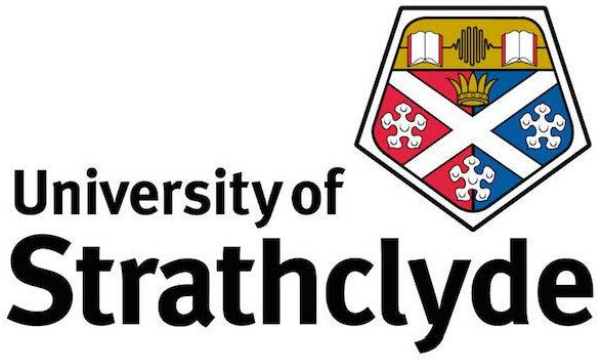


STEREO VISION TECHNOLOGIES FOR RETINAL IMAGING

Submitted in **2021** by **Ian Coghill**
Department of Biomedical Engineering
University of Strathclyde

Thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Engineering in Medical Devices

DECLARATION OF AUTHOR'S RIGHTS

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed: 

Date: May 2021

LIST OF PUBLICATIONS

PEER-REVIEWED CONFERENCE PAPERS

May 2021 **Stereo vision based optic nerve head 3D reconstruction using a slit lamp fitted with cameras: performance trial with an eye phantom** [In Press]
Coghill, I. & Giardini, M. E., *2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC),* Glasgow, UK, 2021

[Contribution: contributed to the design of the experiment; designed, fabricated, and assembled equipment; performed measurements; contributed to data analysis; created and implemented stereo vision algorithms; contributed to manuscript].

Aug 2020 **Low-cost eye phantom for stereophotogrammetry-based optic nerve head topographical 3D imaging**
Coghill, I., Black, R. A., Livingstone, I. A. T. & Giardini, M. E., *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Montreal, QC, Canada, 2020, pp. 1604-1607, DOI: 10.1109/EMBC44109.2020.9175326

[Contribution: contributed to phantom optical design; designed, fabricated and assembled phantom; contributed to the design of the experiment; designed, fabricated and assembled equipment; performed measurements; contributed to data analysis; created and implemented stereo vision algorithms; contributed to manuscript].

Oct 2019 **3D reconstruction of the optic nerve head of a phantom eye from images obtained using a slit lamp fitted with low cost add-ons**
Coghill, I., Jordan, K. C., Black, R. A., Livingstone, I. A. T. & Giardini, M. E., *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Berlin, Germany, 2019, pp. 4717-4720, DOI: 10.1109/EMBC.2019.8856654

[Contribution: contributed to the design of the experiment; designed, fabricated, and assembled equipment; performed measurements; contributed to data analysis; created and implemented stereo vision algorithms; contributed to manuscript].

CONFERENCE SINGLE PAGE ABSTRACTS/POSTERS

Sep 2020 **Phantom and porcine eye optic nerve head 3D reconstruction from stereo images acquired through a slit Lamp fitted with low-cost add-ons**
Coghill, I., Livingstone, I. A. T., Black, R. A. & Giardini, M. E., *Scottish Ophthalmological Club Virtual Autumn Meeting, 2020*

[Contribution: contributed to the design of the experiment; designed, fabricated, and assembled equipment; performed measurements; contributed to data analysis; created and implemented stereo vision algorithms; contributed to manuscript].

Sep 2019 **Phantom eye optic nerve head 3D reconstruction from stereo images acquired using a novel simultaneous stereo fundus imaging technique**
Coghill, I., Livingstone, I. A. T., Black, R. A. & Giardini, M. E., *BioMedEng19 Proceedings: The UK's largest gathering of Biomedical Engineers Medical Engineers and Bioengineers*, London, UK, 2019, pp. 213-213

[Contribution: contributed to the design of the experiment; designed, fabricated, and assembled equipment; performed measurements; contributed to data analysis; created and implemented stereo vision algorithms; contributed to manuscript].

Jul 2019 **Synthetic stereo images of the optic disc from the CORD dataset**

Coghill, I., Jordan, K. C., Menolotto, M., Black, R. A., Livingstone, I. A. T. & Giardini, M. E., *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Berlin, Germany, 2019

[Contribution: contributed to the design of the experiment; designed virtual environment and performed measurements; contributed to data analysis; created and implemented stereo vision algorithms; contributed to manuscript].

Sep 2018 **3D reconstruction of the fundus of a phantom eye through stereo imaging of slit lamp images**

Coghill, I., Jordan, K. C., Black, R. A., Livingstone, I. A. T. & Giardini, M. E., *BioMedEng19 Proceedings: The UK's largest gathering of Biomedical Engineers Medical Engineers and Bioengineers*, London, UK, 2018, pp. 216-216

[Contribution: contributed to the design of the experiment; designed, fabricated, and assembled equipment; performed measurements; contributed to data analysis; created and implemented stereo vision algorithms; contributed to manuscript].

ABSTRACT

Early detection and treatment are crucial in limiting vision loss induced by primary open-angle glaucoma, a highly prevalent subtype of glaucoma, which is the second leading cause of blindness globally. Subjective assessment of the optic nerve head (ONH), which shows morphological change prior to the development of symptoms, is the cornerstone of early detection but suffers from poor intra- and inter-observer variability. Quantitative three-dimensional (3D) imaging technologies, which objectively quantify such change, have excellent repeatability and reproducibility, and have been shown to have higher diagnostic ability than general ophthalmologists. These could boost diagnostic ability in primary/community care if they were available at an affordable cost.

In this work, two novel low-cost quantitative 3D imaging solutions, based on computer stereo vision, for 3D surface reconstruction of the ONH were developed. The first solution featured low-cost cameras and an indirect ophthalmoscopy lens fitted to a slit lamp. Later versions of the system also featured a projector, used to artificially add texture to the ONH to enhance stereo matching performance. Using eye phantoms, the systems were calibrated, and acquired images of ONH targets in the phantoms were processed using developed stereo vision pipelines to yield reconstructions. Such reconstructions were evaluated against ground truth data. Initial testing was also carried out with porcine eyes. Aiming for a miniaturised handheld system, the second solution featured two small cameras in an indirect ophthalmoscopy configuration, with a folded optical path provided by an appropriate set of mirrors, and a miniature projection system. This solution was tested in a simple eye phantom only.

