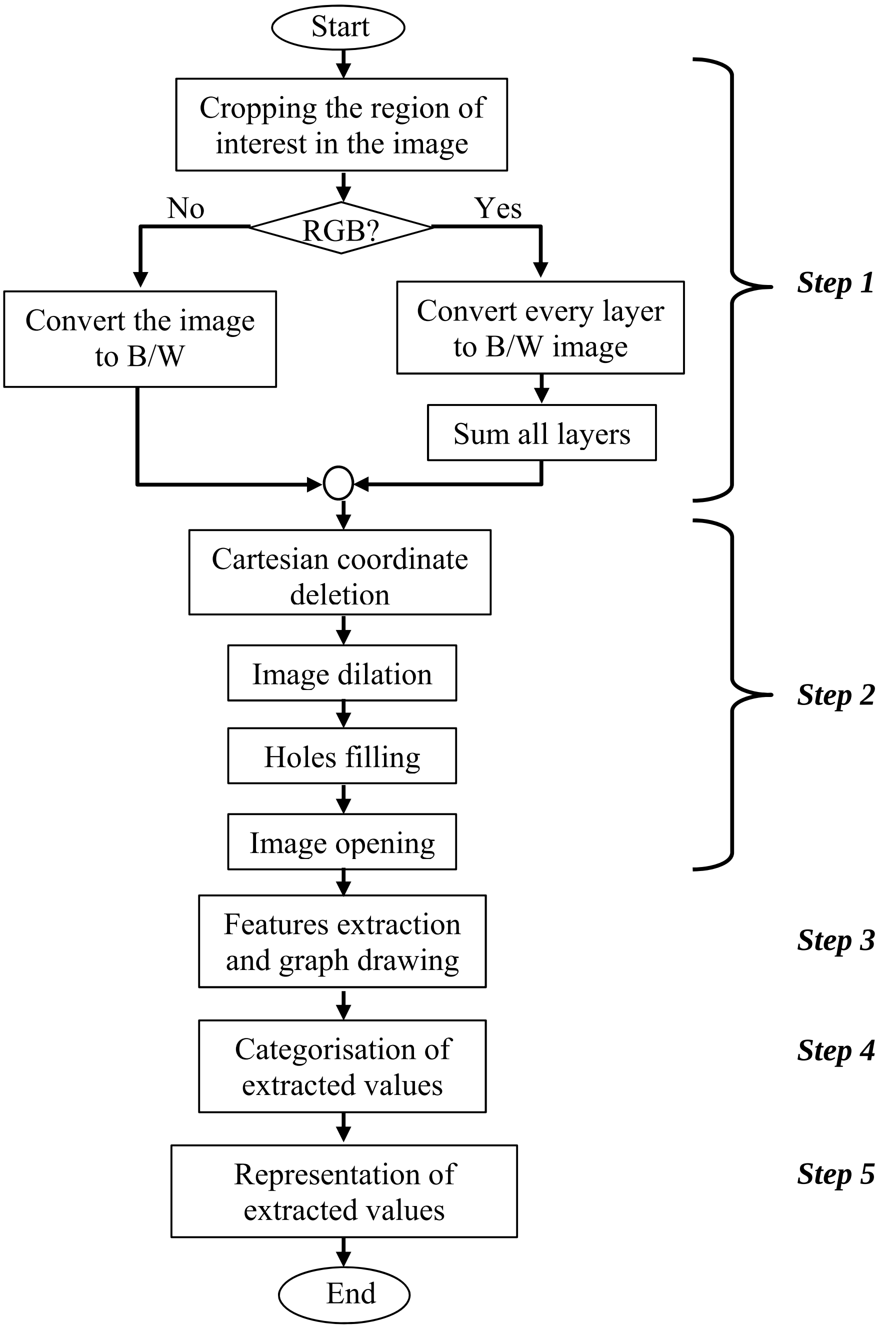


Figure 5.2: The required steps to extract numeral values from the deviation probability map. Step1: Checking the colour channels and converting the image into B/W image. Step 2 and 3: Morphological operations to enhance and extract the symbols features. Step 4: Classifying the symbols and converting them to numeric. Step 5: representing the numeric into 2D arrays

5.3 Visual Symbols Detection

Image segmentation is the first step in object detection by identifying basic objects and boundaries such as lines and curves. It gives the ability to use functions to analyse and extract meaningful information from the image. It helps to identify and isolate these objects using labelling process for assigning the same labels to the aligned image units. Such units are pixels, connected components or a group of pixels have specific visual features (Gupta, et al, 2014; Linda G. and George C., 2001).

Mathematical morphology is a set of operations that process images based on shapes, textures and colours etc. By applying elementary shapes (called structure elements), such as circle, line, square and diamond as masks to an input image and then generate an output image of the same size, such that, each pixel of the output image is compared to the corresponding pixel and its neighbours in the input image. The structuring element determines the size and shape of the neighbourhood, and the operator is sensitive to specific shapes in the input image. Such operators are image erosion, dilation, filling, closing and opening. Objects detection for the visual probability symbols was done using these morphology operators.

Since the cropped image of the pattern has Cartesian coordinates, these coordinates should be located and deleted. Firstly, the image is complemented to set the background to black (value 0) and set the objects to white (value 1). This helps to apply morphological operators effectively. As it can be seen in Figure 5.1(a), the Cartesian coordinates represent the biggest shape in the pattern. Secondly, all of the connected components (shapes/objects) in the image are calculated and their features (number of pixels, area and the boundaries) results in a matrix of cells. Each cell can be processed separately. Thus, the largest connected object with the highest number of pixels was considered as these coordinates and deleted by inverting its value to zero (see Figure 5.3).

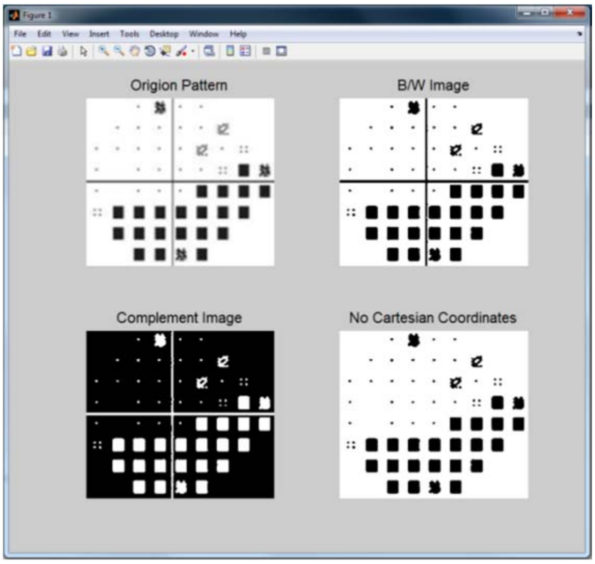


Figure 5.3: B/W image conversion to delete the Cartesian coordinates

Next, the visual probability symbols are located and separated using the most common morphological methods, *dilation*, *erosion, filling, closing* and *opening.*

Dilation and erosion are basic morphological operations. They are used for removing noise, isolating individual shapes and joining disparate elements in the image. Dilation expands a layer of pixels to both inner and outer boundaries of the shape by adding pixels to it. While the erosion shrinks a layer of pixels by removing pixels from the object boundaries. The number of pixels added or removed depends on the size and shape of the structuring element used (Linda G. and George C., 2001).

The ROI should be complemented before applying any of these morphological operators. Image dilation was performed to join the scattered and neighbouring elements by adding layers of pixels to them. After that, if any black holes left inside the objects, a 'hole' morphological filling process was used to fill the inside of the region, thus this region becomes as one block and the whole pixels of it become white (see Figure 5.4).

The morphological open operator is then used to erode the edges of the foreground bright objects (symbols) to remove the noise from these edges. Then it dilates these edges using the same square structural element (which is 'disk'). The open operation was implemented to prepare the image for features extraction.

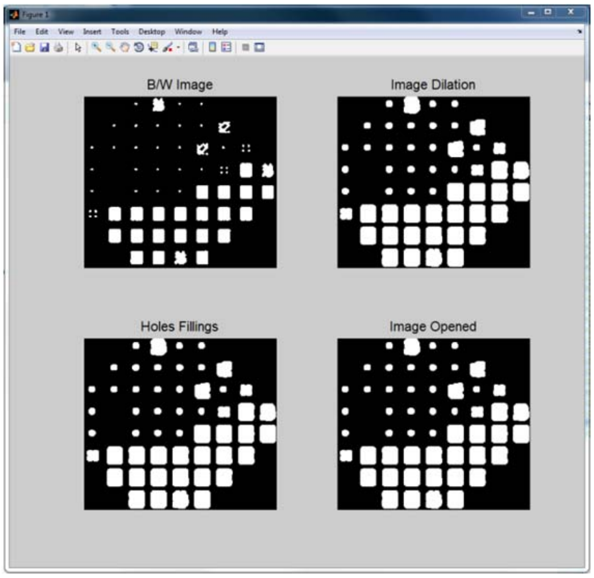


Figure 5.4: Image dilation, holes filling and image opened

5.4 Visual Symbols Extraction

After symbols were dilated, filled and opened, they were labelled to get the properties of these regions. By labelling the last opened symbols' image, the number of symbols in this image can be obtained. Then, using the region properties function within the labelled symbols, many features about these symbols can be obtained. The most useful derived features were Area, Bounding Box, Eccentricities and Centroid.

The eccentricities property helps to find the index of the symbols in the image (Index of Skittles). Thus, using these indices together with the bounding box feature, rectangular shapes were drawn over each symbol in the image. Symbols were located using the 'Centroid' feature and then extracted according to the 'Area' feature (see Figure 5.5). While using the *'Centroid'* and *'BoundingBox'* together, the extracted symbols were cropped then saved into a file for later classification.

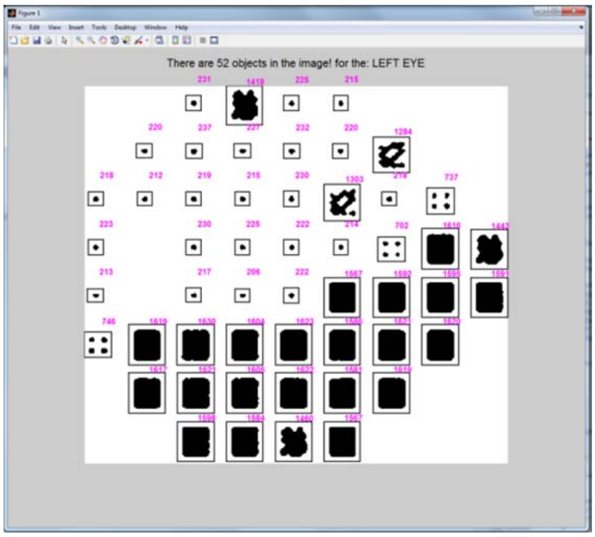
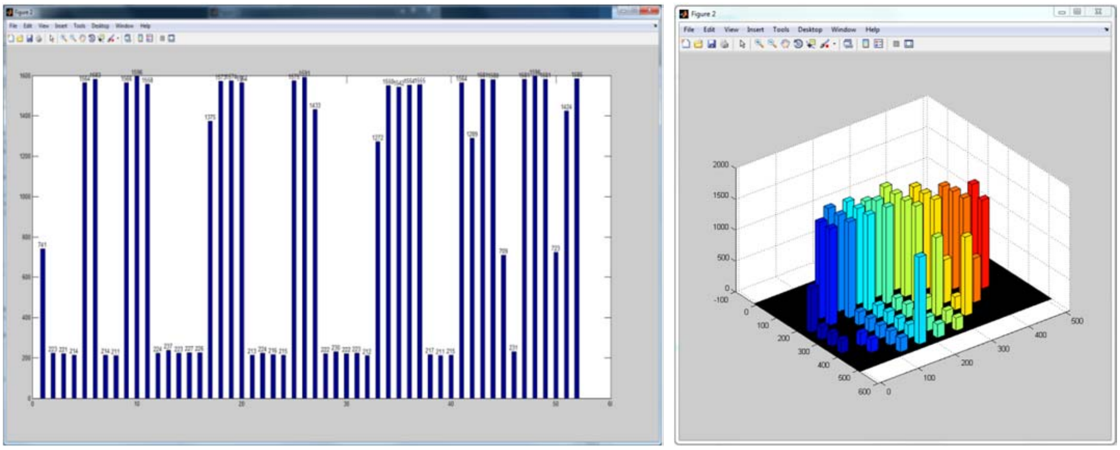


Figure 5.5: Symbols extraction and symbols' identification

The detected symbols are divided into four categories according to the degree of impairment in the corresponding visual areas (see Figure 5.1(b)). They start with the four dot shapes to represent the least damaged areas and end with almost rectangular black shapes for the most deteriorated and blurred areas. The pattern also has one dot shape that refers to the clear vision areas. Therefore, the extracted symbols will be five different shapes. Thus, the current symbol categorization will be [1 to 5] value scales.

Accordingly, the pixels scales and the symbols centroids help to draw these symbols in graphs showing the number of symbols, the number of pixels within each symbol and to locate them easily.

Figure 5.6(a) shows a graph to represent the numeral values in bars with their corresponding number of the pixels of each symbol in the pattern. The same bars in a 3-dimensional graph are shown in Figure 5.6(b).



(a) **(b)**

Figure 5.6: (a) Pixels scales for the five symbols, (b) Pixels scales 3D graph

5.5 Visual Symbols Classification

The extracted symbols were categorised into five scales using a supervised machine learning model called Support Vector Machine (SVM). SVM is a training algorithm used for classification, which analyses data and builds set of hyperplanes in multi-dimensional space. A hyperplane in the three dimension space (Euclidean space) separates it into two half-spaces. The idea of its implementation is to map the input vectors within the feature space through some linear or nonlinear mapping (Vapnik N., 1998; Cortes C. and Vapnik N., 1995).

Referring to the identification process for each symbol, the bounding box property was used to surround the symbols with the smallest box size, see Figure 5.5. Then each box was extracted and saved as an image within a corresponding folder. In each pattern image, the shapes were detected and segmented then extracted and saved in a separated folder.

According to the Computer Vision Toolbox functions provided by MatLab R2015b, pattern classification is done by using the concept of "a bag of visual words". It extracts key-points from the images in the training set, and then it creates a bag of feature vectors of each category from the images. This bag of features defines the visual words by applying the k-means clustering algorithm on the histograms of the features to create the vocabulary that leads to the visual words. The "grid" method is used for extracting the key-points with the Speeded Up Robust Features (SURF) detector which provide greater scale invariance to extract the features (visual words) (see Figure 5.7).

The set of representative images of all categories is partitioned into training and test subsets. Usually, 30% of the images are used for testing and the rest images are for training.

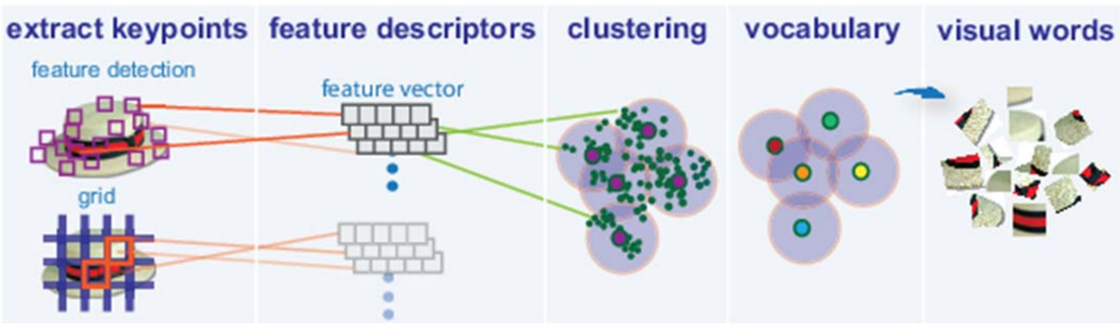


Figure 5.7: Processes to gain the bag of visual words required for SVM classification

(http://uk.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html)

The training process uses the bag of visual words to encode the images into the histogram of the visual words. Then the histograms of the visual words are used as positive and negative samples for classifier training and these histograms converted to features vectors. A new image to be tested is predicted to determine its category (Csurka G., et al., 2004).

Table 5.1 illustrates the classification related functions provided by the Computer Vision Toolbox in MatLab 2015b.

Table 5.1: Computer Vision Toolbox functions supported by MatLab 2015b

Name of Function	Description
imageSet	Organise the categories of the image for training the classifier.
partition	Create subsets from each category of the representative images.
bagOfFeatures	Encode each image from the training set to produce visual words.
trainImageCategoryClassifier	Return an image classifier.
evaluate	Tests the classifier on the test images portion.
predict	Apply the classifier on the new image to define its category.

The constraint of the classification algorithm of the Computer Vision Toolbox is that the training images should be saved in appropriate labelled folders. These labels describe the class or category that they represent. Since the probability symbols are five types, the training symbol images were therefore put in five separated category folders, and each folder is named related with the scale in this folder. For example, the shapes of a single dot symbol are saved into a folder named as "1" and the shapes of unhealthy black rectangular symbols are saved into a folder named as "5". Every extracted shape is classified and given a scale according to the name of the folder it classified with. The least scale (scale 1) indicates the clear vision in these areas, whilst the large scale (scale 5) indicates the total blurred and blocked vision.

The output of the training code shows the sequence of implementations for organising the training categories and the encoding of the feature vectors, then evaluating all the test subsets after matching process. After that, it shows the prediction percentage which is given the percentage of all symbols matched with the five categories. Finally, the output gives the average accuracy which represents the average of the main diagonal of the confusion matrix resulted from the evaluation step.

```
* Evaluating 17 images from category 1...done.
* Evaluating 17 images from category 2...done.
* Evaluating 14 images from category 3...done.
* Evaluating 17 images from category 4...done.
* Evaluating 17 images from category 5...done.

* Finished evaluating all the test sets.

* The confusion matrix for this test set is:


                          PREDICTED
KNOWN    | 1       2       3       4       5
---------------------------------------------
1        | 0.88    0.12    0.00    0.00    0.00
2        | 0.12    0.88    0.00    0.00    0.00
3        | 0.00    0.07    0.93    0.00    0.00
4        | 0.00    0.00    0.18    0.76    0.06
5        | 0.00    0.00    0.00    0.00    1.00

* Average Accuracy is 0.89.
```

Figure 5.8: Training code implementation sequence

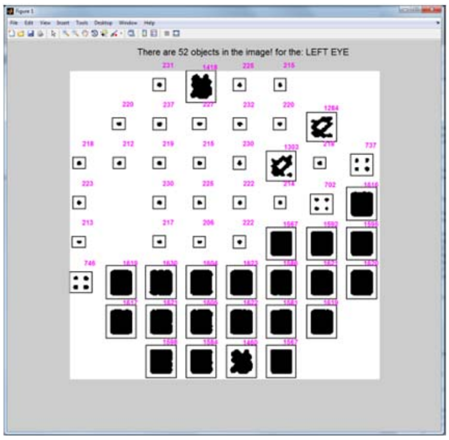
Figure 5.8 shows how the classifier is most accurate in classifying 5 with 100% accuracy. However, it is less accurate in classifying category 4 with 76% and category 3 with 93% and category 2 and 1 with 88% accuracy.

5.6 Visual Symbols Representation

For each pattern deviation probability map, (see Figure 5.1(a)), every symbol is segmented and extracted and saved as a separate image, and then classified into as an integer in the range between 1 and 5. A two-dimensional array is produced to store the corresponding values of the symbols on the map (see Figure 5.9).

There are 54 regions on the map of the probability symbols, but only 52 are tested because two of the regions are representing the blind spots, where no tests take place when the patient focuses at the centre of the Perimetry.

Therefore, only 52 symbols out of these 54 are shown in Figure 5.9. The blind spots are represented by two empty locations in the pattern.



→

0	0	1	4	1	1	0	0
0	1	1	1	1	1	3	0
1	1	1	1	1	3	1	2
1		1	1	1	1	2	5
1		1	1	1	5	5	5
2	5	5	5	5	5	5	5
0	5	5	5	5	5	5	0
0	0	5	5	4	5	0	0

Figure 5.9: Representing the symbol pattern into 2D array of numerals

Then a two-dimensional array of size eight by eight is formed to represent the extracted objects, such that each element in the array holds the corresponding scale of that object. The blind spots were calculated by finding the average value among the eight surrounding neighbours. Besides, the untested regions in the four corners are set to 0. Therefore, the final extracted integer values will be in the range between 0 and 5.

Later, this array will be projected onto a 3D virtual world, and its scales will be projected as five transparent levels of masks. Since the symbols were categorised within five scales, these values can be represented in five grey scales.