This work demonstrated ONH 3D reconstruction using both systems, with promising levels of precision and accuracy. Further work is needed, particularly with biological ONHs, to gauge their clinical potential. If sufficient, these systems have the potential to broaden access to 3D ONH imaging in future.

ACKNOWLEDGEMENTS

I wish to show my gratitude to my primary supervisor, Dr. Mario Ettore Giardini, for his invaluable guidance and support, and incredible input over the course of this project. Throughout the project, he continually made sure I was progressing in the right direction and made himself available whenever help was needed. I am also grateful for the continued support and guidance from Dr. Richard Black during this project. Moreover, I would like to thank Dr. Iain Livingstone for allowing me to visit an Ophthalmology Department, answering my questions regarding clinical matters, and for his input and enthusiasm in this project. Furthermore, I would also like to thank Professor Dr. Ir. Heidi Ottevaere for allowing me to visit her photonics research group in Brussels and the lessons learnt during the visit.

I would also like to express my thanks to Stephen Murray for his technical help and for the 3D printing and laser cutting of parts. Moreover, I would like to thank Catherine Henderson for her immense help in preparing for the experiment with pig's eyes. Furthermore, I would like to thank Dr. Alaster McDonach at the Advanced Forming Research Centre for 3D scanning the optic nerve head models of one of the eye phantoms.

I would also like to acknowledge the support of the UK EPSRC Centre for Doctoral Training (CDT) in Medical Devices and Health Technologies, University of Strathclyde, for the studentship awarded to me. Moreover, I would like to thank Carol McInnes and Professor Patricia Connolly from the CDT for their continued support throughout the program.

I would also like to thank the lab members that I have worked with over the course of my research. It was great to work with you all.

Moreover, I wish to thank my family, friends, and girlfriend Elise, for their encouragement and support during my time as a doctoral student.

Lastly, I am forever grateful to my father, without whom I would not be in the position I am in today.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Acronyms and Abbreviations

3D	Three-Dimensional
CAD	Computer-Aided Design
CNC	Computer Numerical Control
CSLO	Confocal Scanning Laser Ophthalmoscopy
DMD	Digital Micromirror Device
FDM	Fused Deposition Modelling
GPIO	General-Purpose Input-Output
HIPS	High Impact Polystyrene
HRT	Heidelberg Retinal Tomograph
ILM	Internal Limiting Membrane
IOL	Intraocular Lens
IOP	Intraocular Pressure
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
NCC	Normalised Cross Correlation
OCT	Optical Coherence Tomography
ONH	Optic Nerve Head
PCB	Printed Circuit Board
PLA	Polylactic Acid
PMMA	Polymethyl Methacrylate
POAG	Primary Open-Angle Glaucoma
PRISMA-ScR	Preferred Reporting Items for Systematic Reviews and Meta-Analyses – Scoping Reviews
RANSAC	Random Sample Consensus
RGC	Retinal Ganglion Cell
RMS	Root-Mean-Square
RNFL	Retinal Nerve Fibre Layer
SD	Standard Deviation
SLP	Scanning Laser Polarimetry
SURF	Speeded Up Robust Features

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

The second leading cause of blindness globally is glaucoma (Kingman, 2004) – a group of neuropathies that result in irreversible vision loss (Quigley, 2011). The most prevalent form, primary open-angle glaucoma (POAG), was estimated to affect almost 53 million individuals in 2020 (Tham et al., 2014). One of the major reasons why people still go blind from POAG is that many of the people who have the condition are unaware that they have it, leading many to not receive treatment until significant vision loss has occurred (Susanna et al., 2015). In the United States, studies indicate that more than 50% of people with glaucoma are undiagnosed, while in South Africa this figure approaches 87% (Susanna et al., 2015). The reasons for this are thought to be that some do not attend or have access to regular eye check-ups, and that some cases are missed by clinicians (Susanna et al., 2015).

POAG is a slowly progressing condition. Since the visual loss associated with it is irreversible, and treatment is only able to halt or slow its progression (Noecker, 2006), early detection is paramount to preserving vision. Change to a dimple-like structure at the back of the eye, known as the optic nerve head (ONH; shown in Figure 1), the point where the optic nerve meets the eye, clearly visible when imaging the back of the eye, has been shown to precede detectable vision loss (Leung, 2015). As a result, change to the structure of the ONH is a key clinical indicator for POAG. In clinics, the ONH is most commonly evaluated for structural change in a subjective manner using stereoscopic images of the ONH (Giaconi et al., 2016b) taken with a fundus camera, or using a slit lamp (an eye examination instrument which, among many other functions, allows for a magnified and stereoscopic view of the ONH). Subjective assessment of the ONH for signs of glaucomatous damage is not straightforward,

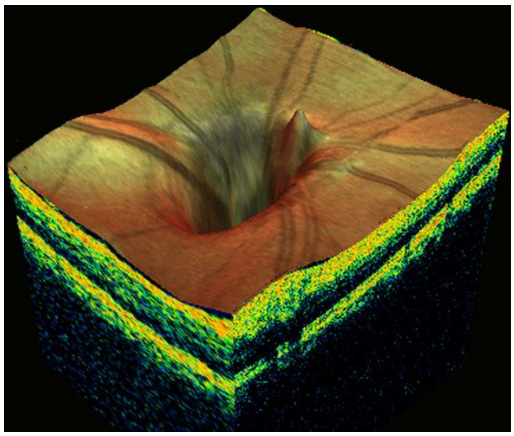


Figure 1: 3D image of a human optic nerve head taken using an OCT system. (Haag-Streit Group, n.d.).

and inter- and intra-observer agreement is often poor, even for specialists (Reus et al., 2007). Quantitative devices, which can take quantitative measurements of the ONH's structure and provide an objective assessment, with excellent reproducibility and repeatability (Ara et al., 2015; Callan et al., 2012; Prata et al., 2010; Rohrschneider et al., 1994), have been shown to have greater diagnostic performance than the subjective assessment by general ophthalmologists (Vessani et al., 2009). The most common quantitative devices used in clinics are optical coherence tomography (OCT) machines, the Heidelberg Retinal Tomograph (HRT; Heidelberg Engineering GmbH, Heidelberg, Germany) and the scanning laser polarimetry (SLP) system known as the GDx (Carl Zeiss Meditec AG, Jena, Germany).