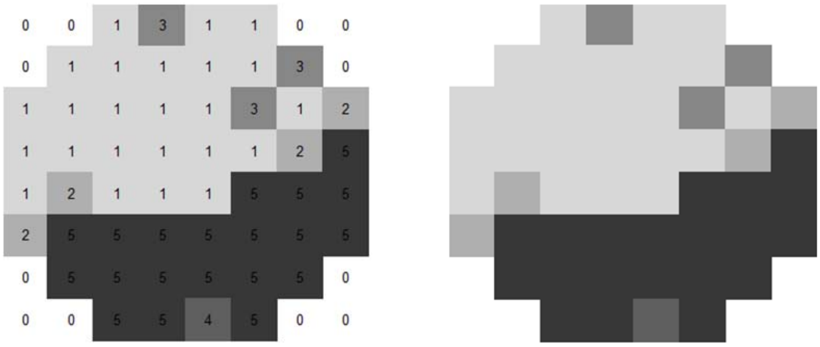


Figure 5.10: Transparency levels corresponding to the visual scales

Figure 5.10 shows the regions in the visual field are discrete. The transition from one region to its adjacent region can show a significant difference in the grey scale level. Therefore to provide a more gradual change, the array was expanded to 15×15 by inserting other elements between each two adjacent regions. Then, the calculations of the inserted elements were done by finding the average of the two neighbours in all rows and in all columns (see Figure 5.12 and 5.13).

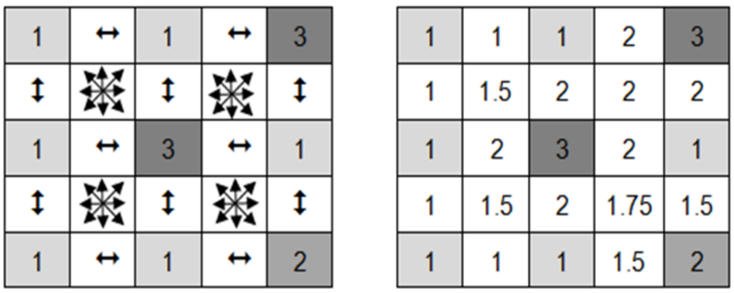


Figure 5.11: The procedure to expand the array of the visual values

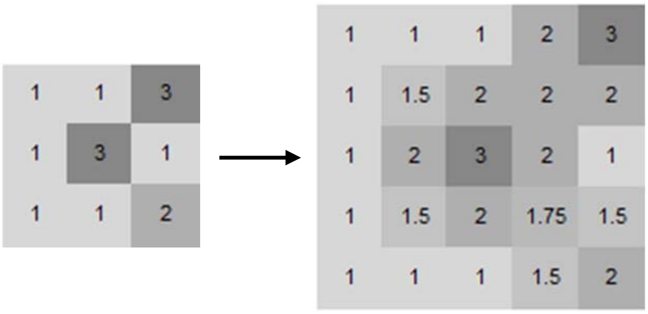


Figure 5.12: Adjusting the edges between scales to smooth transitions

Accordingly, these newly calculated values smooth the transitions between the original values by adjusting their transitions. Thus, these values determine how the real vision of a CVI patient is (see Figure 5.13).

The initial values that extracted from the pattern have five grades only. But, after the extension process, they became twenty-one grades:
[0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2 … 4, 4.25, 4.5, 4.75, and 5].

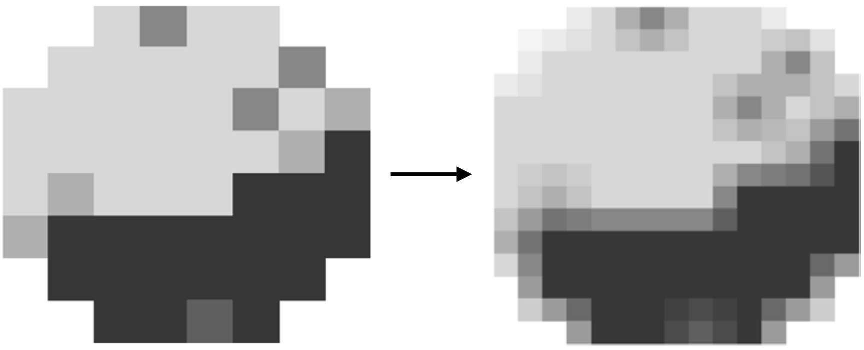


Figure 5.13: The whole visual field before and after adjustment

The expanded array with the calculated values represents the final map of the defected vision areas.

5.7 Summary and Conclusions

The use of the especially and the scientifically latest versions of MatLab software gives the ability and potential to deal with the images efficiently. The need to use such software as well as for the image enhancement is to extract its information, analysing them and then identifying its objects.

The use of the visual field chart resulted from Humphrey's Perimetry gives more ease and flexibility in the visual symbols extraction steps. In addition, the original symbols are aligned up precisely into two-dimensional arrays, which can be extracted and scaled then saved as numerals into a corresponding 2D array.

The classification process of the visual probability symbols is affected by the clarity and accuracy of the chart image. The more accurate printed chart or more high-resolution digital scanning of the chart, the extracted components and their properties will result in more accurate. The predicting percentage for the classification will be close to 1 regarding the confusion matrix.

Five steps are needed to process the visual symbols. The first step is to convert the ROI to B/W image. The second step is to delete the Cartesian coordinates in it and apply some morphological operators. Then is to extract the features in it and draw some graphs to show the number of pixels in each extracted symbols.

The fourth step is to classify these symbols and categorise them. Finally is to represent them into a 2D numeral array and expand it to smooth the transition among these values.

The next chapter will explain how to convert the numeric values of the VF defects into masks to be projected onto the 3D virtual scene.

CHAPTER 6: MASK DESIGN

Brain damage affects the vision because the image from the eyes is not processed properly. Hence there are areas in the visual field are blurred or unseen at all.

The idea is to design visual masks that represent the visual defects. Therefore, when a mask is projected onto a 3D scene, the person viewing the scene through the mask will experience a similar effect as the patient who has the visual impairment.

The projection process requires two pieces of specialist software. One is required for designing specific filters (masks) to represent the vision deficiencies as masks. The second is to project the masks onto a camera in a virtual scene.

6.1 Mask Design for the Transparency and Blurring Effects

Adobe Photoshop (PS) is an effective piece of software for image enhancement, merging, manipulation, objects extraction and other image editing functions. It can be used to manipulate images to create effects such as transparency and blurring.

Refer to the results created in the last chapter, the extracted numeric values have to be formed in transparent or blurred masks. The blurring and transparency masks were created using Adobe Photoshop version CC6 by merging two layers (slices) of images for the mask composition.

The masks' design using PS starts with opening new image with transparent background for the first layer, then to duplicate it into a second 'copy layer'. Next, is to open another new image with white background in the same size of the first image. Then, copying this white image and paste it onto the 'copy layer' of the first image (see Figure 6.1).

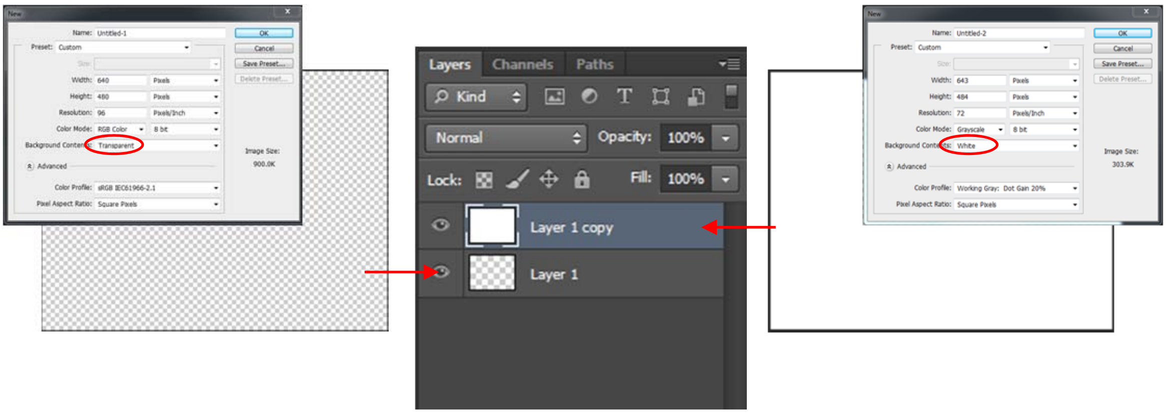


Figure 6.1: Two layers image to create a mask

The transparency of this image (mask) is determined by the *Opacity* and *Fill* properties of the layers in layers menu. Both layers should be selected and both properties have to be changed to the desired level of transparency (assigned to the same level of percentage), see Figure 6.2. The created image is required to be saved in "png" format of three colour channels (RGB). This is a constraint to create transparent images using Photoshop.

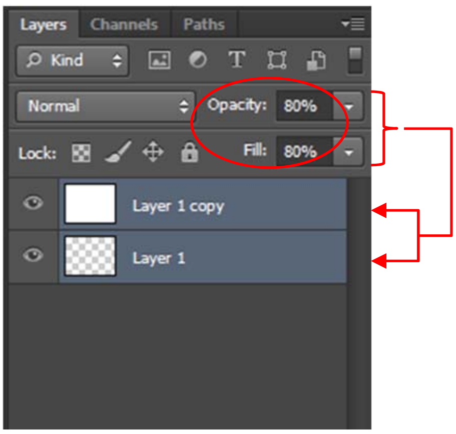


Figure 6.2: Opacity and Fill percentage for both layers of the mask

As mentioned in section 5.6, the scales for the probability symbols have integer values between 0 and 5. After expanding the array to smooth transitions between adjacent regions, there are now 21 values between 0 and 5. Therefore, 21 masks should be designed using the percentage of the opacity and fill properties as shown in Table 6.1.

Table 6.1: The opacity and fill percentage for the masks

Before array expansion			
No	Scale	% Opacity	% Fill
0	0	0	0
1	1	20	20
2	2	40	40
3	3	60	60
4	4	80	80
5	5	100	100

After array expansion			
No	Scale	% Opacity	% Fill
0	0	0	0
1	0.25	5	5
2	0.5	10	10
3	0.75	15	15
4	1	20	20
5	1.25	25	25
6	1.5	30	30
: :	: :	: :	: :
19	4.5	90	90
20	4.75	95	95
21	5	100	100

6.2 Transparency Effects Applied to the Masks

The created mask image has three RGB channels as it is "png" file. Yet, it is not transparent if it projected onto a camera within Unity3D software because it does not have the fourth transparent channel. Transparent images or having a certain level of transparency must consist of four colour channels. The fourth channel for transparency is *Alpha*. Therefore, this image should be converted to a four channels image (R, G, B and A) within the software that it will be used with (Unity3D).

As the software used for the projection and simulation is Unity3D, therefore, these masks will be converted to specific transparent format. The image of the mask should be loaded to the *Assets* folder in Unity. And after running Unity, this image should be selected and it will be considered as a texture asset. Then by checking the *Texture Type* option and selecting the *Advance* submenu, the '*RGBA Compressed DXT5*' format should be selected from the *Format* option. The mask will have a certain level of transparency as it assigned in Photoshop before and will be considered as a texture image (see Figure 6.3).

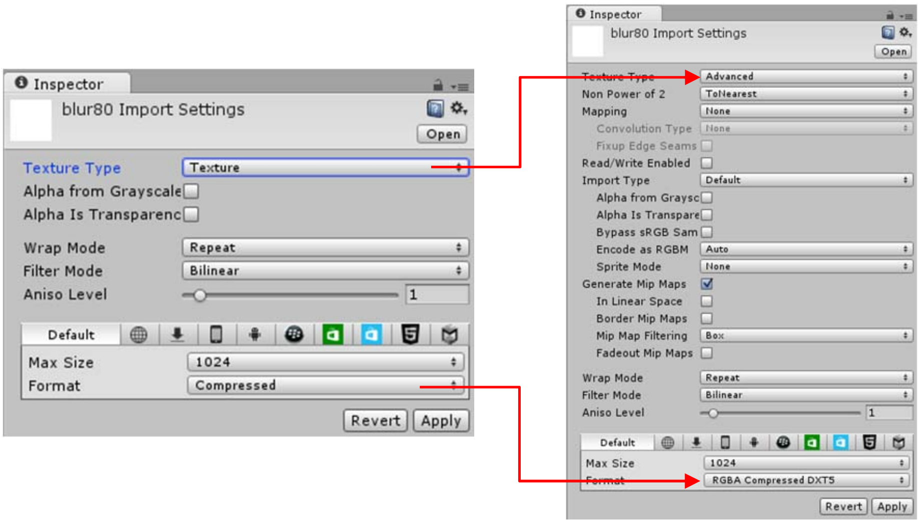


Figure 6.3: Setting texture's transparency in Unity3D

When a texture (mask) is attached to any Unity3D game object, it should have a program that utilises the texture coordinate information to modify the way this game object appears. This program is called *Shader*. Therefore, when attaching the transparent texture image to any game objects and make this transparency appears, the game objects should contain a shader. Accordingly, the object's shader should set to a transparent shader like 'Legacy Shaders/Transparent/Diffuse'. See Figure 6.4.

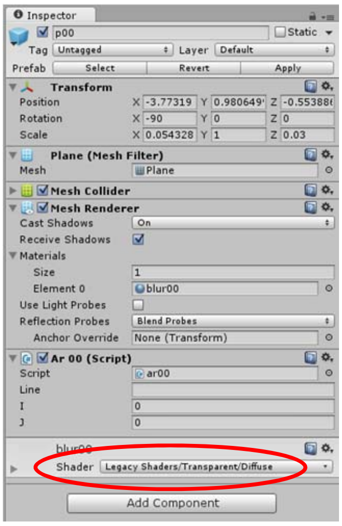


Figure 6.4: Assigning transparent shader to objects having transparent textures

6.3 Summary and Conclusions

Transparent images can be easily designed with appropriate levels of blur and transparency using image manipulation software like PhotoShop. The numeric values of the visual field defects are transformed to transparent and blurred images (masks). These masks were designed by applying two layers of images into one image. The transparency is set by assigning a certain percentage of the opacity and fill properties within each layer. However, the created mask needs to be transformed from a three-colour channels image (RGB) into a four-colour channels image (RGBA) to show transparency. Therefore, a transparent Alpha channel is added to the mask image when using 3D modelling software, such as Unity3D. The masks turn to texture images when they used by Unity3D. Textures have to be attached to 3D objects and shaders should be attached to these objects to show the textures' transparency.

The next chapter will explain in details how to align the created masks into one mask represent the vision masks and how to be projected onto 3D virtual scenes.

CHAPTER 7: 3D VISUAL FIELD PROJECTION

The main purpose of transforming the symbols from the VF test chart to different grades of transparent textures is to project these stacked textures in a 2D array form into a virtual world scene. This projection will reveal what the CVI patient actually sees at every step the avatar will make during the simulation of the navigation.

The first scene was designed as a simple 3D environment of a CVI patient's room to represent a living room environment. Figure 7.1 (a) shows the room design using primitive objects and predesigned prefabs like a bed, chest drawer, wardrobe, file cabinet and chairs.

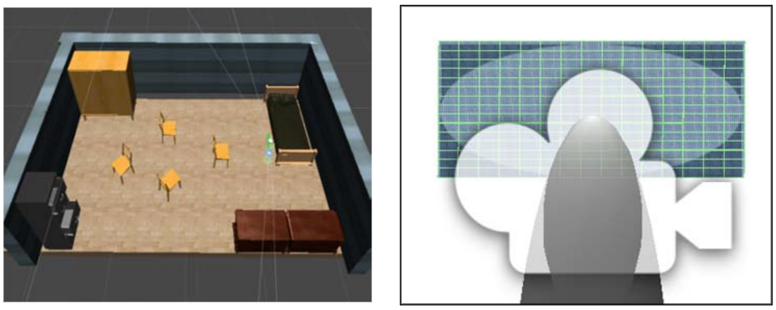
Other scenes were designed to fulfil other daily tasks like going upstairs, downstairs and navigation among moving objects.

7.1 Vision Mask

One of the basic and primitive available objects in Unity is the plane. A plane object was used to reflect the transparent textures. So, setting up and aligning many planes together to reflect the textures on them will project the visual areas on the avatar's viewpoint in the virtual camera.

As mentioned in Chapter 5, the array of the scales was 8×8 and enlarged to be 15×15 to smooth the transition between textures. Therefore, a 15×15 array of planes were designed and centralised in front of the main camera to cover the whole visual view of the patient. All planes have the same dimensions and each one has the corresponding numeric value taken from the extended numerals array. These planes were gathered into one empty object to form the whole vision mask and attached as a child object to the parent main camera. In addition, C# scripts were written and attached to the main camera to load these values to the corresponding planes and accordingly a specific transparent/blurred texture was attached to every plane automatically.

Figures 5.5 and 5.10 show the areas in the four corners of the VF pattern have not been tested because these areas are out of the visual perception during the fixation in Perimetry. Therefore, their values are not taken into consideration and were covered with black triangle corners shape (veil) with a full transparent ellipse in the centre. Figure 7.1(b) illustrates the vision mask array of planes and the veil that covers the untested areas, and attached on the top of the FPS/avatar and centralised within the main camera.

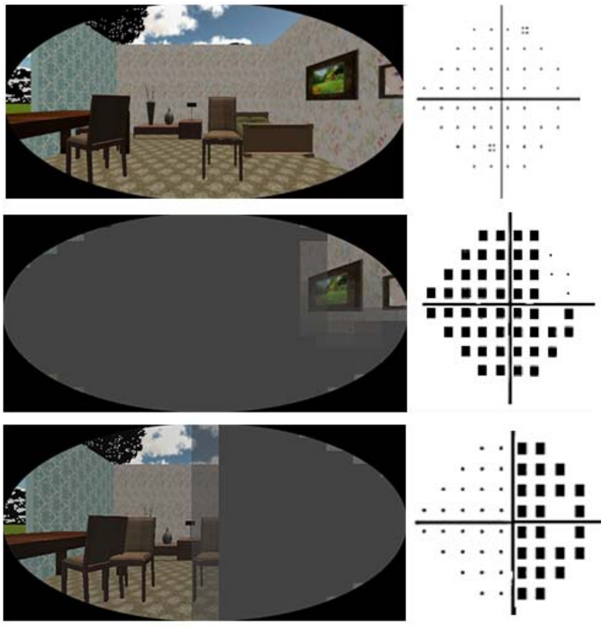


(a) **(b)**

Figure 7.1: (a) Room design and view point of FPS character (avatar), (b) 2D-array of planes of vision masks with the veil attached in front of the main camera

According to the process of the array extension, the array converted from the five original extracted integer grades [0, 1, 2, 3, 4, 5] to 21 extended float grades [0, 0.25, 0.5, 0.75, 1, 1.25, …, 4.5, 4.75, 5]. Therefore, for the position of each plane, a corresponding value in the same position in the numeral array was selected and the desired transparency texture was picked among the 21 textures.

The vision masks move with the camera into the four directions and projects the blurred spots in the patient’s eyes during his looking. This particularity permits to simulate the defects in the CVI patient’s vision at every step he takes during his navigation. Set of results were taken from the set of VF patterns, and each pattern represents a kind of vision deterioration. Thus, parents and educators can clearly conceive the visual areas that a CVI patient can see through them and the other blocked or blurred areas that he cannot see. See Figure 7.2.



(a) (b)

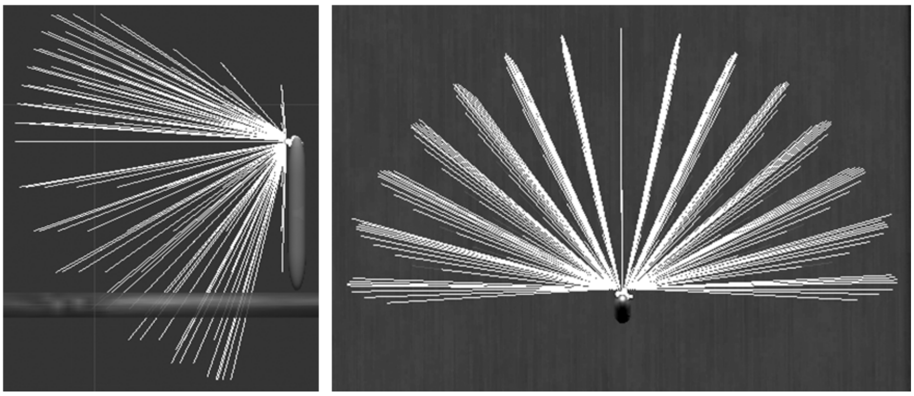
Figure 7.2: VF pattern projection. (a) Projected patterns into Unity, (b) Original patterns

7.2 Vision Depth

The extracted numeral values of the visual defects are converted into a representation of how far the patient can see using the ray-casting physics of Unity3D as vision rays. Vision rays are the beams that start from the eye and reach any point in the 3D space. A second array 15×15 of planes were designed and stacked as a 2D-array and placed in front of the vision mask. Each plane represents the base of one of the rays. The length of each ray depends on the numeric values taken from the original extracted values. Such that, the greater numeral values (means a greater percentage of opacity) represented by short beams in length.

While the smaller numerals (means a greater percentage of transparency) represented by longer beams. This was done by written scripts attached to each of the planes to instantiate the vision rays.