Since the quantitative imaging devices mentioned have been shown to perform better, in terms of diagnostic accuracy, than subjective assessment by general ophthalmologists (who have a higher level of training than the primary eye care health professionals who play the vital role of detecting cases and making referrals to hospital services [Weinreb et al., 2014]), they could have a significant impact in reducing the proportion of cases which are missed. However these devices are not widely available due to their prohibitive cost (Dabasia et al., 2014). Not including maintenance costs, OCT machines can cost up to $150,000, while the HRT and GDx are approximately $40,000 (Kwartz et al., 2005). In the United Kingdom, more and more optometry practices are adopting OCT machines (Kern et al., 2020), however in other countries, particularly in low- to middle-income countries, which also have fewer specialists and are the worst affected (Damji et al., 2017), the availability of quantitative imaging devices is almost certainly lower. Therefore, there is a possibility that the creation of quantitative imaging technologies, at a cost permitting wide adoption, could have a significant impact.

One method which could allow for quantitative imaging at low cost is the application of computer stereo vision techniques for reconstruction of the ONH from stereo images. This method has been explored in the works of a number of researchers (Bansal et al., 2013; Corona, Mitra, Wilson, Krile, et al., 2002; Nakagawa et al., 2008; Tang et al., 2011; Juan Xu & Chutatape, 2006), however it seems as though in every case, a fundus camera was used to acquire the stereo images. Fundus cameras are also expensive, costing thousands of pounds (Fenner et al., 2018). It must be noted here that the low-cost monoscopic fundus cameras that have been proposed (Tran et al., 2012) are not suitable for this purpose since quantitative stereo of non-stationary targets, such as the human eye in vivo, requires

simultaneous capture of both images of the stereo pair (Bradski & Kaehler, 2008). A potentially promising idea, which to the author's knowledge has not been explored before, is to apply computer stereo vision techniques to images obtained using a slit lamp, which is already available in practically all ophthalmology and optometry practices, fitted with low-cost cameras. Moreover, a low-cost handheld stereo fundus camera could be advantageous where slit lamps are not available, and for rural areas where it is difficult to transport bulky equipment.

1.2 Research Aims and Objectives

The core aim of this work is to develop an instrument, consisting of a slit lamp fitted with low-cost components, together with computer stereo vision algorithms, and test its feasibility in performing quantitative three-dimensional (3D) surface reconstruction of the ONH. A secondary aim is to do the same for a low-cost handheld stereo fundus camera – develop the instrument and test its feasibility.

The objectives of this work are to:

1 Develop a simple initial slit lamp-based instrument and simple eye phantom, containing highly textured 3D targets on its retina, and perform uncalibrated reconstruction of the targets by applying standard stereo vision algorithms to stereo images acquired using the instrument.

2 Perform calibrated reconstruction of highly textured ONH targets in the simple eye phantom by applying custom stereo vision algorithms to stereo images acquired using an upgraded version of the instrument.

3 Perform calibrated reconstruction of practically textureless ONH targets in the simple eye phantom by applying custom stereo vision algorithms to stereo images acquired using the instrument, which has been upgraded to enable it to project textured patterns onto the ONH targets, essentially artificially adding texture to allow the algorithms to operate.

4 Develop a complex eye phantom, which closely matches the human eye in terms of optical characteristics, and contains ONH targets with little texture, and perform

calibrated reconstruction of the targets in the manner, and with the instrument, described in objective 3.

5 Perform calibrated reconstruction of the ONH of porcine eyes in the manner, and with the instrument, described in objective 3.

6 Develop an initial handheld stereo fundus camera instrument, which also features a projection system, and perform calibrated reconstruction of practically textureless ONH targets in the simple eye phantom by applying custom stereo vision algorithms to stereo images acquired using the instrument.

1.3 Thesis Outline

Following this introductory chapter, **Chapter 2** provides some background on glaucoma, specifically its epidemiology, pathophysiology, treatment, and diagnosis. The chapter also introduces the theory and practice of computer stereo vision. **Chapter 3** presents a scoping review which was conducted to identify the various works which have been done on the application of computer stereo vision techniques for 3D surface reconstruction of the ONH. Then, given that one of the objectives was to create an eye phantom, which closely matches the human eye in terms of optical characteristics, **Chapter 4** presents a second scoping review which was conducted to identify the various complex phantoms (containing a cornea, lens and fluid-filled interior) which have been detailed in the literature.

In the 5 chapters which follow, the work carried out on the slit lamp system is presented. **Chapter 5** presents the first design iteration of the system and a demonstration of uncalibrated 3D reconstruction of highly textured ONH targets in a simple eye phantom using basic stereo vision algorithms. **Chapter 6** presents an upgraded system and a demonstration of metrically calibrated 3D reconstruction of highly textured ONH targets in a simple eye phantom using tailored stereo vision algorithms. **Chapter 7** presents a demonstration of metrically calibrated reconstruction of ONH targets with a low texture density (more closely matching the texture density of the human ONH) in a simple eye phantom. In this chapter the system is also further upgraded, including the addition of a pattern projection system to artificially add texture to the surfaces of the ONH targets, to provide sufficient texture for the stereo vision algorithms to operate. In **Chapter 8**, a complex fluid-filled eye phantom, which more closely matches the human eye in terms of optical characteristics than the simple

eye phantoms used in the work of the previous chapters, is presented. A demonstration of calibrated reconstruction of practically textureless ONH targets within it is also presented. In **Chapter 9**, initial testing of the latest version of the slit lamp system on porcine eyes is presented.

Chapter 10 details initial work on the development and testing of the low-cost handheld stereo fundus camera system. A demonstration of metrically calibrated reconstruction of practically textureless ONH targets within a simple eye phantom is presented.

Conclusions on the work of this thesis, and recommendations for future work (in the short- and long-term), are given in **Chapter 11**.

CHAPTER 2

GLAUCOMA AND COMPUTER STEREO VISION

2.1 The Human Eye

2.1.1 Introduction

This section explores aspects of the human eye which are relevant to the work of this thesis. First the anatomy of the human eye is explored in order to provide a basis for the following section on glaucoma. Given the importance of the optical characteristics of the human eye in this work, and the fact that eye phantoms were developed which aimed to mimic it, the next subsection briefly explores the optical characteristics of the human eye.