Also, the whole rays were gathered into an empty object to ease positioning and manipulating and to represent the *Rays Map* array. This object of rays is considered as a child object to the parent main camera as it is attached to it. The avatar moves ahead and turns away according to the movement and turning of the rays' object depending on a C# main algorithm script. By using this, it facilitates the control of the avatar's moving and navigation according to the presence of a ray hit (colliding) or not. See Figure 7.3 and Figure 7.4 for normal and deteriorated vision respectively.



(a) Side view **(b) Top view**

Figure 7.3: Vision rays for normal visibility

Therefore, the longer rays when hit an object this indicates that this object had been seen. While the short rays indicate that the avatar will be so close to the object till the object is seen. For the severe visual defected areas, no rays will be represented and the patient is about to bump into obstacles in front of him. Thus, the created vision rays are representing the vision capability of the patient.

(a)

(b)

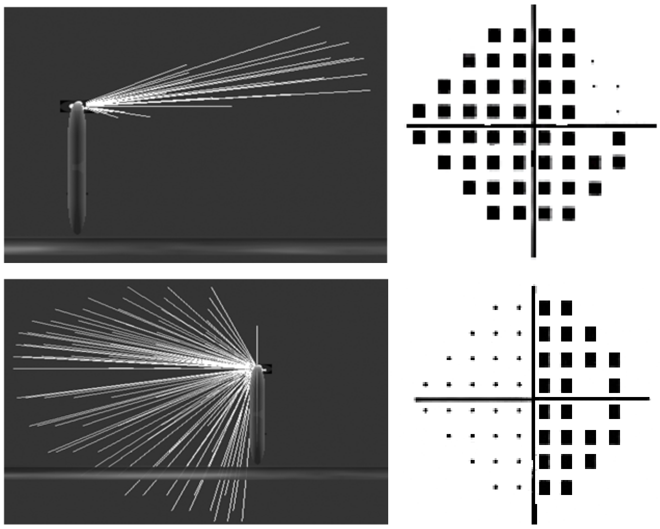


Figure 7.4: Vision rays for deteriorated visibility (back view). (a) Most of the VF are defected, (b) Right side VF defect

The length of each ray is calculated as follows:

$Ratio = View_{depth} / 5$ …………… (1)

$Ray_{length} = View_{depth} - Defect_{level} * Ratio$ …………… (2)

Where the *ratio* is the percentage of the view depth to the largest scale in the pattern, $view_{depth}$ is the depth of a personal vision and it's a value of units supposed by the programmer, and $defect_{level}$ is the scale value extracted and calculated from the pattern.

According to Chapter 2 section 2.4, the VF measurement for lateral field is 95 – 100 degrees and 60 upward (superiorly) and 75 degrees for downward (inferiorly) vision. Yet, the horizontal camera's default view in Unity is 100 units, while the vertical camera's view is 60 units. So, this camera view was adjusted to be 180^{o} horizontally, 60^{o} superiorly and 75^{o} inferiorly. Thus, the half measurement of both nasal and lateral field is 90, and the half of the superior field is 30. But, the inferior field had been adjusted to 75. Each ray was rotated about the x and y-axis according to the i^{th} and j^{th} of their positions in the Rays map. The rotation was calculated as follows:

$y_{rotation} = (j \times 12.857) - 90$ …………… (3)

$x_{rotation} = (i \times 4.286) - 30$ for superior field …………… (4)

$x_{rotation} = (i \times 10.714) - 75$ for inferior field …………… (5)

Where $y_{rotation}$: the rotation about the y-axis.

$x_{rotation}$: the rotation about the x-axis.

The generated rays and their lengths were saved into 2D arrays to be dependent on for further calculations and later processes. This array will represent the actual vison map of the patient clarifying how far the patient can see.

The following Figure 7.5 simplified the exact steps adopted for setting up each plane at the start of the algorithm implementation, and the ray generating during each updated frame of the implementation.

Initialization at the start of the implementation

Get and parse the position of the plane
Calculate the rotation angles about x and y-axes
Calculate the length of the beam to be instantiated
Rotate the ray (transformation of the base plane) according to the rotation angles

In each updated frame during the implementation:

Instantiate the ray starting from plane position to the forward direction
Draw this ray beam according to the calculated length
Save this created ray into a 2D array (Ray Map) in its parsed position.
Save the length of this ray into another 2D array (Vision Map) in its parsed position.

Figure 7.5: Algorithm for ray instantiation attached to each of the ray's planes

Another approach is to use the colliders' physics instead of ray casting for converting the visual defects into a representation of how far the CVI patient can see. These colliders can be instantiated the same way as the ray. The colliders were not used as they do not have the property of the penetration as the rays have. This means that the avatar can navigates among obstacles with no hassles as the rays penetrate the obstacles letting the avatar goes through. This characteristic is not available with the colliders preventing the avatar; for instance, go between two obstacles as the colliders hit both obstacles. For example, if a person can see 2 metres far and the collider is set at 2 metres which means that the avatar cannot get to an object closer to 2 metres, which is not the behaviour that is required to be simulated. The purpose is to see 2 metres not to avoid at 2 metres, therefore, ray penetration is important to let the avatar go closer to the obstacle before turning away from it. Besides that, these colliders' physics will overload the Unity engine and slow down the simulation.

7.3 Summary and Conclusions

The visual values of the VF pattern areas can be represented onto textures (masks). The textures can be aligned into a 2D array to be formed in a Vision Masks. Then, the vision masks can be projected onto a virtual camera of 3D software. The projection will reveal the clear and the defected areas as the real patient sees.

Vision ability is represented by rays that start from the location of a vision mask and end up into 3D virtual space. The length of each ray is calculated according to the defect value. The lengths of all rays are representing the Vision Map (how far the patient can see). So that, the long rays means the patient can see through these visual areas clearly, but the short rays mean the patient can slightly see through these areas. For this reason, when a ray-hit takes place it means that the object that had been hit is seen and if the object is not hit means this object is still unseen and the patient is about to bump into it.

The rays casting represents the Ray Map. Further calculations and other maps depend on the vision map and ray map.

The next chapter will explain the navigation algorithms based on the representational rays. The required calculations for the navigation and the decision-making are explained in details supported by pseudo codes.

CHAPTER 8: NAVIGATION MODELLING BASED-ON VISION RAYS

8.1 Introduction

Human navigation is the intellectual process to visualise routes to move from one place to another. The navigational strategies differ from person to other for planning these routes according to their capability and intelligent.

The human navigation behaviour goes through three stages:

- Grasping visible information: in every step a real person takes, there is a tremendous amount of visual information obtained from the surrounding environment. The avatar as well collects the visual information from the virtual environment according to the occurrence or the absence of the ray-hit.
- Reasoning situation: where the person recognises the surroundings and the obtained information is processed intellectually and necessary measurements are calculated to make a decision and take appropriate actions.
- Action situation: it's when the person (or avatar) decides what action to take. These actions can step forward, stop and wait, speed up, turn away or step backwards.

The avatar in the design uses the intelligent computations and the decision-making supported by the algorithm attached to it to simulate human navigation and the decisions taken to find the best route.

8.2 Vision-Based Situation Assessment

For any written script and algorithm using the Unity3D engine, there are priorities for action execution. For initial and startup actions, values and variables should be written to be executed within the *Awake* or *Start* event functions. While the actions need to be performed and/or updated on each frame should be written and executed within the *Update* event function.

In addition, many C# and Java scripts were written and attached to each of the avatar itself, main camera, each plane of the vision masks and to each plane that generates a ray. These objects individually have their own implementation priority according to their positions in the design hierarchy. As both the object of the vision masks' planes and the object of the rays' planes were attached as children to the parent main camera object; which this is, in turn, a child of the parent FPS (the avatar). The execution order for each event function of these objects must be taken into consideration of heredity. Figure 8.1 shows the hierarchy of the objects' sequence.

Figure 8.1: The hierarchy of the designed project

The main algorithm script is attached to the object of the rays' planes. Before starting the simulation of navigation, the algorithm makes some calculations at the beginning of the runtime for both Start and Awake functions. The algorithm is set to navigate many scenarios which are a local room, going upstairs, going downstairs and navigate among moving objects. So many initial variables for all scenarios need to be set in these functions.

In the Awake function of Figure 8.2, each of the avatar's height, speed and view depth is set. The height and the depth of the stairs are set too. Then, the ratio that represents the percentage of the view depth to the largest scale in the pattern (which is 5) is calculated. This ratio will be used to calculate the length of the rays. Then the two halves of the vision map values have been counted to indicate and select which half of vision is clearest and to be depended on turning decision.

```
In Awake event function:
    Set the following parameters:
        The height and speed of the avatar
        The view depth value
        The height and depth of the stairs
    Calculate the ratio of sight:
        ratio = view depth / 5; (5 is the largest scale of the probability symbols)
    Calculate the length of each ray depending on the origin numeral value array:
        for i = 0 to 15
            for j = 0 to 15
                ray length [i, j] = (view depth – original scale [i, j] * ratio)
            end for
        end for
    Calculate the clearest half of vision:
        Left half = Σ_{i=0}^{14} Σ_{j=0}^{6} ray length [i, j]
        Right half = Σ_{i=0}^{14} Σ_{j=0}^{6} ray length [i, 8 + j]
        if Left half > Right half
            then
                which Half = -1
            otherwise
                which Half = 1
        end if
```

Figure 8.2: Pseudo code for Awake function

While in the Start event functions, the starting time for algorithm running is obtained and all logical variables (flags) that used for sensing going upstairs and downstairs are set. The angle offset between each successor columns of rays is needed to calculate the 14 angles between the 15 rotated columns. Thus, fourteen possible angles/paths (θ – theta) are available and one of them is to be selected according to the column has the longest ray. The flag variables that indicate the walking beside lateral objects (walls, furniture, etc.) are calculated and set. They are used for checking if the avatar is walking beside a wall, so, left and right walk flag is checked.

As the height of the avatar is set in the *Awake* function, the physical capsule object of the avatar is modified and adjusted using the centre of y-coordinate of both the avatar and the main ground, see Figure 8.3.

```
In Start event function:
    Get the starting time.
    Set the logical flag variables.
    Calculate the angle offset between the rotated columns of rays:
        AngleOffset = (total horizontal angle of the view) / no. of columns of rays
    Calculate the rotation angles for the rays about the y-axis:
        for i = 0 to 15
            RayAngle [i] = (i * AngleOffset) – half of the horizontal angle
    Calculate and change the height of the avatar capsule according to the assigned height.
```

Figure 8.3: Pseudo code for Start function

The update function is performed at each frame during the project running; therefore, most of the avatar action is calculated and prepared continuously at the beginning of the running frame referred to the reasoning stage of human navigation.

As the whole project depends on the instantiated rays, therefore the generated rays' physics and rays' lengths are relied on. Accordingly, these data will be relied on to calculate the hit distance. Thus, the actual ray length during the navigation will be either the hit distance if the ray hits or the original instantiated ray length if it is not hit. The *Vision Map* (2D array) represents the originally created rays and holds the original lengths of these rays to clarify the distances of how far the person can basically see. While the *Ray-cast Hit Map* holds the properties of the generated rays' physics, such as the distance and the transform of the hit object.

At each updated frame the avatar having a step forward and the algorithm calculates and checks the following actions:

1. A binary 2D array *Collision Map* is created and updated to clarify which ray is hit and denoted by 1, and which is not and denoted by 0 to indicate it's a free ray. This can be very useful for ease checking the collision of the main three (central, left and right) visual fields.
2. The *Ray Distance Map* array is updated by physics re-casting the rays to hold the current distances between the avatar and the hit or non-hit surrounding objects. If the ray hits, the distance will be the straight line between the beginning of the ray (camera/eye) and the point that the ray hit the object (obstacle). But if not, the distance equals the original length of the ray. By this map, the remaining distance to an obstacle (before the collision occurs) is monitored.
3. In addition, a *Collided Columns Map* is created to indicate that the whole column is hit if at least one of its rays is hit. So that when a ray is hit, the whole elements of the column is set to 1 to indicate that the whole column is not walkable even if there are free rays in it. By this, the avatar can avoid upper obstacles such trees' branches/twigs and handrails and bicycles rails.
4. Then, a vector *Column Hit Map* is created to hold binary values such that each element in this vector indicates the situation of the whole column (1 hit, 0 non-hits). This map will facilitate the checking for the hit columns instead of checking the whole 15 elements in each column.
5. Turning away about the visible obstacles is depending on a user-defined value of the allowable minimum distance to turn before bumping. Therefore

and according to the *Collided Columns Map*, the shortest ray in the column that hit or the free ray in the column that not hit will be stored in *Min Distance Map*. Using this map will ease discovering the minimum distance in central, inferior, superior and lateral vision that force the avatar to avoid obstacles. This map will be repeatedly checked to find how far the avatar is away from obstacles. So that, if the avatar vision is good enough it will turn away when reach the allowable distance.

6. The longest distances in each half of the *Min Distance Map* represent the angles (θ_1 and θ_2) of possible walkable paths, one to the right and one to the left. Then, the angle θ of the walkable path in best half-side vision is chosen.
7. The same process of point 5 and 6 but on the inferior rays only is applied to find the minimum distance with its rotation angle to be considered when the avatar reach low objects (beds, chest-drawers, coffee tables, etc.) and nearby its body; whereas the superior rays are either free or greater than the allowanced distance.

Figure 8.4 shows the algorithm for the necessary preparations and calculations for the reasoning stage at the beginning of each frame runtime.

The preparation and calculation at the beginning of each updated frame:

Calculate the *Ray-cast Hit Map* using the physics ray cast of both *Ray-cast Map* and *Vision Map*.

Calculate the current rays length (Ray Distance Map) either it is the original ray length if it not hits or it is the hit distance if it hits.

Create the Collision Map (0, 1). 0 is for not hit (free ray), 1is for a hit.

Create the Collided Columns Map which indicates that the whole column is hit even if there is a free ray in it.

Create the Column Hit Map vector to indicate the situation of each column either hit or not.

Create the Min Distance Map (vector) that holds the shortest ray in each column.

Check the in front obstacles; if there is nothing then set the angle of turning away to 0 otherwise check the clearest half-side of vision and select the angle from that half.

Figure 8.4: The preparation and calculation algorithm before each updated frame

8.3 Navigation Decision Making

The rays of vision explain the areas and the surrounding objects that can be seen or not. The hit rays' means objects detected, however, the not hit rays' means objects are not detected and not seen. This principle facilitates the navigation through different scenarios depending on the vision rays only.

Vision, in fact, relies on the central visual field in the first place then relies on the lateral visual field. The algorithm for navigation depends on this fact. Therefore, it starts checking the central visual field for the obstacles in the front paths. While the field of lateral vision can be relied on to select the proper angle for turning away. And/or it can be relied on avoiding the objects that on the sides, like walls.

Therefore, the avatar using the algorithm keeps checking for the obstacles in front of it and at the same time checking for the obstacles that it passes nearby like walls and side furniture. The checking includes the hitting event and the minimum distance of allowance.

Different decisions have to be taken into consideration during the avatar navigation depending on the artificial intelligent computations which simulate the human intellectual decisions:

- The avatar keeps walking ahead if there are no obstacles in front of it and the central visual rays are not hitting anything or the central visual rays hit but the distances are still larger than the allowed distance. This fact works even if the avatar is very near to lateral obstacles; like passing between two chairs.
- During the navigation, the avatar keeps checking the central and inferior VF to avoid the obstacles ahead and not to bump into the low and small objects.

Then it checks the left and right visual field to contribute to the decision-making instantly for turning away from the lateral obstacles.

- People with visual impairment tend to turn to the clearest side of their vision. Therefore, the visual field is divided into two halves. Then in each half, the values of the rays' lengths are summed and the avatar whenever has to turn away from an obstacle it turns to a proper angle within the largest half which represents the clearest/best side of the vision.
- The fifteen columns of the rays represent fifteen possible paths. So, the angle of the most suitable path within the clearest side vision is selected. The suitable path can be considered as the angle of the largest free ray or the longest hit ray. The walkable path is considered as the angle of the largest ray that hits.
- In real situations of navigation, visual impairment people might be faced obstacles (no walkable path) within their best side vision. In such situation, people ignore the best side of their vision and they turn to the unseen side trying to flee from obstacles. This situation had been taken into consideration by checking the walk-flags variables (sensors) for both left and right side. Thus, if the avatar faced an obstacle and its best side vision faced a lateral obstacle too, it will turn to the unseen side by a small angle and keeps turning till it gets rid of obstacle and resuming walking ahead. This technique simulates the visually impaired person when he/she expects a path to the unseen side and turn away to it while his best visible side is facing obstacles.
- In the case of navigating among other moving objects, these intellectual decisions have a new direction in terms of taking the appropriate actions. Such actions are to stop and wait until the moving obstacle is passing by, or

move forward regularly, or stop and take steps backwards, or if the moving object is approaching fast then it’s better to accelerate the speed of the next steps.

The following Figure 8.5 illustrates a diagram for the navigation scenarios.

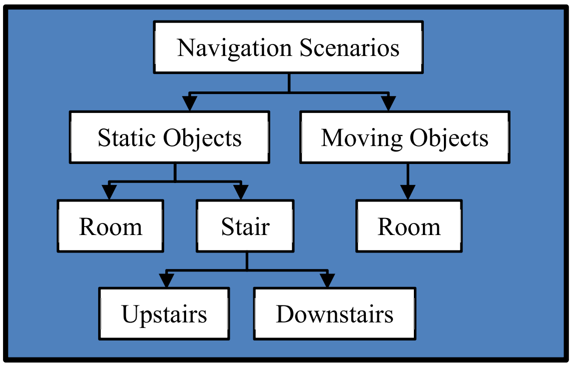


Figure 8.5: Navigation scenarios diagram

8.4 Scenario 1: Navigation Simulation with Static Objects (Room)

A room scene as a flat-surface platform was designed using Unity. Room assets were imported and included (such as a bed, chest drawers, a wardrobe, a file cabinet and chairs) to simulate a living environment that a CVI patient might live in, see Figure 8.6 below.

Figure 8.6: Room scene with static furniture and objects

The purpose of designing such room is the common environment and every day place for the patient to live; and it is designed to use different obstacles in terms of height and size. Some of them are low obstacles such as tables and chairs, others are tall like walls and wardrobe and the hanging ones like chandeliers.

Since the avatar has a 15 by 15 2-D array of rays, the computations and comparisons to check the whole visual field at each frame for hit events, distances, best angle for the best route and a proper decision to take an action will be long and complicated. Therefore, the algorithm divided the whole VF into three fields to ease the process of these computations and comparisons. Besides, as it mentioned, vision relies on the central VF during the navigation then onto the lateral VF, therefore, instead of checking the central columns which consist of 15-row elements each, it's enough to check few central elements in the *Column Hit Map* vector. If any of these columns hit and the hit object is static (will be explained in detail in the moving object scenario) and the distance between the avatar and the obstacle is less or equal to the allowanced distance then according to the computations for decision-making, the avatar will turn away from the obstacle.

Since the height of the avatar is variant by the assigned height value in the *Awake* function, the avatar height might represent the height of a child. So, the distance between the avatar rays' object and the ground will be smaller than the allowanced distance. Therefore, the avatar will keep turning round and round as the condition of the distance will be fulfilled. Therefore, the perpendicular distance between the rays' object and the main ground will be calculated to represent the inferior allowanced distance.

The same processing will be done with the central inferior VF to detect and avoid the low objects, such as coffee tables, balls, boxes, etc. The comparisons of the inferior VF will be overlapped with the rest of the central VF; because in some cases the central VF rays would be free with no hit events, while the inferior rays would hit small and low obstacles. Taking into consideration the navigation decisions; the avatar will check the left and the right existence of the side obstacles too. This lateral VF check is similar to the process of central VF except the minimum distance allowance will be the lateral allowable distances.

If the condition of the ray hit does not exist, this means the obstacle had not been seen. As a result, the avatar will keep stepping forward until it hits this obstacle. Then it will escape away using its clearest half of vision and by a small amount of turning angle. This simulates the impaired and blind people when they bump into something, they use their hands to touch the obstacle and turn away from it using small steps and a small amount of turning away angles.

Bumping event caused by the collider physics for both the obstacle and the avatar. When the avatar capsule collider touches (Unity enter event) other obstacles' colliders, then the decision of turning away will be made. This turning away will continue until the avatar collider doesn't touch any other colliders (Unity exit event). An audio source is attached to the avatar, so, whenever the colliding event takes place this audio will be played. The following Figure 8.7 shows the algorithm for navigating environments with static objects.