2.1.2 Anatomy

The human eye is configured in a similar way to a camera. Cameras feature lenses which focus light, incoming from the environment, onto an image sensor. The image sensor converts the light incident on it into electrical signals, which can be stored digitally. Similarly, the human eye contains optical elements that focus light onto the retina – a layer at the back, and inside, of the eye which converts the light into neural signals. After initial pre-processing on the retina itself, these neural signals are delivered to the brain, via nerve fibres, for higher-level processing.

Figure 2 illustrates the structure of the human eye. The main supporting structure, which is responsible for the white of the eye, is known as the sclera (the thick layer shown in bluish white in the figure). Attached to the sclera, and at the front of the eye, is a thin transparent structure known as the cornea. It is a type of positive meniscus lens, and the interface between the cornea and air accounts for the majority of the optical power of the eye (around two thirds of its total optical power). The remainder of the eye's optical power is provided by the crystalline lens (or "lens" for short). The lens serves the important function of allowing the eye to adjust its focus, allowing for focusing on objects at different distances. In order to allow for this, the power of the lens is variable. Its power is modified by the action of ciliary muscles, which contract and relax to change its shape. The iris of the eye, a colourful ring-shaped structure, responsible for the colour of the eye, is located just in front of the lens and acts as the eye's diaphragm. Its opening, known as the pupil, is adjusted depending on the intensity of light entering the eye, to control the amount of light reaching the retina, to match it to the retina's dynamic range and to protect it from damage by excessive light. The space between the iris and cornea is known as the anterior chamber and is filled with a fluid known as aqueous humour. The space between the lens and iris is filled with the same fluid and is

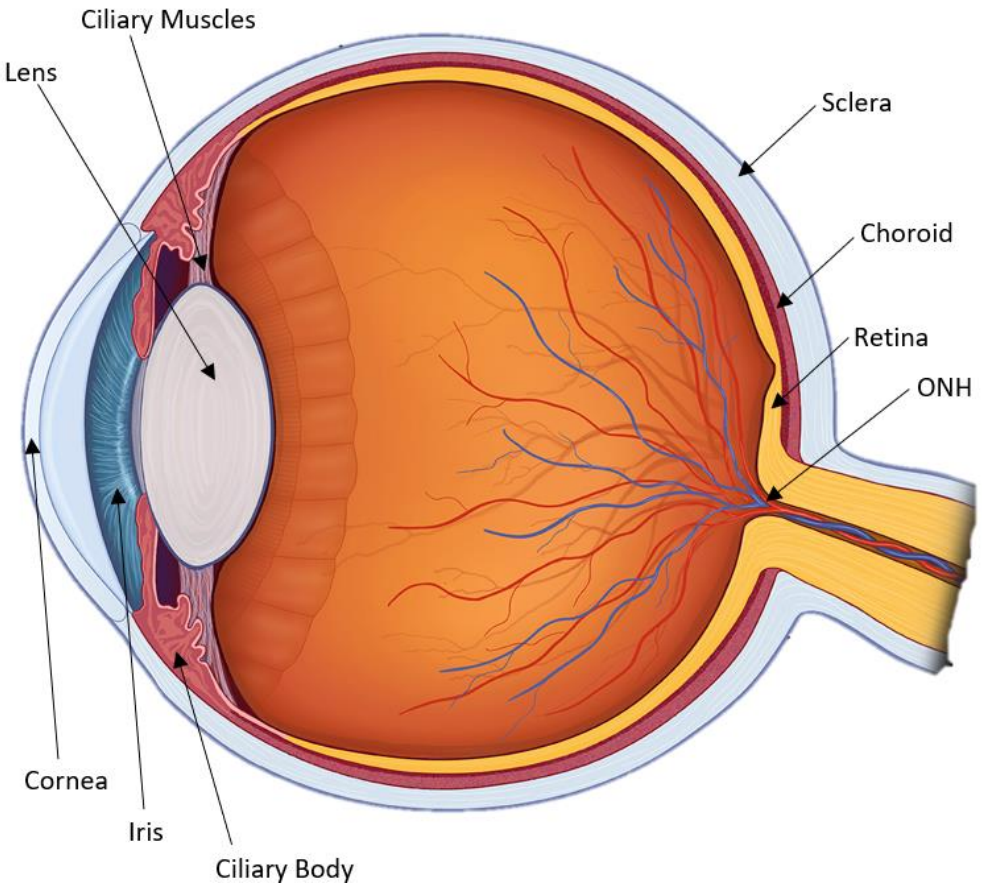


Figure 2: Diagram illustrating the main structure of the human eye. (Modified from Marriner, n.d.).

called the posterior chamber. The space behind the lens, known as the vitreous body, is filled with a fluid known as vitreous humour, which is thick and gel-like, contrasting with the thin watery aqueous humour.

Inside the eye, the sclera is lined with two layers: the retina and choroid. The choroid is a highly vascular layer that provides oxygen and nutrients to the outer layers of the retina. The retina is the light sensitive layer, consisting of millions of photoreceptors called cone and rod cells, that convert light into neural signals. The retina is composed of many different cell types, and layers. Figure 3 illustrates the different cell types present within the retina and their location in relation to its different layers. A labelled OCT image, which shows a cross-sectional view of the different layers, is provided in Figure 4 to complement the illustration in Figure 3. The deeper layers contain the photoreceptors. Information from the photoreceptors is passed through the cells of the various layers to the ganglion cells, which are present within the ganglion cell layer. The ganglion cells are the neurons which then pass the information to the brain. The axons, which make up the retinal nerve fibre layer (RNFL),

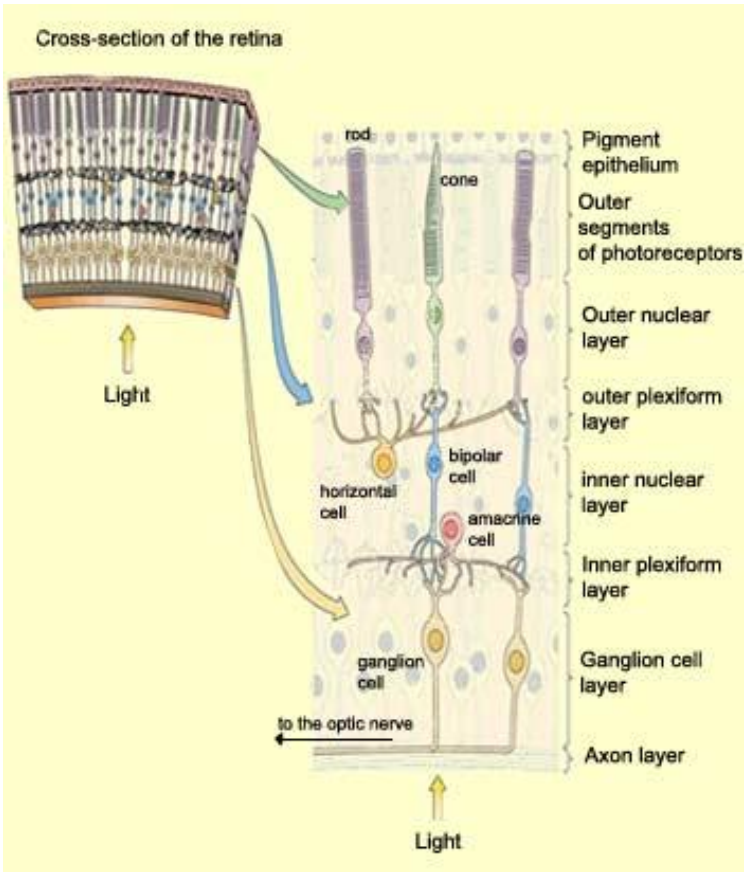


Figure 3: Diagram illustrating the different cells and layers within the human retina. (Paquet, 2020).

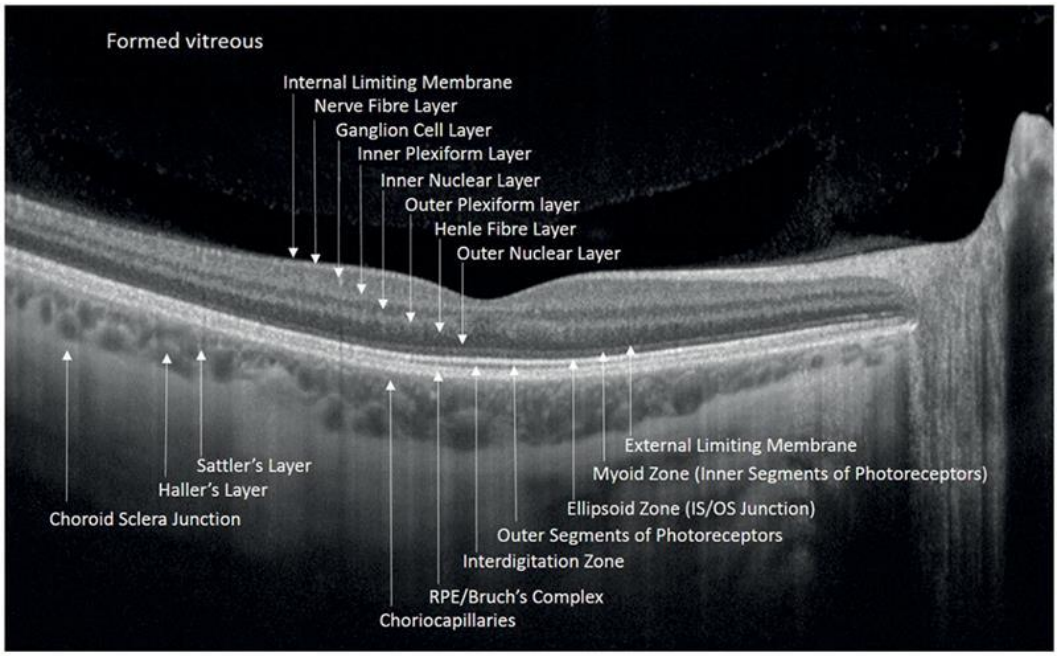


Figure 4: OCT image showing the cross section of the different layers within the human retina, choroid, and the sclera. (Elbassiouny et al., 2018).

termed axon layer in Figure 3, of these cells run towards the single point of exit at the back of the eye, known as the ONH.

Figure 5 illustrates the ONH structure. A manually segmented OCT image is provided in Figure 6 to complement this illustration. At the ONH, the axons of the ganglion cells exit the eye through a mesh-like structure within a hole in the sclera, known as the lamina cribrosa, and form the optic nerve. At the ONH, the choroid and deeper layers of the retina terminate, and the axons turn in towards the lamina cribrosa, resulting in the characteristic dimple-like shape of the ONH. At the ONH, the choroidal layer, where it terminates, is separated from the axons by fibrous tissue, known as the Border Tissue of Elschnig, which extends from the sclera to the innermost layer of the choroid (Bruch's membrane). The ONH can be directly visualised by looking into the back of the eye using a retinal examination instrument such as an ophthalmoscope, slit lamp or fundus camera. Stereo images taken of an ONH using a fundus camera, which can be viewed stereoscopically to perceive its 3D structure, are shown in Figure 7. The blood vessels which can be seen emanating from it serve the purpose of oxygenating and nourishing the outer layers of the retina. The centre of the ONH is typically located 15° nasally and 1.5° upwards, in relation to a structure known as the fovea (Atchison & Smith, 2000). The fovea is located typically 5° off of the best fit optical axis of the eye in

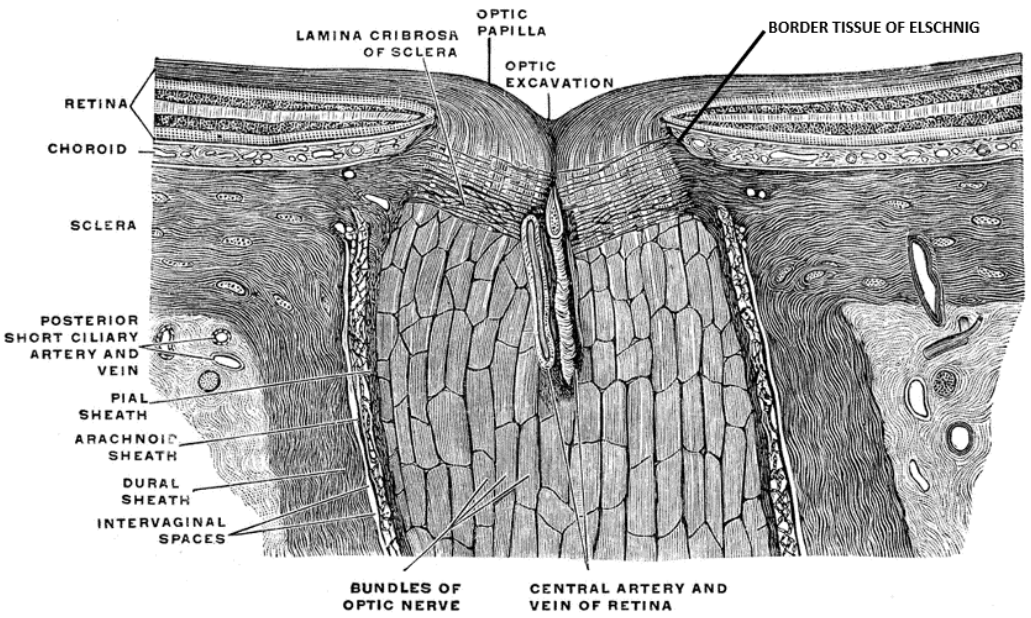


Figure 5: Diagram illustrating the structure of the optic nerve head. (Gray's Anatomy. Descriptive and Applied., 1913).