The processes for navigating the CENTRAL visual field:
    Check the central elements of the column hit vector; if **any** hit then
        Check the hit object is **static** then
            Check the minimum distance vector
                If the distances in the central vision are less/equal to the allowanced distances of the central VF or the central inferior VF then
                    Turn away.
                If these distances still greater than the allowanced distances then
                    Move ahead.
    If there no hit and the avatar hit the obstacle then the avatar will
        Bump and escape away.
The processes for navigating the RIGHT lateral visual field:
    Check the right elements of the column hit vector; if **any** of the **last** elements hit then
        Check the hit object is **static** then

Check the minimum distance vector
If the distances in the right vision are less/equal to the lateral allowanced distances of the right VF or the right inferior VF then
Turn away.
Else if the avatar hit the obstacle then
Bump and escape away.
The processes for navigating the LEFT lateral visual field:
Check the left elements of the column hit vector; if **any** of the **first** elements hit then
Check the hit object is **static** then
Check the minimum distance vector
If the distances in the left vision are less/equal to the lateral allowanced distances of the left VF or the left inferior VF then
Turn away.
Else if the avatar hit the obstacle then
Bump and escape away.

Figure 8.7: Navigation algorithm according to the central and lateral visual field

8.5 Scenario 2: Navigation with Stairs

The most suitable way to check if the stairs are seen or not and to be climbed or to be gone down is to use logical parameters (flag sensors). These flags can be set depending on the validity of the computations that done. These computations including the lengths of rays that hit and the hit-point that rays hit the ground or the stairs.

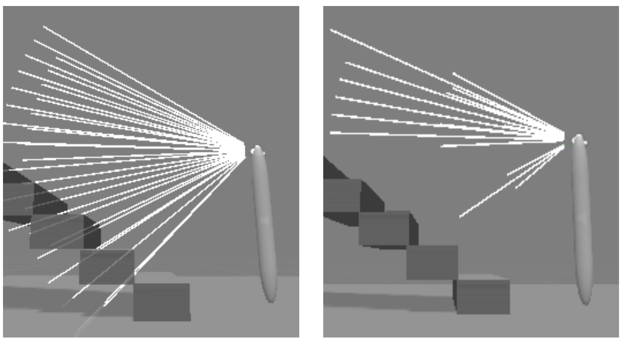
8.5.1 Upstairs Navigation Simulation

As human navigates depending on his central field of vision, he depends on his inferior VF to go upstairs. Thus, the algorithm for going upstairs relied on the central inferior rays. This because of the whole stairs may consist of a couple of stairs or many. Therefore, the lower vision was taken under consideration to check and calculate if the object in front is stairs or not. So, the checking for going upstairs process starts from the last row of the inferior visual rays backwards to the central visual rays.

As the rays map consists of rows, the algorithm starts from the last ray and compares the ray hit distance with the prior ray hit distance in the previous row. If each ray is less or equal to the previous ray then a counter will count the number of the rays that fulfil this condition. Thus all rays' hit distances have to be in increasing in length. Thereupon, a flag set to true to declare that the stairs have been seen and the rest checking of the other central visual rays is no more needed.

If the flag is set to declare that the stairs are seen, then the distance between the avatar and the first stair is calculated using triangulation to make it walk forward till the avatar reaches it. The distance between them will be compared to a small value simulating the human's foot doesn't touch the stairs when he reaches it.

The information of the climbing stair can be obtained using the physics of both the avatar and the stairs. The avatar will extract the local height scale of a stair and move up accordingly. The moving up process is by translating the position of the avatar about the y-axis by the stairs local height. Figure 8.8 shows how stairs recognition depends on the inferior VF, where if they hit the stairs will be seen, or if they not hit the stairs will not be seen and will be bump into.



(a) (b)

Figure 8.8: Going upstairs (a) Stairs is seen; (b) Stairs is not seen

If the avatar reaches the stairs while the flags are still set to false, this means the stairs are not detected and the avatar will bump into it simulating the stumbling and falling over them. Then the avatar will escape away from the stairs using small angle for its rotating. This escape will be towards the clearest half of its vision. See the algorithm in Figure 8.9.

The processes for going upstairs:

Check if the flags are false then
- Set the counter to 0.
- Start comparing each ray length with its predecessor; if it is less/equal then
  - Increment the counter.

If counter greater equal to the number of rays in the inferior visual field then
- Set the flag to true.

When the flags are set to true do
- Calculate the distance between the avatar and the first stair.
- Compare the distance between the avatar's foot and the stair with a small value
- Extract the stair information including the local height and local depth.
- If the avatar reaches the stair then
  - Move up to the height of the stair.
  - Walk along the depth of the stair.

If the avatar reaches the stair and the flags are false then
- Bump into the stair.
- Escape away to the clearest half of vision.

Figure 8.9: Algorithm for going upstairs

8.5.2 Downstairs Navigation Simulation

The process of getting down the stairs also starts from the last row of the inferior visual rays backwards to the central visual rays. Each ray length compared with the corresponding calculated down-sight limit; if it's greater than the down-sight limit then the stairs are detected and seen, but if not, this means that the stairs are not detected.

Each ray in the rays map has its rotation angle (θ) about the x-axis. And, the exact distance between the rays mask object and the walking platform can be found (perpendicular distance). Thus, the exact proper ray's length (oblique distance) between the rays mask object and the platform can be calculated using arc cosine for (90 – θ). The exact distances for the rest inferior rays are calculated too. These distances represent the down-sight limits, and they will be found during the *Start* event function. See Figures 8.10 and 8.11.

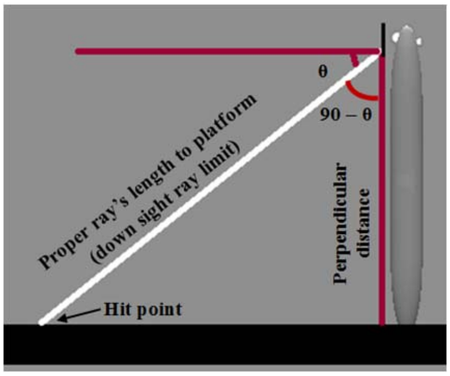


Figure 8.10: The proper length of the ray to the platform

The preparations and calculations before and at each updated frame:
At *Start* event function:

```
    Calculate the down-sight limit:
        Assign the set of the rotating angles about the x-axis for the inferior visual rays
        for p = 0 to 5
          down-sight limit [p] = perpendicular distance / cos (90–rotating angle [p] * π / 180)
```

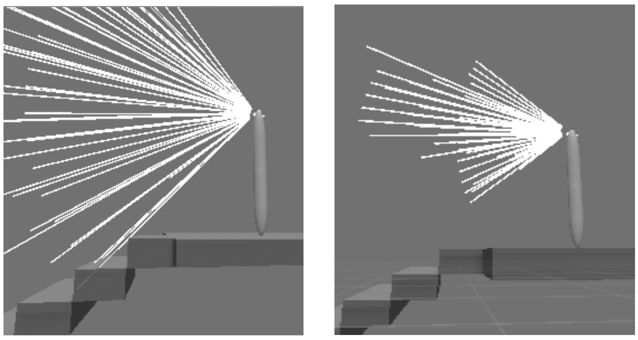
At *Update* event function

```
    Calculate the hit points in the platform (main ground) for the inferior rays:
        for i = 0 to 5
            if (ray hits) then grab the z position of the stairs
```

Figure 8.11: The preprocessing before and at each updated frame for downstairs detection

Depending on these oblique distances, the algorithm can then compare between the ray hit distance of the lower vision's rays and the calculated distances. If the ray hit distance greater than the down-sight limit meaning there is no floor ahead and the downstairs has been seen, and the getting downstairs flag is set and there is no need to check the rest of the inferior rays, see Figure 8.12(a). Then, the distance between the local centre of the avatar and the first stair to go down is found using the z-position of the ray hit point. Then, the distance between the avatar and the first stair is calculated to make the avatar walk along this distance until reach it to indicate that the avatar reaches the cliff of the stair. A half of the avatar radius will be added to the calculated distance to prevent the avatar back from colliding with the ground when getting down.

If the ray length less than the down-sight limit (ray is not hit the ground) means that the stairs are not seen and the avatar will fall over. The downstairs boolean variable indicator remains false, see Figure 8.12(b).



(a) **(b)**

Figure 8.12: Going downstairs (a) Stairs is seen, the inferior rays are greater than the down-sight limit; (b) Stairs is not seen, the inferior rays are not hit and less than the down-sight limit

When the avatar reaches the edge of the stair, it bows by rotating the camera about the x-axis with its entire attached vision mask and rays mask. This way can provide the avatar the whole information about the next stair including the stairs local height. Then the avatar straight up and moves down to the next stair, then to move ahead according to the grabbed local depth of the stair. This process will continue until there are no more stairs to move down. Figure 8.13 explains the algorithms for getting downstairs.

The processes for getting downstairs:

If the flags are false, then

Compare the ray length in the inferior visual field with the corresponding down-sight limit.

If it is greater than the down-sight limit then

Set the flag to true.

Calculate the distance between the current avatar position and the hit point position of the ray in the beginning of the first stair.

Add half of the radius of the avatar to prevent it from colliding the stairs during moving down.

If the avatar reaches the edge of the stair, then

Bow/Rotate down the camera object.

Extract the stair information including the local height and local depth.

Move down to the height of the stair.

Recalculate the distance to the edge of the new stair by adding the depth of the stair to the current avatar's position.

Walk along the depth of the stair.

If the avatar reaches the stair and the flags are false then

Fall over the stair.

End the run of the demonstration.

Figure 8.13: Algorithm for getting downstairs

8.6 Scenario 3: Navigation Simulation with Moving Objects

The detection and recognition of the moving obstacles are the most difficult issue that the partial visual impairment considers. The idea of detecting these moving objects depends on the rays' hitting point position.

When the ray hit occurs, the global coordinates of the object's centre (x, y, and z) can be obtained by the position of the object's transform using the ray-cast-hit physics property. For the static obstacles, as long as the avatar walking and the ray hit the same object, the same centre coordinates will be obtained. While for the moving obstacles, different centre positions can be obtained at each step of the avatar. The moving objects on a flat surface, their y-axis will be the same unchanged; but the x- and z-axis could be changed according to the direction of the movement at each updated frame. For this reason, a 3D array was created with the same size of the vision map (15×15) and consists of four layers. Each layer contains the positions of all hit objects in the visual scene. For the free and non-hit rays, zero positions are selected. Then for each column, the positions in the four layers are checked. If the four positions are the same for the same ray this means that the hit object is static, and a boolean flag variable is set to false as there is no moving object in this direction. Otherwise, if three layers or more are a different one from each other in positions' information; it means that the hit object is a moving one, and the moving boolean flag variable is set to true as there is a moving object in this direction.

Accordingly, the navigation algorithm for the central and lateral visual field will be developed to involve the detection of the not static objects. The central elements in the *Column Hit Map* vector will be checked for the occurrence of a hit. Then, the distance to the obstacle/moving object is checked too. If the obstacle is static, it means the moving boolean flag is false. Then the avatar will pick the proper angle for turning away as described in Figure 8.6. If the moving boolean flag is true, the avatar then either stop (waiting for the obstacle cross away) or if the object is coming directly to the avatar, therefore, the avatar will escape away. But, if the moving object is passing across and the distance is greater than the allowanced distance, the avatar will keep stepping forward.

The same process is applied to the left and right lateral visual rays' columns. By which, the side approaching objects will be detected too. If the moving object is approaching from lateral sides and the distance between them is less than the allowanced distance, the avatar will accelerate its speed and stepping forward fast to avoid the collision. See Figure 8.14 for the required processes to detect the moving obstacles.

The processes for navigating the CENTRAL VF and moving objects avoidance:

Check the central elements of the column hit vector; if **any** hit then

Check the min distance vector; if the distances in the central vision are less/equal to the allowanced distance then

If the moving flag is true then

Pause

Else

Turn away (as it is a static obstacle)

If these distances still greater than the allowanced distance then

Move ahead.

If the central elements of the column hit do NOT hit and the avatar hit the obstacle then the avatar will

Bump and escape away.

The processes for navigating the RIGHT lateral VF and moving objects avoidance:

Check the right elements of the column hit vector; if **any** of the **last** right elements hit then

If the min distance is less/equal to the lateral allowable distance then

If the moving flag of this column is true then

Increase the speed for acceleration

Else

Turn away (as it is a static obstacle)

Else if the avatar hit the obstacle then

Bump and escape away.

Check the right elements of the column hit vector; if **any** of the **first** right elements hit then

If the min distance is less/equal to the lateral allowable distance then

If the moving flag of this column is true then

Pause

Else

Turn away (as it is a static obstacle)

Else if the avatar hit the obstacle then

Bump and escape away.

The processes for navigating the LEFT lateral VF and moving objects avoidance:

Check the left elements of the column hit vector; if **any** of the **first** left elements hit then

If the min distance is less/equal to the lateral allowable distance then

If the moving flag of this column is true then

Increase the speed for acceleration

Else

Turn away (as it is a static obstacle)

Else if the avatar hit the obstacle then

Bump and escape away.

Check the left elements of the column hit vector; if **any** of the **last** left elements hit then

```
If the min distance is less/equal to the lateral allowable distance then
    If the moving flag of this column is true then
        Pause
    Else
        Turn away (as it is a static obstacle)
Else if the avatar hit the obstacle then
    Bump and escape away.
```

Figure 8.14: Algorithm for navigation the central and lateral VF avoiding moving objects

8.7 Summary

Rays in Unity can be generated and used as rays of vision. By using the ray-casting and the ray-cast-hit functions, a lot of properties can be gained and used for the required computations for the avatar navigation. The whole project for navigation is depending only on these vision rays and the calculations derived from them. Therefore, no other sensors were used but boolean variables flags.

Rays casting are the best way to represent the vision rays due to their sufficient functions and properties. Their properties facilitate their representation and their use in the simulation. During the avatar navigation, the rays collide and hit the surrounded objects exploring what is seen (hit) and what is not seen (not hit). Also, the rays can penetrate the objects making the avatar keep stepping ahead as long as the obstacle is not in front or far enough. This eases the avatar's navigation when it passing through two obstacles. Although the big number of the created rays didn't overload the Unity engine and the demonstration keeps work fluently.

The algorithm navigates the central VF first to check if there are any obstacles in front of the avatar then it checks the right and left lateral VF. If the obstacles are detected in front of the avatar and near enough then the avatar will turning away to a walkable path within its best vision side. If the obstacles are detected and close enough to the avatar’s side then it will turn away to the nearest walkable path within the other side.

The normal people’s actions (decisions) when navigating their local environments with static obstacles (e.g.: furniture) are either to stepping ahead or turning away from obstacles. While when they navigating environments with moving objects (e.g.: pedestrians, cars, etc.), their actions will extend to further than stepping ahead or turning away. They will be either stop until the obstacles pass by or step back or speed up the next steps when there are fast obstacles approaching from the side.

8.8 Conclusions

People depend on their intelligent during their navigation to choose the appropriate path and take the appropriate action to avoid the obstacles. Besides, they can take different decision for their navigation depending on their visual perception and the available alternatives, whereas the avatar in the proposed system depends only on the attached computational algorithm.

The navigational strategies differ from person to another for planning their visualised routes according to their capability and intelligent, while the avatar uses the intelligent computations and the decision-making supported by the algorithm attached to it. This algorithm consists of the rays generating and all of the necessary computations to find decisions that the avatar can use to find the suitable path. These rays clarify how far this patient can see and detect what the patient is seeing.

Important maps are generated and updated depending only on the rays and their properties. The whole idea depends on the original vision map, the ray length map, the colliding (hit) map and the distance map.

Most of the required computations used the triangulation by using the ray length and its rotating angle. Other computations and decisions used the comparisons between the initialised and the calculated maps.

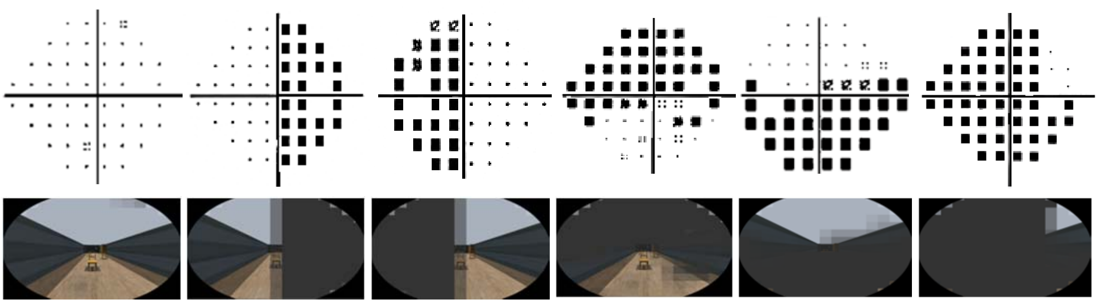
The next chapter shows results of the avatar behaviour during the navigation using different scenarios. A scenario of a room, scenarios of climbing upstairs or downstairs and a scenario for navigating among moving obstacles.

CHAPTER 9: SIMULATION RESULTS ON NAVIGATION BEHAVIOUR

9.1 Introduction

A number of virtual scenarios have been designed and created to mimic the living environments that CVI patients may face in their normal living as these scenarios are the most common and difficult daily situations for visual impairment people. The scenarios are used for avatar simulation to explore the behaviour and the expected decisions and steps made by CVI patients while navigating. These scenarios include a room with furniture, stairs for going up and down and a room with moving objects, like people moving around in it.

As mentioned in section 8.4 scenario 1, the purpose of designing such room is to use different obstacles in terms of height and size. Such that, low obstacles (i.e.: tables and chairs) are used whether they are recognised or not with the inferior or most defects VF projections, others are tall (i.e.: walls and wardrobe and the hanging ones like chandeliers) to check whether they are recognised using superior or most defects VF projections. Figure 9.1 below shows the patterns that were used in this study.



(a) (b) (c) (d) (e) (f)

Figure 9.1: The views of the avatar of the same room with same furniture arrangement but with different VF patterns. (a) Normal vision, (b) Right-half side defects, (c) Left-half side defects, (d) Superior defects, (e) Inferior defects and (f) Most of the VF defects

9.2 Simulation of Normal Vision

For the avatar using the normal VF, the simulation shows the expected behaviour as a normal person would do. Figures 9.2, 9.3 and 9.4 show the path in blue which the avatar took, turning when necessary to avoid an obstacle.