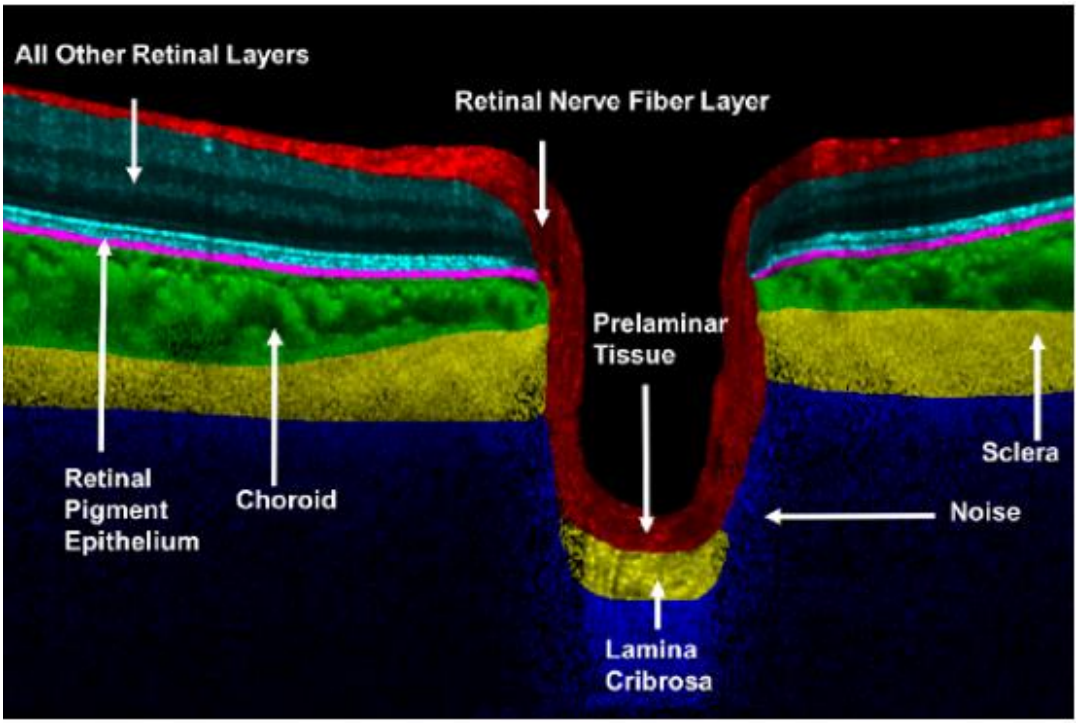


Figure 6: Manually segmented OCT image showing the different structures within the ONH. (Devalla et al., 2017).

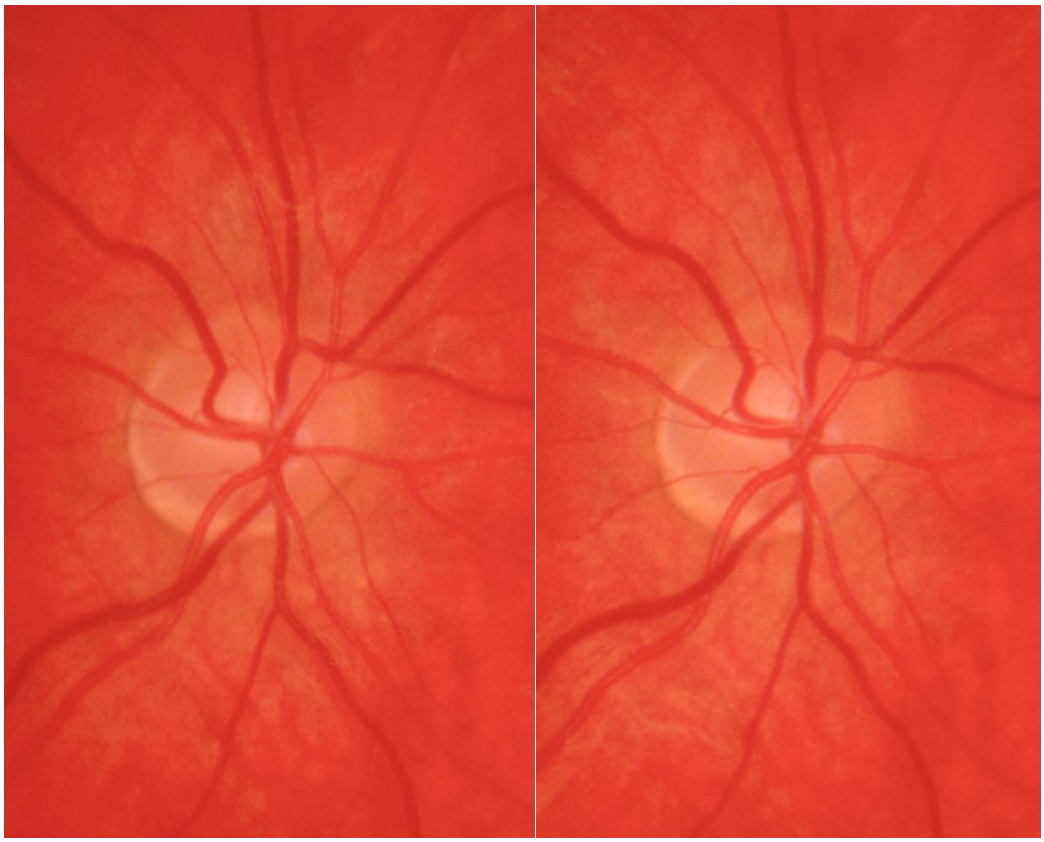


Figure 7: Stereo images of a human optic nerve head. (Menolotto et al., 2019).