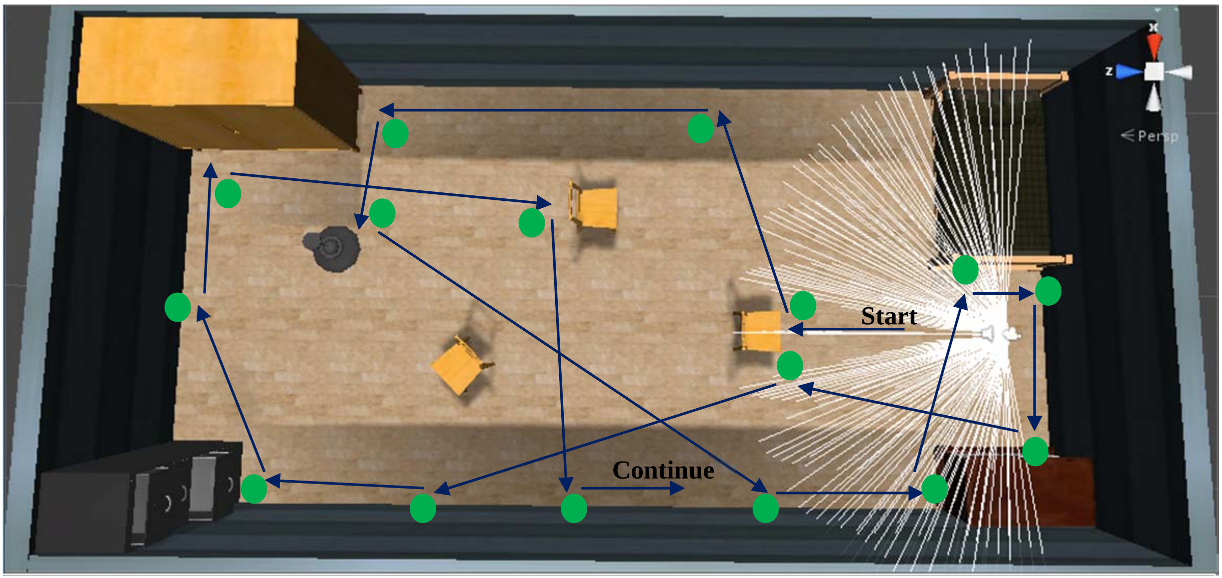


Figure 9.2: The path and the behaviour of the avatar with clear visual field

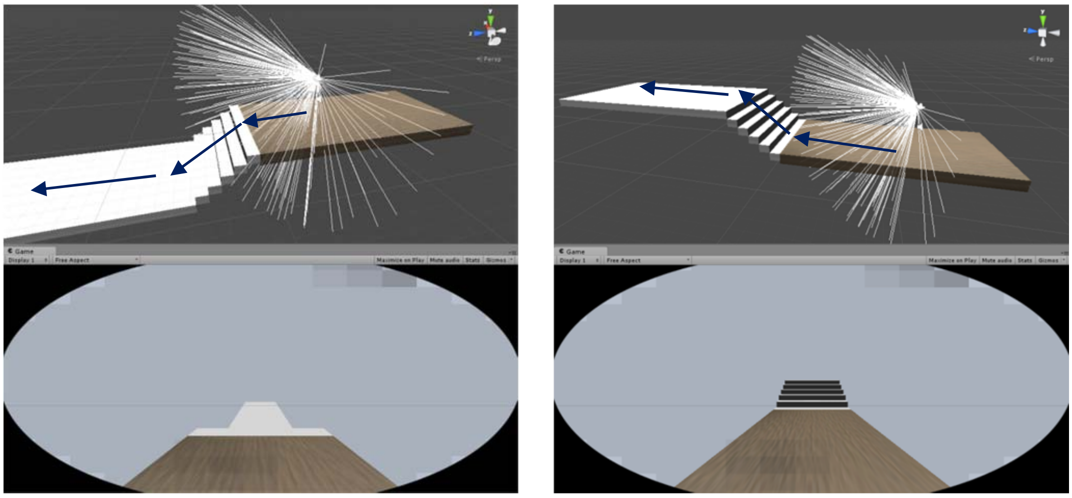
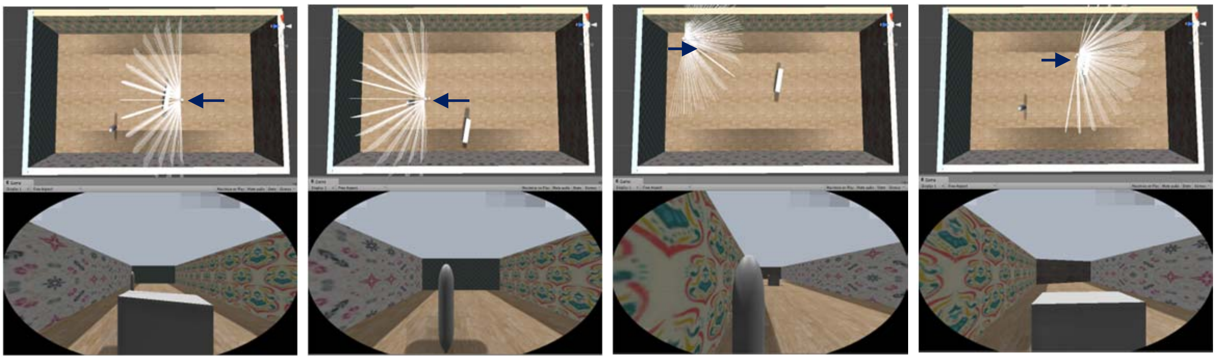


Figure 9.3: The behaviour of the avatar with clear visual field going up and down the stairs without problem



(a) (b) (c) (d)

Figure 9.4: The behaviour of the avatar with clear visual field navigating a room scene among moving objects. (a and d) Cube is seen and stop action has been taken, (b and c) capsule is seen and stop action has been taken

9.3 Simulation of Right-Side VF Defects

With the avatar VF with a defect on the right side, the simulation shows as expected the avatar is able to turn to avoid obstacles on the left but bumps into objects that are on the right of its paths. Figures 9.5, 9.6 and 9.7 show the behaviour of the avatar in different scenes.

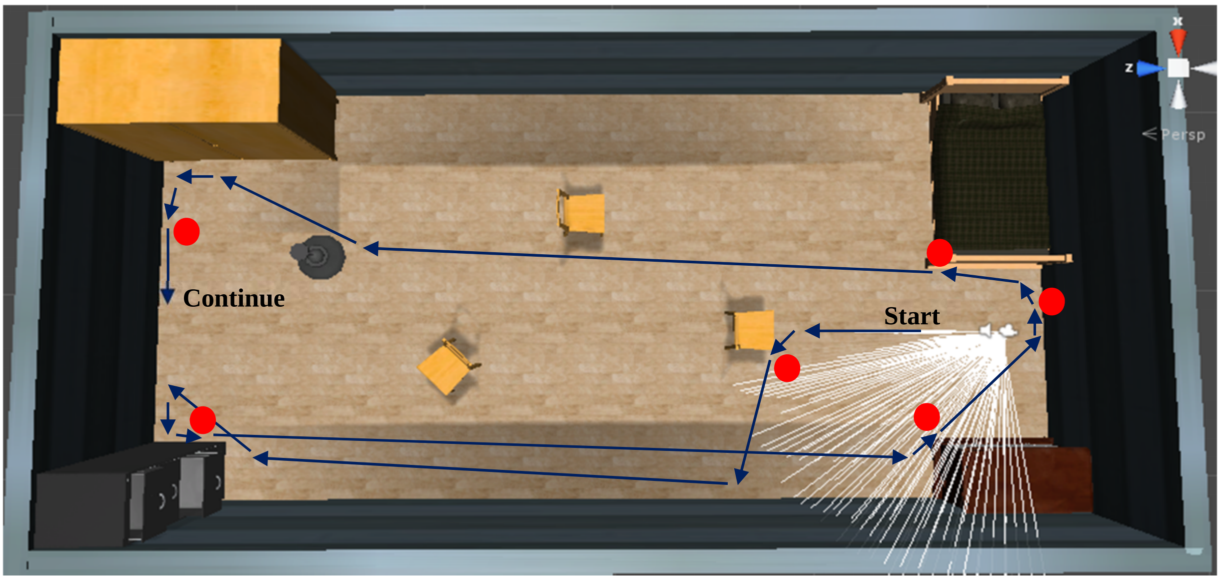


Figure 9.5: The behaviour of the avatar with right-side VF defects

● Object is bumped

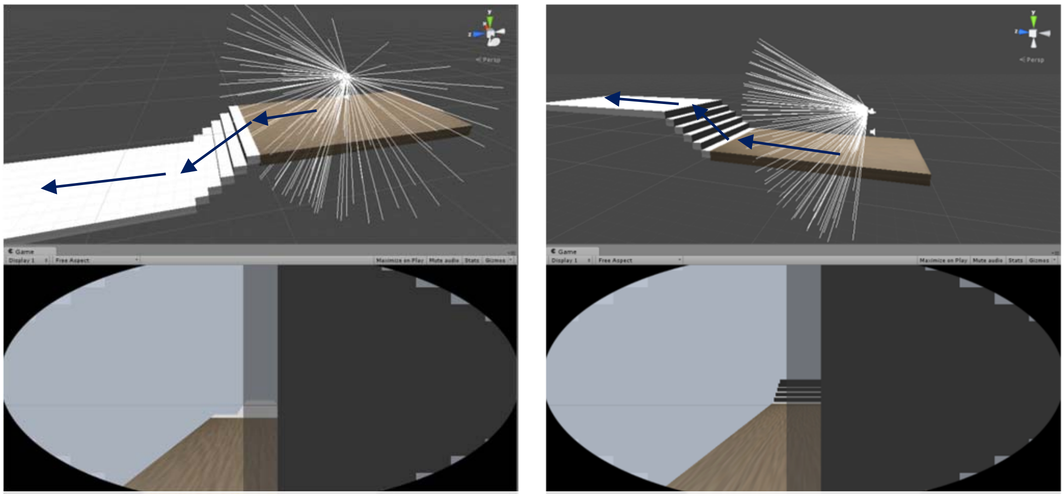
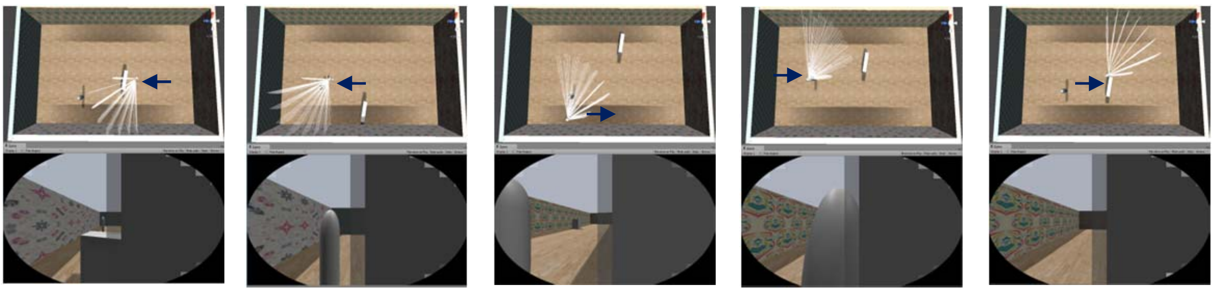


Figure 9.6: The behaviour of the avatar with right-side VF defects going up and down the stairs without problem



(a) (b) (c) (d) (e)

Figure 9.7: The behaviour of the avatar with right-side VF defects navigating room scene among moving objects. (a) Cube is seen and stop action has been taken, (b) Capsule is seen and stop action has been taken, (c) Capsule is seen and running action has been taken and (d and e) Capsule and cube are not seen and bumped with as they approaching from the unseen right-side.

● **Object is bumped**

In Figure 9.7(c), the behaviour of the avatar is to accelerate its speed and step ahead quickly as the capsule is approaching fast from the avatar visible left-side. While in Figure 9.7(e), the cube approached from the unseen right-side, therefore, a bump event occurred.

9.4 Simulation of Left-Side VF Defects

Figures 9.8, 9.9 and 9.10 show the behaviour of the avatar with VF defects on the left side. The behaviour is expected, it is able to avoid objects on the right but bumps into objects on the left.

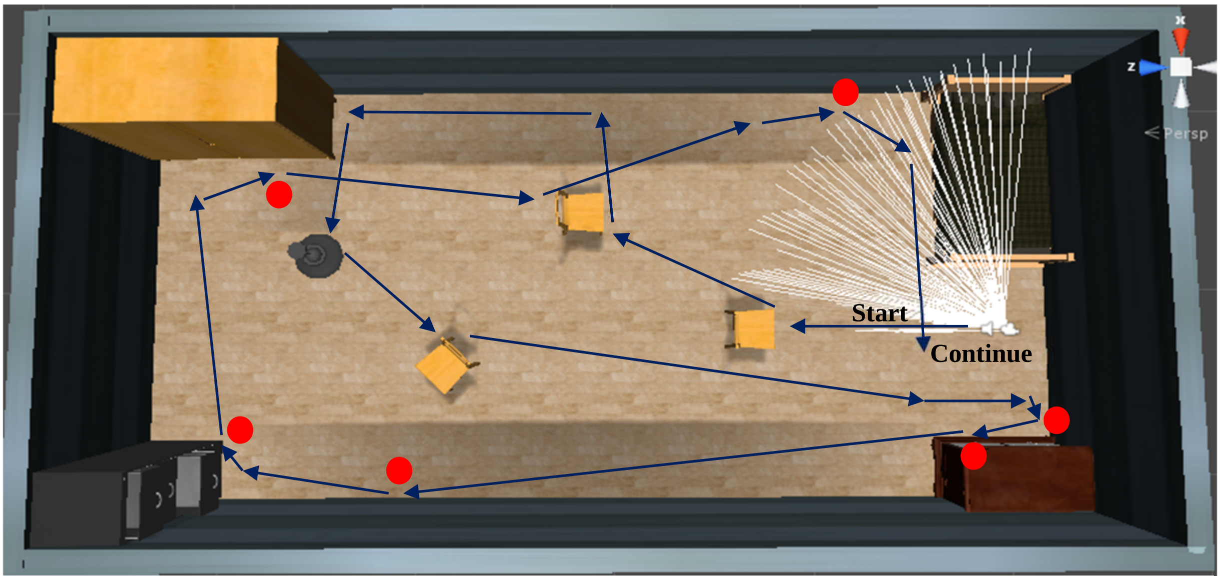


Figure 9.8: The behaviour of the avatar with VF defects in the left-side

● Object is bumped

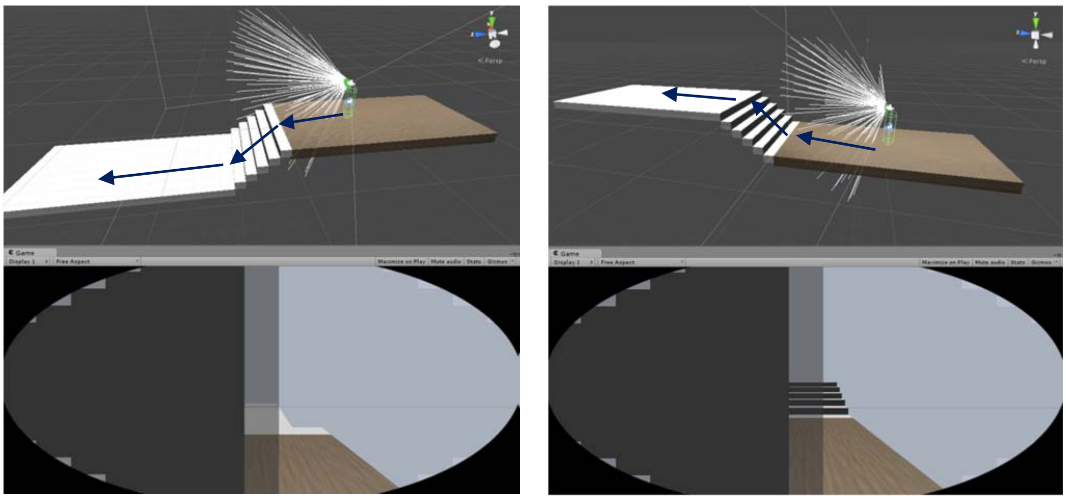
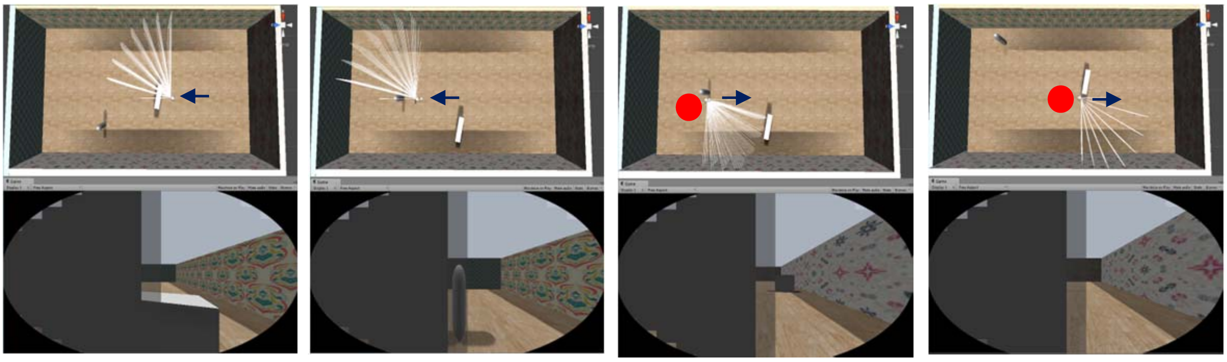


Figure 9.9: The behaviour of the avatar with VF defects in the left-side going up and down the stairs without problem



(a) (b) (c) (d)

Figure 9.10: The behaviour of the avatar with VF defects in the left-side navigating room scene among moving objects. (a) Cube is seen and stop action has been taken, (b) Capsule is seen and stop action has been taken, (c and d) Capsule and cube are not seen and bumped with as they approaching from the unseen left-side

● **Object is bumped**

9.5 Simulation of Superior VF Defects

Patients with superior VF defects can navigate environments without bumping into things, including going up and down. However, they would bump into hanging objects like lamps and tree branches. Figures 9.11, 9.12 and 9.13 show the behaviour of the avatar moving in different scenes.

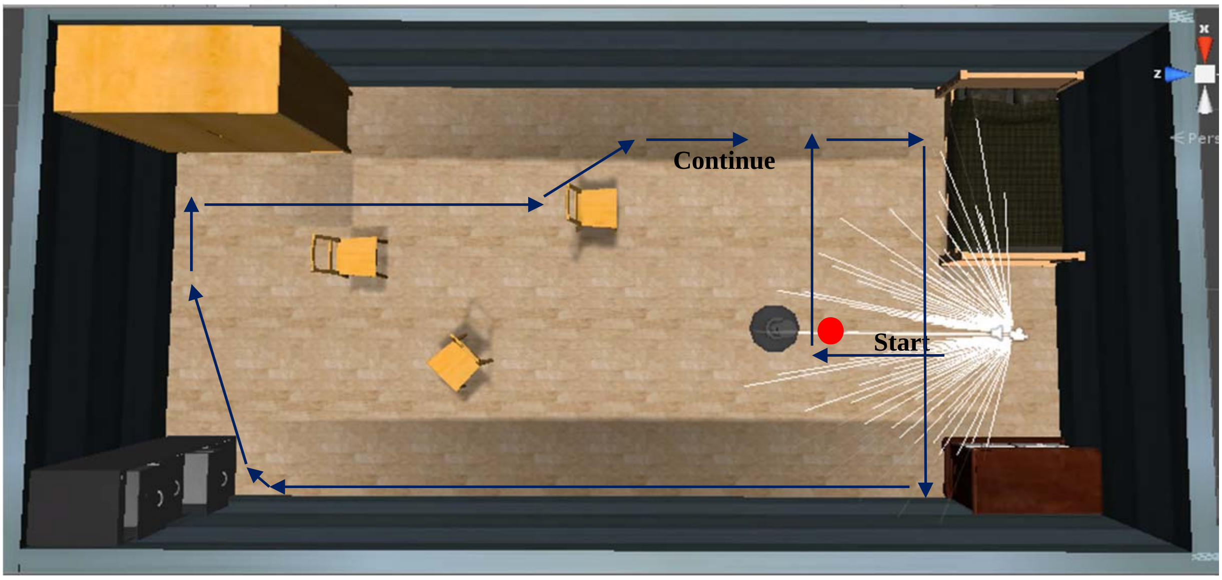


Figure 9.11: The behaviour of the avatar with superior VF defects

Object is bumped

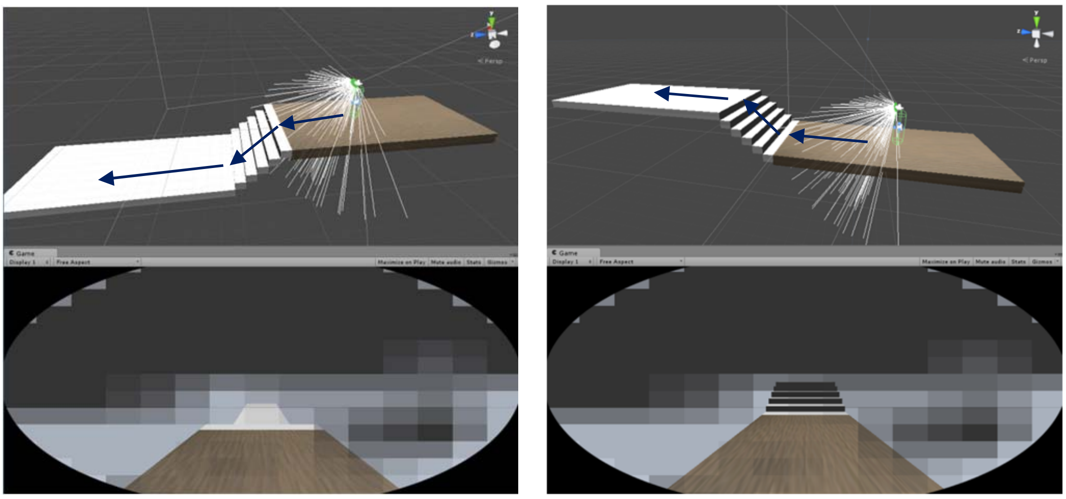


Figure 9.12: The behaviour of the avatar with superior VF defects going up and down the stairs without problem

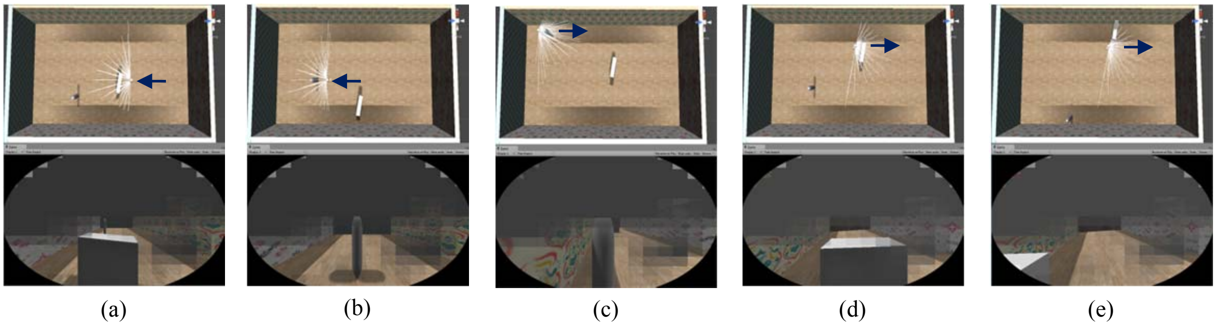


Figure 9.13: The behaviour of the avatar with superior VF defects navigating room scene among moving objects. (a, b, c and d) Cube and capsule are seen and stop action has been taken and (e) Cube is seen and running action has been taken

9.6 Simulation of Inferior VF Defects

The avatar with inferior VF defect cannot see low objects. It can, however, see high and hanging objects and is able to avoid them. Figures 9.14, 9.15 and 9.16 illustrate the simulated behaviour.