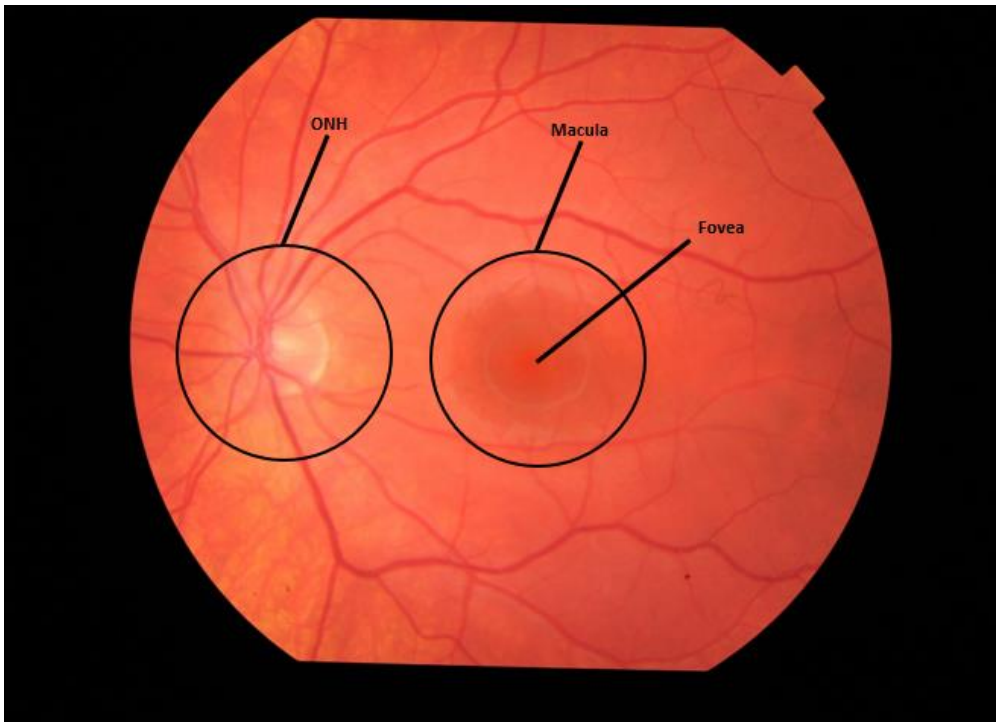


Figure 8: Fundus image showing the ONH and macula. (Menolotto et al., 2019).

the temporal direction, meaning the ONH is approximately 10° off axis in the human eye. A fundus image of the retina, showing the position of the ONH relative to the fovea, is provided in Figure 8. The fovea is also a small dimple-like structure and is located at the centre of a region known as the macula. The macula is responsible for the high-resolution of the human eye's central vision. The fovea results in the highest resolution, compared with all other areas of the retina.

2.1.3 Optical Characteristics

As was mentioned above, the optical elements of the eye focus light onto the retina, where the crystalline lens, the active optical component, changes in shape to allow for focusing on objects at different distances from the eye. Given that ophthalmic instruments operate with the eye in its relaxed state (i.e., focused to view objects far in the distance), only the eye in its relaxed state is discussed here. When the light is able to be focused on the retina in the relaxed state, this is known as emmetropia. The left image of Figure 9 illustrates how the rays of light focus onto the retina in an emmetropic eye. In the population, many individuals have refractive errors. In the relaxed state, where the light is focused in front of the retina, as is illustrated in the right image of Figure 9, the condition is known as myopia. Where the light is focused behind the retina, this is termed hypermetropia. In order to simplify matters in

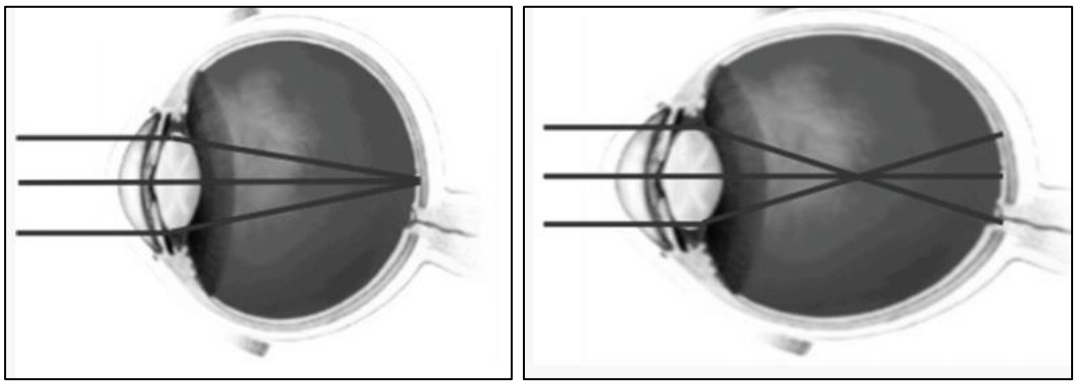


*Figure 9: Images showing how light from a distant object is focused for an emmetropic (**LEFT**) and myopic eye (**RIGHT**). (Artal, 2016).*

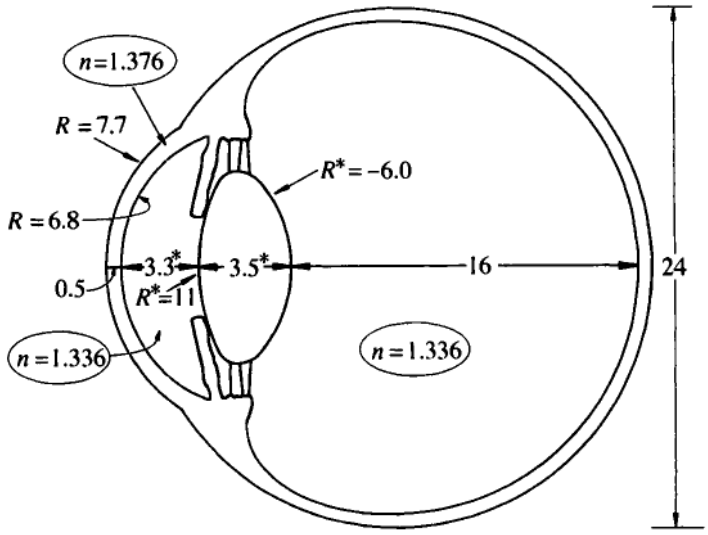


Figure 10: Typical dimensions and refractive indices of the different elements of the human eye in a relaxed state. Dimensions are in units of millimetres, and the terms with an asterisk are those which vary with accommodation. (Atchison & Smith, 2000).

this work, which is at a very early research stage, refractive errors are not considered, and only emmetropic eyes in a relaxed state.