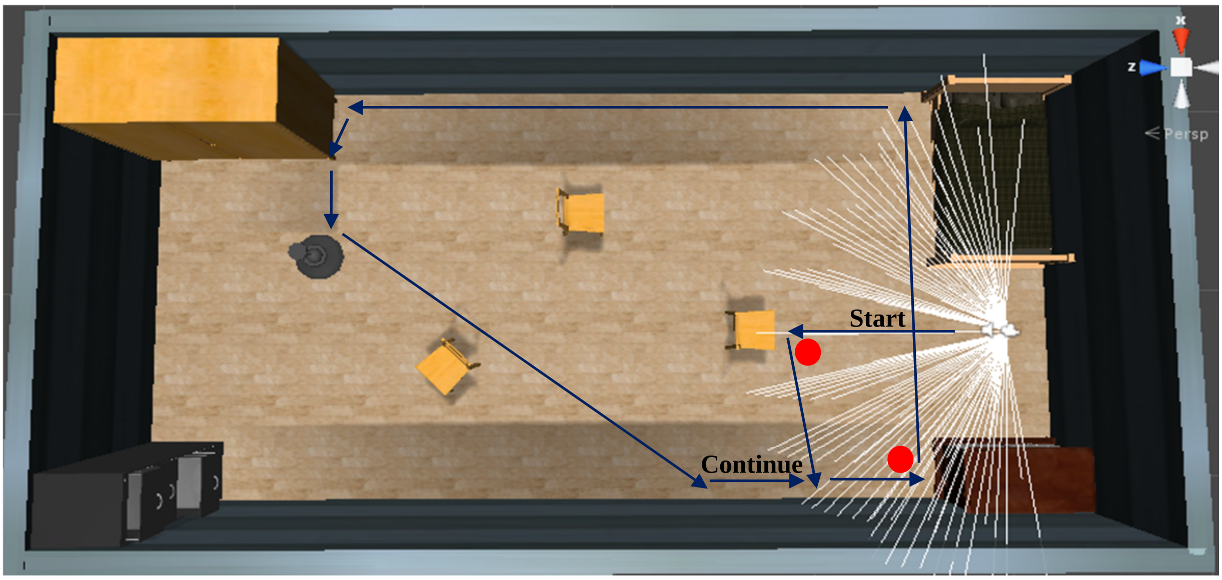


Figure 9.14: The behaviour of the avatar with inferior VF defects

Object is bumped

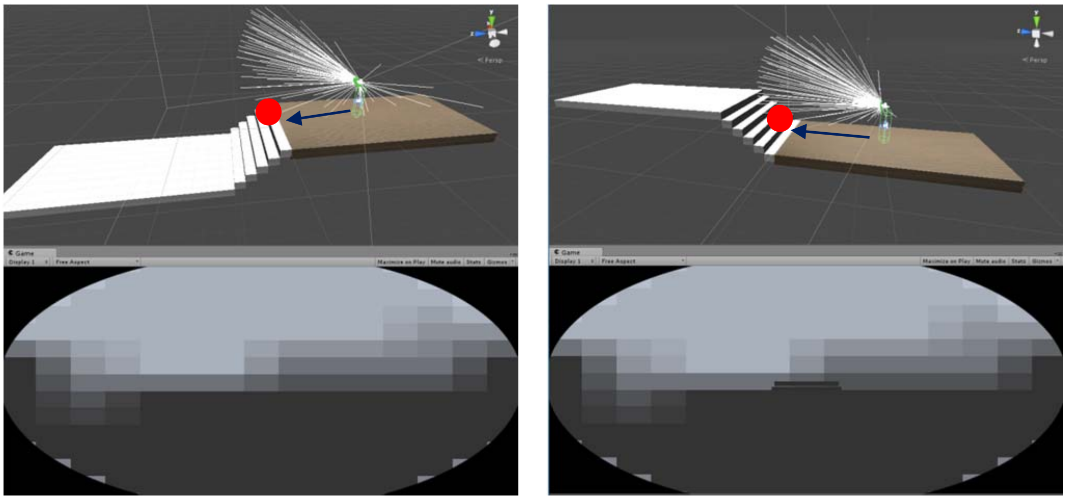
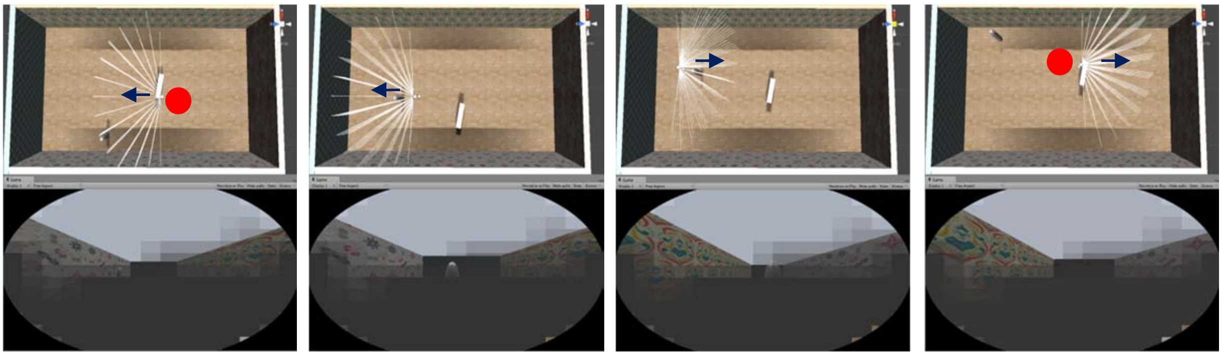


Figure 9.15: The behaviour of the avatar with inferior VF defects going up and down the stairs

● **Stairs is not seen**

The avatar with inferior VF defects has major problems with stairs. It would bump into the stairs when going up and would fall over the stairs when going down.



(a) (b) (c) (d)

Figure 9.16: The behaviour of the avatar with inferior VF defects navigating room scene among moving objects. (a and d) Cube is not seen and bumped with and (b and c) Capsule is seen and stop action has been taken

● **Object is bumped**

9.7 Simulation of the Most Defected VF

The VF pattern in Figure 9.1(f) is very serious defected. With this pattern, the avatar bumps into low, high and hanging objects. It is only able to avoid high objects on the upper right-side of its visual field. Figures 9.17, 9.18 and 9.19 illustrate this behaviour.

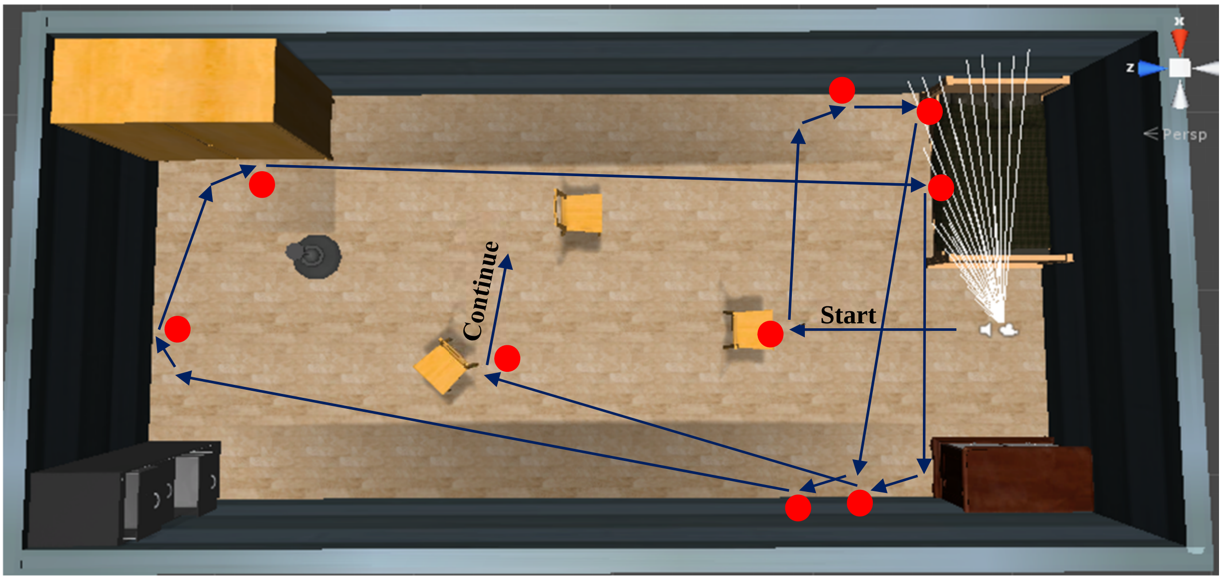


Figure 9.17: The behaviour of the avatar with most defected VF

● **Object is bumped**

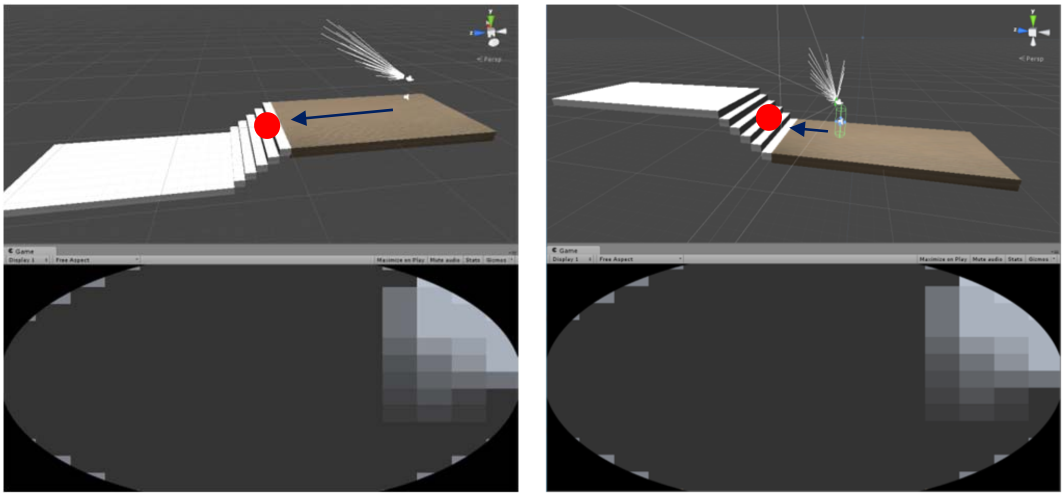
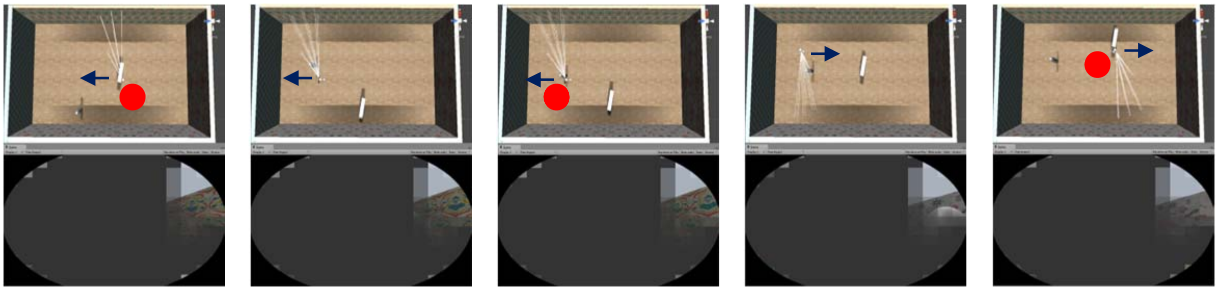


Figure 9.18: The behaviour of the avatar with most defected VF going up and down the stairs

● **Stairs is not seen**

The avatar with most defected VF has problems with stairs. It would bump into the stairs when going up and would fall over the stairs when going down.



(a) (b) (c) (d) (e)

Figure 9.19: The behaviour of the avatar with most defected VF navigating room scene among moving objects. (a and e) Cube is not seen and bumped with (b) Capsule is seen and stop action has been taken, but (c) Capsule is no longer seen and bumped with and (d) Capsule is seen and running action has been taken

● **Object is bumped**

The avatar bumps into both capsule and cube when they are on the left and centre of the defected VF.

9.8 Summary and Conclusions

Different scenarios and different VF defects are simulated using the avatar. The simulations show different behaviours of the avatar during the navigation. The avatar detects and turning away from the visible objects or bumping with the unseen objects. High objects are seen within all visual patterns but the superior VF defects, while the low and small objects are seen through all visual patterns except the inferior and most defected VF. In addition, the avatar went up and down the stairs without any problem except when these stairs are not seen by the inferior VF and most defected VF. In these cases, the avatar either bump to the upstairs or fall over the downstairs.

In conclusion, the avatar behaved as expected demonstrating the kind of problems that patients have similar defects would face in those situations. The simulations show reasonable behaviours of the avatar as it uses a computational intelligent algorithm for its navigation.

CHAPTER 10: EXPERIMENT IMPLEMENTATION AND RESULTS DISCUSSION

10.1 Introduction

Computer simulations are increasingly being used to solve problems and to aid in decision-making. Results from simulation models are validated through experiments (Sargent R.G., 2011). Accordingly, the aim of the experiment in this project is to evaluate the avatar simulation by comparing its behaviour with the behaviour of the participants in the experiment navigating the same virtual scenes. Besides that, the experiment is to let the people with healthy vision experience what these CVI patient are encounter during their navigation.

10.2 Overview of Experiment

The experiment is based on projecting the impaired VF onto an HTC VR Vive to be worn by real persons (participants). The participants with the projected VF defects imitate the real CVI patients during their navigation. The experiment gives the opportunity to people who are not visually impaired to experience what CVI patients encounter by navigating the same virtual environments with static or moving objects that the avatar moved around in the simulation. The results obtained from the experiment (real people behaviour with simulated CVI vision) and the results obtained from the avatar simulation (avatar behaviour) were compared. Therefore, the experiment was designed and carried out to validate the navigation behaviour of the avatar.

10.3 Experiment Design

The experiment is set to let the participants navigate predesigned virtual scenes which are same scenes that the autonomous avatar navigated in the simulation. The experiment uses an HTC Virtual Reality Vive Headset to be worn by the participants and to let them navigate virtual scenes after projecting the VF defects onto the Vive. The experiment consists of three parts: The virtual scenes, the participants and the virtual body of a participant in the scene.

10.3.1 Virtual Scenes

The scene is the virtual environment designed for the participants' navigation. It includes virtual static or moving assets and objects like chairs, bed, walls, etc. While a scenario includes a scene with either static or moving object or includes a scene of either going upstairs or going downstairs.

The same scenes that were used in the simulation for both scenarios of navigating a 3D environment with static objects and with moving objects are designed and set for the experiment. Such that, all objects in the experiment's scenes are set to the same positions in the simulation's scenes.

Figure 10.1 shows the same environment (room) is used with four different layouts of the obstacles (furniture) which provide 4 different scenes. It also explains another three different layouts for three different moving objects scenes. In the first and second arrangements, two moving objects were used to continue travelling from one wall to the opposite wall and get back again. The first object represents a car or a bike and the second object represents a person. Both are moving cross the participants' way. The third arrangement used for moving objects that move together simulating a group of people moving together. Different speeds were set for the moving objects.

The length and width of the room used in the virtual scenes are scaled to 4m and 3.5m respectively of a real physical room. Although the same room is used for all the scenes, the furniture arrangements are different. Therefore, each scene is a new situation for the participants.

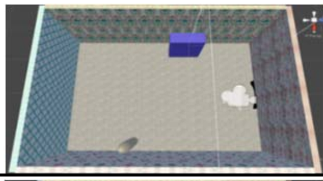
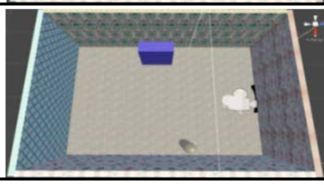
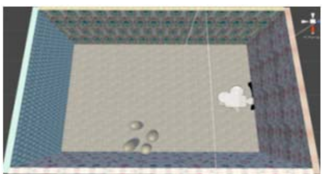
Static objects	Layout 1	
	Layout 2	
	Layout 3	
	Layout 4	
Moving objects	Layout 1	
	Layout 2	
	Layout 3	

Figure 10.1: The scenes used in the 3D environment with *static* and *moving* objects

10.3.2 Participants

As the aim of the experiment is to let the individuals explore and experience what it is like to have similar impairment. These individuals (participants) will explore exactly how the CVI patients see the environment and experience the difficulties that face the patients during their navigation. Therefore, ten adult participants with good eye sights from both sexes and different ages took part in the experiment. Table 10.1 shows the age and sex of the participants.

Table 10.1: The age and sex of the participants

No.	Gender	Age (Yrs)
1	Female	27
2	Male	36
3	Male	33
4	Male	35
5	Female	45
6	Male	46
7	Female	36
8	Female	40
9	Female	38
10	Male	35
Sex%	**50% Males**	**Age average is 37.1 years**
	50% Females	

All participants were given an information sheet that includes information about the study and the experiment. They were also have an instruction sheet that explains what they are expected to do. After the participants have read all the information and agreed to take part then they signed the consent form.

The participants were asked to navigate the virtual scenes with simulated visual impairment. They were asked to look forward and not to make any head or eye movements vertically or horizontally.

If a participant carried out any undesirable action, such as moving their head to the left and the right or moving their heads up and down to explore the paths, then the experiment was repeated.

Each participant's navigation behaviour in each scene was recorded as a video of MP4 format. Similarly, the avatar's simulation using the same scenes were captured in videos too. Their behaviours are analysed and compared.

10.3.3 Participant in the Scene

The avatar in the simulation is removed. The avatar's camera is replaced with the HTC Vive headset view. The body of the avatar is replaced by a capsule-shaped object with the Vive's camera attached to it on the top. An algorithm is applied to adjust the height of the capsule to match the height of the participant in the experiment. The capsule is positioned to touch the ground in the virtual scene so that as the participant moves the capsule (the simulated participant in the scene) moves along the ground too. Whenever the capsule collides with an object a sound will be made to clarify that a collision has occurred.

Besides that, an audio clip is attached to the capsule, and it is played when the participant body (capsule) enter a collision (hit) with objects in the scene. The sound simulates the sign of bumping occurrence. Figure 10.2 shows the pseudo code for initializing and adjusting the virtual body of the participants.

```
In Awake event function:
    Find the game object "body" in the scene: the virtual body of the participant.
    Find the game object "main ground" in the scene: the virtual floor.
In Start event function:
    Get the audio source component to be played on bumping occurrence.
In Update event function:
    Get the x and y-axis positions of the virtual body of the participant.
    Get the y-axis position of the virtual floor.
    Calculate the distance of the y-axes between the virtual body and the virtual floor.
    Set the local height of the virtual body to the calculated distance.
    Increase the y-axis position of the virtual body by the old y-axis position of the virtual
      body minus the calculated distance.
```

Figure 10.2: Pseudo code for initializing and adjusting the virtual body of the participants

10.4 Experimental Tasks

The participants were asked to go through two tasks:

Task one: navigate four virtual room scenes with the furniture arranged differently in each scene so that a participant cannot navigate around the room from memory. On each scene, a different VF pattern is projected onto the VR Vive. Thus the participants will experience different VF patterns projected within different room scenes. Each participant is given at least 50 seconds to navigate a scene so that the entire room could be explored.

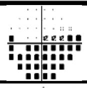
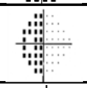
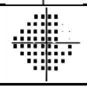
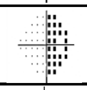
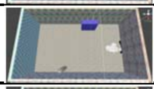
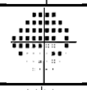
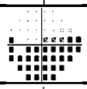
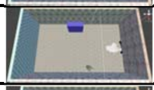
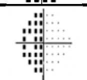
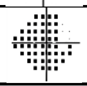
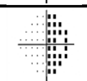
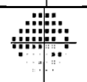
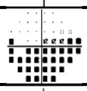
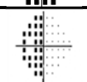
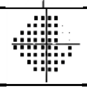
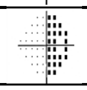
Task two: navigate three virtual room scenes with moving objects. These objects in different shapes could represent different things that move in a real situation. Each participant is asked to move from one end of the room to the other and then back again. This gives the participant the chance to experience the moving objects coming from the left and right sides.

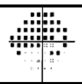
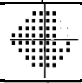
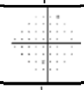
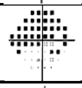
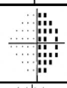
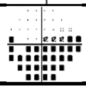
The differences in the scenes and the arrangement of the objects in both scenes would prevent the participants from navigating from memory.

10.5 Experiments and Data Collection

The seven scenes (4 scenes with static objects and 3 scenes with moving objects) are experimented with using 21 VF defects (see Table 10.2). Each participant carried out 21 trials so a total of 210 trials were carried out. Table 10.2 shows the type of the VF pattern projected onto a specific scene.

Table 10.2: The projected VF patterns onto scenes

No.	Type of impairment	VF pattern	Scene	Type of the scene's objects	No. of participants
1	Inferior			Moving objects Arrangement 1	10
2	Left				10
3	Most				10
4	Right				10
5	Superior				10
6	Inferior			Moving objects Arrangement 2	10
7	Left				10
8	Most				10
9	Right				10
10	Superior				10
11	Inferior			Moving objects Arrangement 3	10
12	Left				10
13	Most				10
14	Right				10

15	Superior				10
16	Most			Static objects Arrangement 1	10
17	Normal				10
18	Left			Static objects Arrangement 2	10
19	Superior				10
20	Right			Static objects Arrangement 3	10
21	Inferior			Static objects Arrangement 4	10

During the participant navigation, screen capture software (Share X) was used to record the navigation in MP4 video format.

Each video is analysed to identify when the participant had avoided an object or collided with an object.

10.6 Results and Discussion

An example of the analysis results is shown in table 10.3; green indicates object avoidance and red indicates object collision. The colour white is used to indicate that an object was not on the participant’s route (not visited). The behaviour of the autonomous avatar is shown in the last raw of the table.

A mathematical formula was set to measure the matching between their behaviour:

$$\text{Matching average for all participants} = \frac{\sum_{i=1}^{10} \text{No of (match + not visited)behaviour of Participant}_i}{\text{Total no of participants}} \quad \ldots(1)$$

$$\text{Percentage of matching accuracy} = \frac{\text{Average of matching}}{\text{no of obstacles on the routs}} * 100 \quad \ldots(2)$$

"Number of matching behaviours" is the matching event of the avoidance or bumping events in the same colours for the virtual avatar and real-world participant. "Matching accuracy" presents the same action events over total action events on the navigation routes of the avatar and participant.

This mathematical formula is subjected to a constraint. The matching of the behaviour between the avatar and each participant at each obstacle was taken into consideration if both the avatar and the participant visited the obstacle (the obstacle was within their navigation route). If one of them did not reach the obstacle then none of their behaviours is taken into consideration as the other behaviour is unknown.

The following tables show the behaviour of the participants and the avatar navigating the same scene. The index number of the carrying out experiment is shown in Table 10.2.

Table 10.3: Results of *Most* defected VF simulation in *Room1* (*Static Objects*)
Experiment index number in Table 10.2 is 16
The percentage of matching is 99.2%

Participant No.	Objects Name											
	Chair1	Chair2	Chair3	Lamp	File cabinet	Wardrobe	Bed	Drawers	Right wall	Left wall	Front wall	Back wall
1	bump		bump	bump	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
2	bump	bump	bump	bump	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
3		bump	bump	bump	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
4	bump	bump	bump	bump	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
5	bump	bump	bump	bump	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
6	bump	bump	bump	avoid		avoid	bump	bump	avoid	avoid	avoid	avoid
7	bump		bump	bump	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
8	bump	bump	bump	bump	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
9	bump	bump	bump	bump	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
10	bump	bump	bump	bump		avoid	bump	bump	avoid	avoid	avoid	avoid
Avatar	bump		bump	bump	bump	avoid	bump	bump	avoid	avoid	avoid	avoid

In Table 10.3, the avatar behaviour is matching the participants' behaviour during their navigating with the projection of the most of their VF is defected. The matching percentage is 99.2%. It is noticeable that all big and high obstacles are seen and the behaviour at big objects is consistent, but the small obstacles are not.

For the normal visual field (Table 10.4), all obstacles are detected and recognised by both the avatar and the participants. Their behaviour matching is 100%.

Table 10.4: Results of *Normal* VF simulation in *Room1* (*Static Objects*)
Experiment index number in Table 10.2 is 17
The percentage of matching is 100%

Participant No.	Objects Name											
	Chair1	Chair2	Chair3	Lamp	File cabinet	Wardrobe	Bed	Drawers	Right wall	Left wall	Front wall	Back wall
1	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
2	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
3	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
4	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
5	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
6	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
7	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
8	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
9	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
10	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
Avatar	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid

In Table 10.5, as expected, people with left defected VF can detect obstacles on their right side of their vision. The avatar behaviour matches most of the participants' behaviour except the walls detections. This refers to the way of the participants thinking. People use their intelligence during their navigation. If they detect an obstacle ahead, and if the obstacle will remain in their defected side after they turning away off it, they will step aside enough away from the obstacle and continue their navigation. This intellectual estimation of the distance away from the obstacle is done by all the participants while the avatar used the available angle on its right side. When the avatar faced an obstacle, it turns away to the right and during its turning the wall will be unseen. Thus, when the avatar continues to move ahead it will bump into the wall from its unseen left side. This what the people avoid using their intelligence. Accordingly, the avatar behaviour matches the participants' behaviour except for this case. The matching degree is 83.3%. It is low because of the virtual avatar is not good enough to deal with the wall when having left-side visual defects.

Table 10.5: Results of *Left* VF defects simulation in *Room2* (*Static Objects*)
Experiment index number in Table 10.2 is 18
The percentage of matching is 83.3%

Participant No.	Objects Name																							
	Chair1		Chair2		Chair3		Lamp		File cabinet		Wardrobe		Bed		Drawers		Right wall		Left wall		Front wall		Back wall	
	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right
1	B	A	B	A		A	B	A		A	B	A	B	A	B	A	A	A	A	A	A	A	A	A
2	B	A	B	A		A	B	A		A	B	A	B	A	B	A	A	A	A	A	A	A	A	A
3	B	A		A		A	B	A		A		A	B	A	B	A	A	A	A	A	A	A	A	A
4	B	A	B	A		A	B	A		A	B	A		A	B	A	A	A	A	A	A	A	A	A
5	B	A	B	A		A	B	A		A	B	A	B	A	B	A	A	A	A	A	A	A	A	A
6		A	B			A	B	A		A	B	A	B	A	B	A	A	A	A	A	A	A	A	A
7	B	A	B	A		A	B	A		A	B		B	A	B	A	A	A	A	A	A	A	A	A
8	B			A		A	B	A		A		A		A	B	A	A	A	A	A	A	A	A	A
9	B	A	B	A		A	B	A		A	B	A	B	A	B	A	A	A	A	A	A	A	A	A
10	B	A	B			A	B	A		A		A	B	A	B	A	A	A	A	A	A	A	A	A
Avatar		A		A		A	B	A	B	A	B	A	B	A	B	A	B	A	B	A	B	A	B	A

In the following Table 10.6, people with superior defected VF can recognise low obstacles and the bottom of the high obstacles. Therefore, all obstacles in the simulations are seen and be avoided by the participants and the avatar. But, they couldn't detect the hanging over obstacle (lamp) as it lies into their defected part. The avatar and the participant's behaviour are 100% matching.

Table 10.6: Results of *Superior* VF defects simulation in *Room2* (*Static Objects*)
Experiment index number in Table 10.2 is 19
The percentage of matching is 100%

Participant No.	Objects Name											
	Chair1	Chair2	Chair3	Lamp	File cabinet	Wardrobe	Bed	Drawers	Right wall	Left wall	Front wall	Back wall
1	avoid	avoid	avoid	bump	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
2	avoid	avoid	avoid	bump	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
3	avoid	avoid	avoid	bump	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
4	avoid	avoid	avoid	bump	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
5	avoid	avoid	avoid	bump	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
6	avoid	avoid	avoid	bump	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
7	avoid	avoid	avoid	bump	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
8	avoid	avoid	avoid	bump	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
9	avoid	avoid	avoid	bump	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
10	avoid	avoid	avoid	bump	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
Avatar	avoid	avoid	avoid	bump	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid

The same situation in the left defected VF is also applied to the right defected VF. Table 10.7 shows the mismatching the avatar bumping into walls against the participants' avoidance. Again, people stepped away enough from the walls but the avatar used its available angle on its left side. Until the wall is not seen, the avatar bumped into the walls from its unseen right side. That's why the avatar's right side indicators are red while the participants' are green. The avatar and the participants' behaviour are matching around 83.3%.

Table 10.7: Results of *Right* VF defects simulation in *Room3* (*Static Objects*)
Experiment index number in Table 10.2 is 20
The percentage of matching is 83.3%

Participant No.	Objects Name																							
	Chair1		Chair2		Table		Lamp		File cabinet		Wardrobe		Bed		Drawers		Right wall		Left wall		Front wall		Back wall	
	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right
1	A	B		B	A	B	A	B	A	B	A	B	A		A		A	A	A	A	A	A	A	A
2		B	A		A	B	A			B	A	B	A	B	A	B	A	A	A	A	A	A	A	A
3	A	B	A			B		B	A		A	B	A	B	A	B	A	A	A	A	A	A	A	A
4	A	B	A	B	A	B		B	A		A	B	A	B	A	B	A	A	A	A	A	A	A	A
5	A	B	A	B		B	A		A			B		B		B	A	A	A	A	A	A	A	A
6		B	A	B	A	B			A		A	B	A	B	A		A	A	A	A	A	A	A	A
7	A	B	A		A			B	A		A	B	A		A		A	A	A	A	A	A	A	A
8	A	B	A	B		B	A		A			B		B		B	A	A	A	A	A	A	A	A
9	A		A	B		B	A	B	A		A	B		B			A	A	A	A	A	A	A	A
10		B	A	B		B	A	B	A	B	A		A	B	A	B	A	A	A	A	A	A	A	A
Avatar	A	B	A	B	A	B	A	B		B	A	B	A	B	A	B	A	B	A	B	A	B	A	B

According to Table 10.8, the high and tall obstacles are recognised by both the participants and the avatar with the projection of inferior defects, while the low obstacles are not seen and bumped with. The table shows the exact matching between their behaviour with 100% match.

Table 10.8: Results of *Inferior* VF defects simulation in *Room4* (*Static Objects*)
Experiment index number in Table 10.2 is 21
The percentage of matching is 100%

Participant No.	Objects Name								
	Lamp	Table	Wardrobe	Bed	Drawers	Right wall	Left wall	Front wall	Back wall
1	avoid	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
2	avoid	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
3	avoid	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
4	avoid	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
5	avoid	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
6	avoid	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
7	avoid	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
8	avoid	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
9	avoid	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
10	avoid	bump	avoid	bump	bump	avoid	avoid	avoid	avoid
Avatar	avoid	bump	avoid	bump	bump	avoid	avoid	avoid	avoid

For the moving object scenes, the simulation results had been monitored and assigned when a participant navigated forward and backwards in the scene. The obstacle avoidance or bumping is indicated in the tables (10.9 to 10.13) for both directions.

Table 10.9 shows the results of the simulations for both the avatar and the participants with the inferior defected VF navigating the room with moving obstacles. As the capsule in the scene is a high enough, a patient with inferior defected is able to recognise it. In the other hand, the cube in the same scene is low enough to be not recognised. Thus, the participant/avatar is not able to see this cube and bump with it. The navigation behaviour of both the participants and the avatar is 100% matching.

Table 10.9: Results of *Inferior* VF defects simulation into all rooms (*Moving Objects*)
Experiments index numbers in Table 10.2 are 1, 6 and 11
The percentage of matching is 100%

Participant No.	Scene 1				Scene 2				Scene 3	
	Objects Name				Objects Name				Objects Name	
	Cube		Capsule		Capsule		Cube		Capsules	
	Going	Return	Going	Return	Going	Return	Going	Return	Going	Return
1	bump	bump	avoid	avoid	avoid	avoid	bump	bump	avoid	avoid
2	bump	bump	avoid	avoid	avoid	avoid	bump	bump	avoid	avoid
3	bump	bump	avoid	avoid	avoid	avoid	bump	bump	avoid	avoid
4	bump	bump	avoid	avoid	avoid	avoid	bump	bump	avoid	avoid
5	bump	bump	avoid	avoid	avoid	avoid	bump	bump	avoid	avoid
6	bump	bump		avoid	avoid	avoid	bump	bump	avoid	
7		bump	avoid	avoid	avoid	avoid	bump	bump	avoid	avoid
8		bump	avoid	avoid	avoid	avoid		bump	avoid	avoid
9	bump	bump	avoid	avoid	avoid	avoid	bump		avoid	avoid
10		bump	avoid	avoid	avoid	avoid	bump	bump	avoid	avoid
Avatar	bump	bump	avoid	avoid	avoid	avoid	bump	bump	avoid	avoid

Table 10.10: Results of *Left* VF defects simulation into all rooms (*Moving Objects*)
Experiments index numbers in Table 10.2 are 2, 7 and 12
The percentage of matching is 100%

Participant No.	Scene 1				Scene 2				Scene 3	
	Objects Name				Objects Name				Objects Name	
	Cube		Capsule		Capsule		Cube		Capsules	
	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right
1	bump		bump			avoid	bump	avoid	bump	avoid
2	bump	avoid		avoid	bump	avoid	bump	avoid	bump	avoid
3	bump	avoid		avoid	bump	avoid		avoid	bump	avoid
4	bump		bump		bump	avoid		avoid		avoid
5	bump	avoid		avoid	bump	avoid	bump	avoid	bump	avoid
6	bump	avoid	bump	avoid	bump	avoid	bump	avoid	bump	avoid
7	bump	avoid		avoid	bump	avoid	bump	avoid	bump	avoid
8		avoid	bump	avoid	bump		bump	avoid	bump	avoid
9	bump	avoid	bump	avoid	bump	avoid	bump	avoid	bump	avoid
10	bump	avoid	bump	avoid	bump	avoid		avoid	bump	avoid
Avatar	bump	avoid	bump	avoid	bump	avoid	bump	avoid	bump	avoid

The behaviour of the participants with left defected VF (see the above Table 10.10) is that they can recognise the moving obstacles approaching from the right side but they couldn't recognise their approach from their defected side. Their behaviour is 100% matching.

According to the projection of the most defected VF pattern and as the clear visual area is located on the upper right side, some of the participants saw the top of the moving obstacle (capsule) in their forward or backwards routes. For the avatar, the capsule navigates the entire scene by travelling from the left wall to the right wall. The avatar detected the capsule due to the appearance of the capsule near the wall in the same moment the avatar is navigating near the wall and the capsule will be seen. This situation makes the avatar stops to avoid bumping. Thus, the matching was 85%. See Table 10.11 for the most defected VF.

Table 10.11: Results of *Most* defected VF simulation into all rooms (*Moving Objects*)
Experiments index numbers in Table 10.2 are 3, 8 and 13
The percentage of matching is 85%

Participant No.	Scene 1				Scene 2				Scene 3	
	Objects Name				Objects Name				Objects Name	
	Cube		Capsule		Capsule		Cube		Capsules	
	Going	Return	Going	Return	Going	Return	Going	Return	Going	Return
1	bump		avoid	bump	bump		bump	bump	bump	avoid
2	bump	bump		bump	bump	bump	bump	bump	bump	avoid
3		bump		bump	bump	bump	bump	bump	bump	avoid
4	bump	bump		bump		bump	bump	bump	bump	avoid
5	bump	bump	avoid	bump		bump	bump			bump
6	bump	bump	avoid	avoid	bump	avoid	bump	bump	bump	bump
7		bump	bump	bump	avoid		bump	bump	avoid	bump
8	bump	bump	bump	avoid	bump	bump	bump		bump	bump
9	bump	bump		bump	avoid	avoid	bump	bump	bump	bump
10	bump	bump	avoid	bump	avoid	avoid	bump			bump
Avatar	bump	bump	avoid	bump	bump	bump	bump	bump	bump	bump

The behaviour of the people with right defected VF is predictable as they can avoid obstacles from the left side and about to bump with others on the right side. Table 10.12 shows the matching behaviour of the participants and the avatar and the matching is 100%.

Table 10.12: Results of *Right* VF defects simulation into all rooms (*Moving Objects*)
Experiments index numbers in Table 10.2 are 4, 9 and 14
The percentage of matching is 100%

Participant No.	Scene 1				Scene 2				Scene 3	
	Objects Name				Objects Name				Objects Name	
	Cube		Capsule		Capsule		Cube		Capsules	
	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right
1	avoid	bump	avoid	bump	avoid	bump	avoid	bump		bump
2	avoid			bump	avoid	bump	avoid	bump		
3	avoid	bump	avoid		avoid		avoid	bump		bump
4	avoid	bump	avoid	bump	avoid	bump	avoid	bump		bump
5	avoid		avoid	bump		bump	avoid	bump		
6	avoid	bump	avoid	bump	avoid		avoid	bump		bump
7	avoid	bump	avoid		avoid	bump	avoid			bump
8	avoid	bump	avoid	bump	avoid	bump	avoid			
9	avoid	bump	avoid	bump	avoid	bump	avoid			bump
10	avoid	bump	avoid		avoid	bump	avoid	bump		bump
Avatar	avoid	bump	avoid	bump	avoid	bump	avoid	bump	avoid	bump

Superior deficiency allows the patients to see and recognise the surrounding lies in their lower VF. Table 10.13 shows the behaviour of both is matching 100%.

Table 10.13: Results of *Superior* VF defects simulation into all rooms (*Moving Objects*)
Experiments index numbers in Table 10.2 are 5, 10 and 15
The percentage of matching is 100%

Participant No.	Scene 1				Scene 2				Scene 3	
	Objects Name				Objects Name				Objects Name	
	Cube		Capsule		Capsule		Cube		Capsules	
	Going	Return	Going	Return	Going	Return	Going	Return	Going	Return
1	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
2	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
3	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
4	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
5	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
6	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
7	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
8	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
9	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
10	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid
Avatar	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid	avoid

The overall matching percentage for the scenarios with either the static or the moving objects with the overall percentage of the whole experiments is shown in the following Table 10.14.

A 94.3% of matching behaviour between the avatar and the participants during their navigation rooms with static obstacles. While the matching behaviour percentage between them during their navigation rooms with moving obstacles is 97%. Accordingly, the overall matching behaviour during their navigation all the scenes using all VF defects projection is 95.53%.

Table 10.14: The matching percentage for scenarios with static/moving objects

	Kind of the defected VF	Room scene	Matching percentage	Overall %
Scenarios with static obstacles	Most	1	99.2%	94.3%
	Normal	1	100%	
	Left	2	83.3%	
	Superior	2	100%	
	Right	3	83.3%	
	Inferior	4	100%	
Scenarios with moving obstacles	Inferior	1	100%	97%
		2		
		3		
	Left	1	100%	
		2		
		3		
	Most	1	85%	
		2		
		3		
	Right	1	100%	
		2		
		3		
	Superior	1	100%	
		2		
		3		
Overall percentage of all matching				**95.53%**

10.7 Summary and Conclusions

The aim of this chapter is to validate the avatar's behaviour of the proposed system by setting an experiment to match the avatar's behaviour with the participants' behaviour that navigates the same scenarios. The experiments were carried out using the HTC Oculus Vive headset. The navigated virtual environments were designed as the same as the environments in the proposed system that uses the avatar. Twenty-one scenarios (including 21 VF patterns and 7 testing scenes) for different scenes were tested by 10 participants. The participants were asked to keep navigating each scene for one minute approximately to visit most of the objects in the scene. Besides, they had been asked to focus their sight to the centre and not to make any head movements (left, right, up, down) and not to make any of the eye-side-movements.

The navigation behaviour of each participant was recorded as a video MP4 film to assign the similarities and the differences between the avatar's and each participant's behaviour for matching their behaviour at each obstacle in the virtual scene.

The matching percentage of their behaviour was calculated using the proposed mathematical formula based on their behaviour at each obstacle they visited. The percentages show significant matching of 94.3% using virtual scenes with static obstacles and 97% using virtual scenes with moving obstacles. A couple of scenes have low percentages were carried out within two cases of the left or the right VF pattern. This was during the detecting of walls when they have visual defects in their left or right side. They gave a low accuracy of 83.3% and 83.3% respectively due to their differences in behaviour when avoiding the walls. The participants use their intellectual thinking to step enough away from walls but the avatar uses the available angle on its non-defected side. When the avatar faced an obstacle, it turns away to the non-defected side and during its turning, the wall will be unseen. Thus, when the avatar continues to move ahead it will bump into the wall. This what the people avoid using their intelligence.

In general, the results of the experiment show significant matchings in the behaviour of the participants and the avatar simulation in both scenarios (static and moving objects). The overall matching percentage was around 95.53%.

CHAPTER 11: SUMMARY, CONTRIBUTIONS AND FUTURE WORK

11.1 Summary

This thesis investigated the behaviour of navigation of the cerebral visual impairment patients by simulating their visual field into 3D virtual environments. The visual field deteriorated spots of the CVI patients can be gained from the visual field test's chart of Humphrey's Perimetry. The final visual probability symbolic patterns in the chart are applied in this project. Image processing techniques and morphological operators are used to segment and extract the symbolic patterns. They are then categorised using SVM, a supervised machine learning technique. They are scaled and saved as numeric values into a 2D array. These values are used by a projection system that mimics the patient visual field and how well the patient sees according to the results of the VF chart.

The clarity and blurring visual areas are determined by the probability of degree of deterioration as indicated on the VF chart. Thus, they determine the level of transparency and the amount of blurring in the different parts of the visual field according to the chart. Therefore, masks are designed with appropriate levels of blurring in PhotoShop using textures.

The game engine Unity3D is used for projecting the field of vision of the avatar given a VF pattern. The scenes for avatar simulation are designed using Unity3D and other 3D modelling software assets. The avatar in the game engine can imitate the vision of any patient if given their visual field chart results.

Novel algorithms have been designed and implemented to mimic the behaviour of people with different visual impairments. Scenarios were created to test the algorithms. The avatar demonstrates the expected behaviour when given different VF defects and in different scenes.

The final objective of the study is to enable people to experience what it is like to have a visual impairment of the kind discussed in the thesis. The VF is projected directly to the HTC Oculus Vive headset worn by the people. The projected distortion spots block the vision of the people. Thus, it reflects the original visibility of the imitated patient.

An experiment is set and implemented to assess the behaviour exhibit by participants using the HTC VIVE against the avatar simulation. The participants navigated virtual scenes as same as the avatar navigated. The participants' navigations are captured and compared with the avatar navigation. The results show significant matchings in the behaviour of the participants and the avatar. The matching percentage between the avatar and 10 participants' navigation is 84%.

11.2 Future Works

Every study and project has advantages and limitations. These limitations determine the proposed solutions as future works.

11.2.1 Real CVI Patients (or their Parents/Carers) to Validate the System

One limitation of the project is that the evaluation of the proposed system has not been carried out. The evaluation of the avatar simulation is done by proposing an experiment that involves healthy participants instead of real CVI patients. The participants wore an HTC Oculus Vive headset and the impaired visual field patterns are projected onto it to mimic the real patient's VF. Then, the navigation behaviour of the avatar is compared with the participants'. Therefore, carrying out an experiment with real patients with CVI is needed. It will clarify whether their navigation behaviours actually match the avatar simulations given the VF charts of those patients.

In the other hand, the parents or members of the CVI patient's family or the carers can take part in this experiment to experience and explore the limitation in the patient's vision and to use their navigation to evaluate the proposed avatar simulation.

11.2.2 Using Colour Contrast Along with the Visual Field Defects

The simulation in this work depends only on the visual field pattern, while the vision of CVI patients affected by other factors. As the visual field test uses of different flashing light contrast, thus the sensitivity of the patient's retina affected by these contrast (see section 2.1.1, page 9-10), and thus, each patient has different sensitivity to lights and consequently to different colour contrast. Therefore, a map of the colour contrast of the patient can be provided to the simulation system along with the VF pattern. Accordingly, some visual areas in the VF can be defected and block the vision relatively or completely, while other areas have low contrast to specific colours and the patient cannot recognize these colours and objects with these colours can be faded and relatively unseen.

In the other hand, patients with CVI have a rapid movement of the eye between fixation points, this is called eye saccades. Thus, another proposed input to the simulation is to mimic these eye movements by shaking the camera (avatar eyes) by the degree and the direction of the patient's saccades.

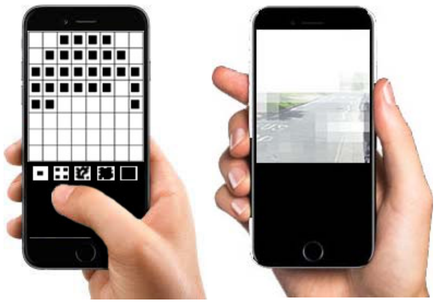
11.2.3 Speeding up the Algorithm Processing

The simulation of behaviour of navigation in this system depends only on computational techniques and on the representation of the inputs and all numeric values in form of matrices. Accordingly, the implementation of the proposed algorithm and all computations and calculations are done repeatedly at each frame of the simulation, meaning every step taken by the avatar. Besides, these calculations overload the running of the using software (Unity3D) for the simulation. Thus, the need to reduce these calculations to speed up the system is required. Therefore, all implementations of the algorithm sections needed to be optimised to speed up the algorithm and to reduce the time of doing these calculations.

11.2.4 Application for VF Projecting

The proposed system in this thesis uses virtual environment during the VF projections and navigation simulations. Therefore, the projection of the VF defects onto real world is needed for the parents, carers and educators to see realistically what the patient is seeing. Therefore, a proposed approach is to project the visual field symbols onto the cell phone screen while the rare camera is on. The vision masks will appear above the camera view and block parts of the view representing the deteriorated visual areas of the patient. The parents and carers can navigate their patient's local environment experiencing the actual seen and unseen visual spots and what the patient is really seeing. In case of the patient is a child, then it will exactly aid the understanding of the parents about their CVI child when the parents navigate their child's local environments.

The parent can input each of the VF symbols to its position on the cell phone screen and then the application generates the vision masks for them and projects them on the screen. Thus, the parent can see what his child is really seeing and knowing what their child's vision capabilities are, see figure 11.1.



(a) (b)

Figure 11.1: The proposed cell phone application. (a) Entering the visual symbols according to the visual field chart and (b) the projection of the visual field onto the main cell phone camera

REFERENCES

AISB society. (1964). What is Artificial Intelligence. Reviewed on (5 Feb 2016). Retrieved http://www.aisb.org.uk/public-engagement/what-is-ai

Andy Thé (2014, October 15). Image Processing Made Easy. *MathWorks*. [video file]. Retrieved from: www.uk.mathworks.com/videos/image-processing-made-easy-81718.html

Ashburner J. and Friston K. (2000). Voxel-based morphometry – the methods. *Neuroimage*. 11: 805-21.

Beckett J. (2016, October 27). This Powerful Wearable is a Life Changer for the Blind. *NVIDIA*. Reviewed on (28 April 2017). Retrieved from: www.blogs.nvidia.com/blog/2016/10/27/wearable-device-for-blind-visually-impaired/

Bello J.A., et al (2013, August). ACR–ASNR–Practice guideline for the performance and interpretation of magnetic resonance imaging (MRI) of the brain. *American College of Radiology*. Available from: http://www.acr.org/~/media/ACR/Documents/PGTS/guidelines/MRI_Brain.pdf

Bernas-Pierce J., Pollizzi J., Altmann C., Lee B., Hoyt C. (1998). Cortical visual impairment paediatric visual diagnosis fact sheet. SEE/HEAR. Blind Babies Foundation. Retrieved from: http://www.tsbvi.edu/seehear/fall98/cortical.htm

Cancer Research UK (2013, August 14). CT Scan. Retrieved from: http://www.cancerresearchuk.org/cancer-help/about-cancer/tests/ct-scan

Caon M. (2004). Voxel-based computational models of real human anatomy: a review. *Radiat Environ Biophys. Springer Verlag.* 42: 229-235

Carl Zeiss Meditec, Humphrey, HFA, STATPAC, Visual Field Index. A Guide to Interpretation for HFA II-*i* Single Field Analysis. USA, 2013. Retrieved on 27/7/2015, from: www.zeiss.com/content/dam/Meditec/us/download/Glaucoma%20Landing%20Page/hfasinglefieldguidehfa5268

Carroll J.N. and Johnson C.A. (2013, August 22). Visual field testing: from one medical student to another. *Ophthalmology and Visual Science.* Retrieved from:

http://webeye.ophth.uiowa.edu/eyeforum/tutorials/VF-testing/

Casteels I., Demaerel P., Spileers W., Lagae L., Missotten L. and Casaer P. (1997). Cortical visual impairment following perinatal hypoxia-clinicoradiologic correlation using magnetic resonance imaging. *Journal of Pediatric Ophthalmology and Strabismus.* 34(5): 297-305.

Cioni G., Fazzi B., Ipata A.E., Canapicchi R. and van Hof-van Duin J. (1996). Correlation between cerebral visual impairment and magnetic resonance imaging in children with neonatal encephalopathy. *Dev Med Child Neurol.* 38(2): 120-132.

Cortes C. and Vapnik N. (1995). Support-vector-networks. *Machine Learning*. 20(3):273. doi:10/1007/BF00994018.

Crabb D.P. (2014). Personal Communication, with Professor David Crabb, School of Health Sciences, Division of Optometry & Visual Sciences, City University, London, UK.

Csurka, G., Dance C., Fan L., Willamowski J. and Bray C. (2004). *Visual Categorization with Bags of Keypoints*. Workshop on Statistical Learning in Computer Vision. ECCV 1 (1–22): 1–2.

Duffy M. (2013, May 1). Google Glass Applications for Blind and Visually Impaired Users. *VisionAware Blog*. Reviewed on (18 Aug 2015). Retrieved from:
www.visionaware.org/blog/visionaware-blog/google-glass-applications-for-blind-and-visually-impaired-users/12

Dutton G.N. (2003). Cognitive vision, its disorders and differential diagnosis in adults and children: knowing where and what things are. *Eye. The Edridge Green Lecture 2002.* 17: 289-304.

Dutton G.N. (2008). Strategies for dealing with visual problems due to cerebral visual impairment. *Scottish Sensory Centre.* Retrieved from:
www.ssc.education.ed.ac.uk/courses/vi&multi/vjan08i.html
www.ssc.education.ed.ac.uk/courses/vi&multi/vjan08ii.html

Dutton G.N. (2009). Dorsal stream dysfunction and dorsal stream dysfunction plus: a potential classification for perceptual visual impairment in the context of cerebral visual impairment. *Developmental Medicine & Child Neurology.* 51(3): 170-172.

Dutton G.N., McKillop E.C.A., Saidkasimova S. (2006). Visual problems as a result of brain damage in children. *Br J Ophthalmol.* 90(8): 932-933.

Dutton G.N., Saeed A., Fahad B., et al. (2004). The association of binocular lower vision field impairment, impaired simultaneous perception, disordered visually guided motion and inaccurate saccades in children with cerebral visual dysfunction: a retrospective observational study. *Eye.* 18: 27-34.

EBU (2014, May 10). Facts, figures and definitions concerning blindness and sight loss. Retrieved from: http://www.euroblind.org/resources/information/#details

Glen F.C., Smith N.D. and Crabb D.P. (2013). Saccadic eye movements and face recognition performance in patients with central glaucomatous visual field defects. *Vision Research.* 82: 42-51.

Glen F.C., Smith N.D. and Crabb D.P. (2014). What types of visual field defects are hazardous for driving? *ARVO Annual Meeting Abstracts*. 2014. Program No: 3009. Page 3. Retrieved from: http://www.arvo.org/webs/am2014/abstract/sessions/327.pdf

Glen F.C., Smith N.D., Jones L. and Crabb D.P. (2016). I didn't see that coming: Simulated visual fields and driving hazard perception test performance. *Clinical and Experimental Optometry*. 99(5): 469-475. doi: 10.1111/cxo.12435

Good W.V., Jan J.E., deSa L., et al (1994). Cortical visual impairment in children: a major review. *Surv Ophthalmol*. 38(4): 351-364.

Goodale M., Milner A. (1992). Separate visual pathways for perception and action. *Trends Neurosci.* 15(1): 20-25.

Greco L. (2014, March 1). Accessibility in Education. *The Global Initiative for Inclusive ICTs*. Reviewed on (18 Aug 2015). Retrieved from: www.g3ict.org/resource-center/newsletter/news/p/id_515

Groenewald C. and Damato B. (2002, June 10). Visual Field Test. *Royal Liverpool University Hospital*. Retrieved from: http://testvision.org/

Groenveld M. (2004). Children with cortical visual impairment. *American Printing House for the Blind.* Retrieved from: http://www.aph.org/cvi/articles/groenveld_1.html

Grzybowski, A. (2009), Harry Moss Traquair (1875–1954), Scottish ophthalmologist and perimetrist. *Acta Ophthalmologica*. 87: 455–459. doi: 10.1111/j.1755-3768.2008.01286.x

Gupta M., Patel A., Dave N., Goradia R. and Saurin S. (2014). Text-Based Image Segmentation Methodology. *2nd International Conference on Innovations in Automation and Mechatronics Engineering. ICIAME.* 14:465-472. doi: 10.1016/j.protcy.2014.08.059. Retrieved from: www.sciencedirect.com/science/article/pii/S2212017314000954#

Harrell L. (2004). Cortical visual impairment – A challenging diagnosis. *American Printing House for the Blind.* Retrieved from:

http://www.aph.org/cvi/articles/harrell_1.html

Herman G.T. (2009). Fundamentals of computerized tomography: image reconstruction from projections. *Springer*. Retrieved from: http://books.google.co.uk/books?hl=en&lr=&id=BhtGTkEjkOQC&oi=fnd&pg=PA1&dq=fundamentals+of+computerized+tomography:+image+reconstruction+from+projections&ots=Rl8b0cwUfs&sig=zhODSjonNOgMGiBsVKUcFYdGPXI#v=onepage&q&f=false

Hess C.P. (2012, April 11). Exploring the Brain: Is CT or MRI Better for Brain Imaging? *UCSF Imaging*. Retrieved from: http://blog.radiology.ucsf.edu/neuroradiology/exploring-the-brain-is-ct-or-mri-better-for-brain-imaging/

HORUS (2017, April 28). Horus The Invisible Made Audible. Retrieved from: www.horus.tech/horus/?l=en_us

Huo R., Burden S.K., Hoyt C.S., Good W.V. (1999). Chronic cortical visual impairment in children: aetiology, prognosis, and associated neurological deficits. *Br J Ophthalmol*. 83(6): 670-675.

Iannizzotto G., et al. (2005). Badge3d for Visually Impaired. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshop*. Retrieved from: www.ieeexplore.ieee.org/document/1565326

Isak L. Nilsson, William V. Lindberg (2008). Visual Perception: New Research. *Nova Science Pub Incorporated*.

Jan J.E. and Groenveld M. (1993), Visual behaviours and adaptations associated with cortical and ocular impairment in children. *Journal of Vision Impairment & Blindness*. 87(4): 101-105.

Jennifer L.W. (2009). Voxel-based morphometry: An automated technique for assessing structural changes in the brain. *Neuroscience j*. 29(31): 9661-9664.

Kalapurayil, M. (2013, July 8). What Is MRI? How Does MRI Work? *Medical News Today*. Retrieved from: http://www.medicalnewstoday.com/articles/146309.php.

Kaufman A., Cohen D., Yagel R. (1993). Volume Graphics. *IEEE computer*. 26(7): 51-64

Kurzweil Educational System (2017, April 27). Accessible Text to Speech Support for Individuals with Low Vision Or Blindness. Retrieved from: www.kurzweiledu.com/special/kurzweil-1000-text-access-blind-;ow-vision/

Leibs A. (2015). Computer Resources List for Blind & Visually Impaired.

About.Com Tech. Reviewed on (17 Aug 2015). Retrieved from: www.assistivetechnology.about.com/od/ATCAT1/a/computer-resources-list-for-blind-and-visually-impaired.htm

Linda G.S. and George C.S. (2001). Computer Vision. New Jersey, Prentice-Hall, 279-325. ISBN 0-13-030796-3.

Lusby F.W. (2013, July 2). Visual Field. *midlinePlus*. Retrieved from: http://www.nlm.nih.gov/medlineplus/ency/article/003879.htm

Macintyre-Béon C., Ibrahim H., Hay I., Cockburn D., Calvert J., Dutton G.N., Bowman R. (2010). Dorsal stream dysfunction in children: a review and an approach to diagnosis and management. *Current Pediatric Reviews*. 6(3): 166-182

Math Works (2016, March 18). *Image Classification with Bag of Visual Words*. Retrieved from: http://uk.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html

McRobbie D.W., Moore E.A., Graves M.J. and Prince M.R. (2007). MRI from picture to proton. *Cambridge University Press*. Available from: http://ucrfisicamedica.files.wordpress.com/2010/10/mri.pdf

Nordqvist C. (2014, May 15). What is a CT scan? What is a CAT scan? *Medical News Today*. Retrieved from: http://www.medicalnewstoday.com/articles/153201.php

Owano N. (2013, August 14). OpenGlass apps show support for visually impaired. *PHYSORG*. Reviewed on (18 Aug 2015). Retrieved from: www.phys.org/news/2013-08-openglass-apps-visually-impaired-video.html

Palmer S.E. (1999). Vision Science: Photons to Phenomenology. Cambridge, MA: The MIT Press.

Peter Ganza's Blog (2013, March 20). How to Product Manage a Brain Tumor. Retrieved from: http://pganza.wordpress.com/2013/03/20/how-to-product-manage-a-brain-tumor/

Pinola M. (2011, March 4). The Cambridge Face Memory Test Measures How Good You Are At Remembering Faces. *Lifehacker*. Retrieved from: www.lifehacker.com.au/2011/03/the-cambridge-face-memory-test-measures-how-good-you-are-at-remembering-faces/

Rouse M. (2005, September 1). What is Brialle Display? – Definition from WhatIs.com. Reviewed on (16 Aug 2015). Retrieved from: www.whatis.techtarget.com/definition/braille-display

Rouse M. (2007). Voxel. *TechTarget*. Retrieved from: http://whatis.techtarget.com/definition/voxel

Rouse M. (2013, July 1). Google Glass. *IoT Agenda*. Reviewed on (28 April 2017). Retrieved from: www.internetofthingsagenda.techtarget.com/definintion/Google-Glass

Sanjeev A. (2012, October 3). A tale of two scans: which is better-a CT scan or an MRI? *THE SENTINEL*. Retrieved from: http://www.hanfordsentinel.com/lifestyles/health-med-fit/a-tale-of-two-scans-which-is-better--/article_82dce24a-0d87-11e2-bb76-0019bb2963f4.html

Sargent R.G. (2011). Verification and Validation of Simulation. Proceedings of the 2011 Winter Simulation Conference. Retrieved from https://dl.acm.org/citation.cfm?id=2431538

Schenk-Rootlieb A.J., van Nieuwenhuizen O., van Waes P.F. and van der Graaf Y. (1994). Cerebral visual impairment in cerebral palsy: relation to structural abnormalities of the cerebrum. *Neuropediatrics*. 25(2): 68-72.

Segre L. (2014, May 1). The Eye Chart and 20/20 Vision. *All About Vision*. Retrieved from: http://www.allaboutvision.com/eye-test/

Smith J. (2007). Cortical vision impairment. *Special Education Service Agency (SESA)*. Retrieved from: http://sesa.org/pub/Vision/SESASpring07CVI.pdf

Smith N.D., Glen F.C. and Crabb D.P. (2012). Eye movement during visual search in patients with glaucoma. *BMC Ophthalmol*. 12: 45.

Smith N.D., Glen F.C., Mönter V.M., and Crabb D.P. (2014). Using Eye Tracking to Assess Reading Performance in Patients with Glaucoma: A Within-Person Study. *Journal of Ophthalmology*. 2014: Article ID 120528. doi:10.1155/2014/120528

Souse D.A. (2001). How the special brain learns. Thousand Oaks, CA: Crown Press.

Stevens T. (2013, April 20). Google Glass review. *Engadget*. Reviewed on (18 Aug 2015). Retrieved from: www.engadget.com/2013/04/30/google-glass-review/

Swift S.H., Davidson R.C. and Weens L.J. (2008). Cortical impairment in children: presentation, intervention, and prognosis in educational settings. *TEACHING Exceptional Children Plus*. 4(5), Article 4.

Retrieved from:
http://journals.cec.sped.org/tecplus/vol4/iss5/art4/

Ungerleider L.G., Mishkin M. (1982). Two cortical visual systems. Analysis of Visual Behaviour. The MIT Press: Cambridge, Mass: 549-586.

Vapnik N. (1998). Statistical Learning Theory. New York, John Wiley & Sons, 421-422. ISBN 0-471-03003-1.

Vilvestre J. (2016, November 1). New AI Powered Wearable Can Help the Blind Read and Navigate. *Futurism*. Reviewed on (28 April 2017). Retrieved from: www.futurism.com/new-ai-powered-wearable-can-help-the-blind-read-and-navigate/

Walker H.K., Spector R.H., Hall W.D., Hurst J.W., editors. (1990). Chapter 116. Visual Fields. In: Clinical Methods: The History, Physical, and Laboratory Examinations. (3rd Ed). Boston: Butterworths; Available from:
http://www.ncbi.nlm.nih.gov/books/NBK220/

Watkins K.E., Paus T., Lerch J.P., Zijdenbos A., Collins D.L. Neelin P., Taylor J., Worsley K.J. and Evans A.C. (2001). Structural asymmetries in the human brain: a voxel-based statistical analysis of 142 MRI scans. *Cerebral Cortex.* 11(9): 868-877.

Watson L. (2005). What is a Screen Reader?. *Nomensa*. Reviewed on (17 Aug 2015). Retrieved from:
www.nomensa.com/blog/2005/what-is-a-screen-reader

ZAMOJC I. (2012). Introduction to Unity3D-Tuts+Code Tutorial. *Code Tuts+*. Web 10 Aug 2015. Retrieved from:
www.code.tutsplus.com/tutorials/Introduction-to-unity3d—mobile-10752