Typical dimensions and refractive indices of the different components within a human eye are shown in Figure 10. The optical elements function together to produce an overall focusing power of around 60 dioptres. It can be seen in the figure that the refractive index of the crystalline lens is not shown. This is due to the fact that it has a complicated distribution of refractive index. An illustration of the refractive index distribution within the crystalline lens, along with its layered nature, is provided in Figure 11. If the index distribution is ignored, the uniform index equivalent value of the lens is roughly 1.42 (Atchison & Smith, 2000).

It is important to mention that, in contrast with a camera, the sensor of the eye (i.e., the retina) is curved. It is curved in order to match the field curvature produced by the optics of the eye, allowing for focus to be maintained off-axis. Given that they do not feature strongly in this work, a thorough discussion on the other optical aberrations which are present in the eye is not presented here.

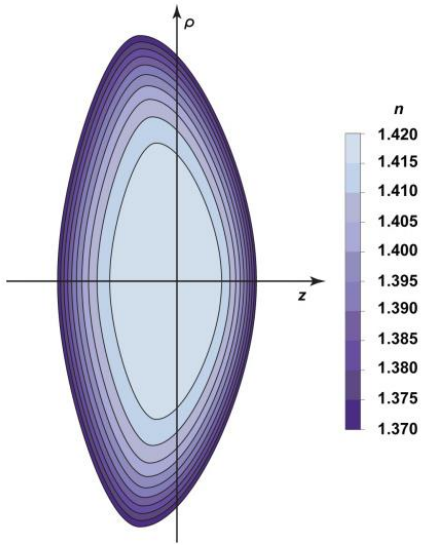


Figure 11: Model of crystalline lens showing its layered nature and refractive index distribution. (Bahrami et al., 2016).

2.2 GLAUCOMA

2.2.1 INTRODUCTION

This section on glaucoma presents aspects of glaucoma which are relevant to the work of this thesis. In order to highlight the scale of the problem, the section begins with a subsection on epidemiology. Following this, the pathophysiology of the different types of glaucoma is explored. Then, after a brief look at the treatment options available for the condition, aspects pertaining to its diagnosis are detailed, with a particular focus on the techniques used to view or image the ONH.

2.2.2 EPIDEMIOLOGY

In 2020, the second leading cause of blindness worldwide was glaucoma, at 3.6 million cases (Adelson et al., 2021). It has been reported that of those blinded by glaucoma, 38% suffer

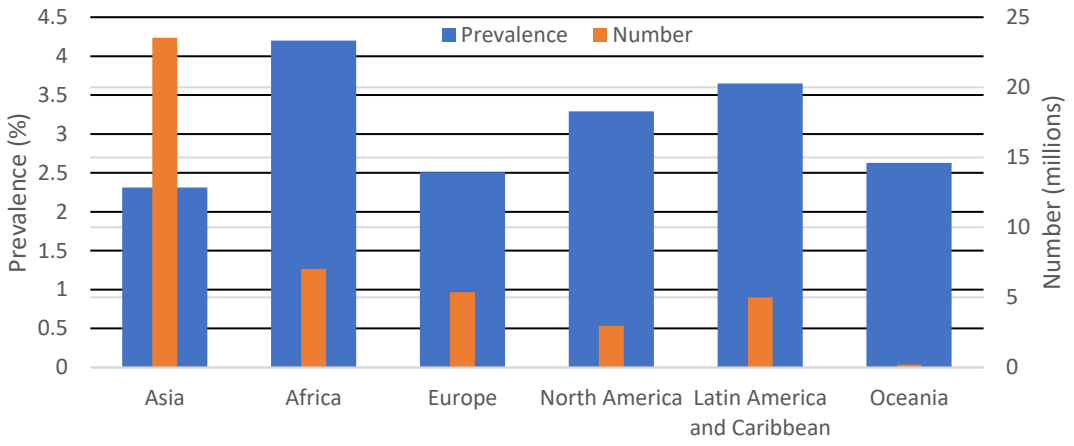


Figure 12: Column chart showing the prevalence and number of individuals, between the ages 40 and 60, with POAG in the different regions of the world. (Data from Tham et al., 2014).

from POAG (Parihar, 2016). People living in low-income countries are more likely to progress to blindness than those in high-income countries (Delgado et al., 2019), and it is estimated that the proportion that progress to blindness could be twice as high (Chen, 2004).

It was estimated in 2013 that, the number of individuals with glaucoma globally, between the ages of 40 and 80 years old, was 64.3 million (corresponding to a global prevalence of 3.54%; Tham et al., 2014). It was estimated that this value would have been 76 million during the year 2020 and will rise to 111.8 million by 2040. Of the 76 million with glaucoma in 2020, it was estimated that 52.68 million would have had POAG (corresponding to 69.3% of cases). POAG, due to its chronic progressive nature, is more prevalent in older populations. In fact, per decade of age increase, the odds ratio for POAG prevalence has been estimated to be approximately 1.73 (Tham et al., 2014). Moreover, it has been found that men are more likely than women to have POAG, with an odds ratio of around 1.36 (Tham et al., 2014). The number of people, and the corresponding prevalence, in the different regions of the world with POAG in 2013, between the ages of 40 and 80, are presented in the column chart in Figure 12. It can be seen that Africa had the highest prevalence and Asia the highest number of people affected. In terms of ethnicity, the prevalence of POAG in people of African descent has been shown to be higher than those of different ethnic origins (Tham et al., 2014).

2.2.3 Pathophysiology

Glaucoma is an umbrella term that encompasses several neuropathic conditions which lead to degradation of the retinal ganglion cells (RGCs). It is known that the pressure level within the eye, termed the intraocular pressure, is linked with the death of these cells (Weinreb et al., 2014). The intraocular pressure of the eye is a function of the secretion flowrate of

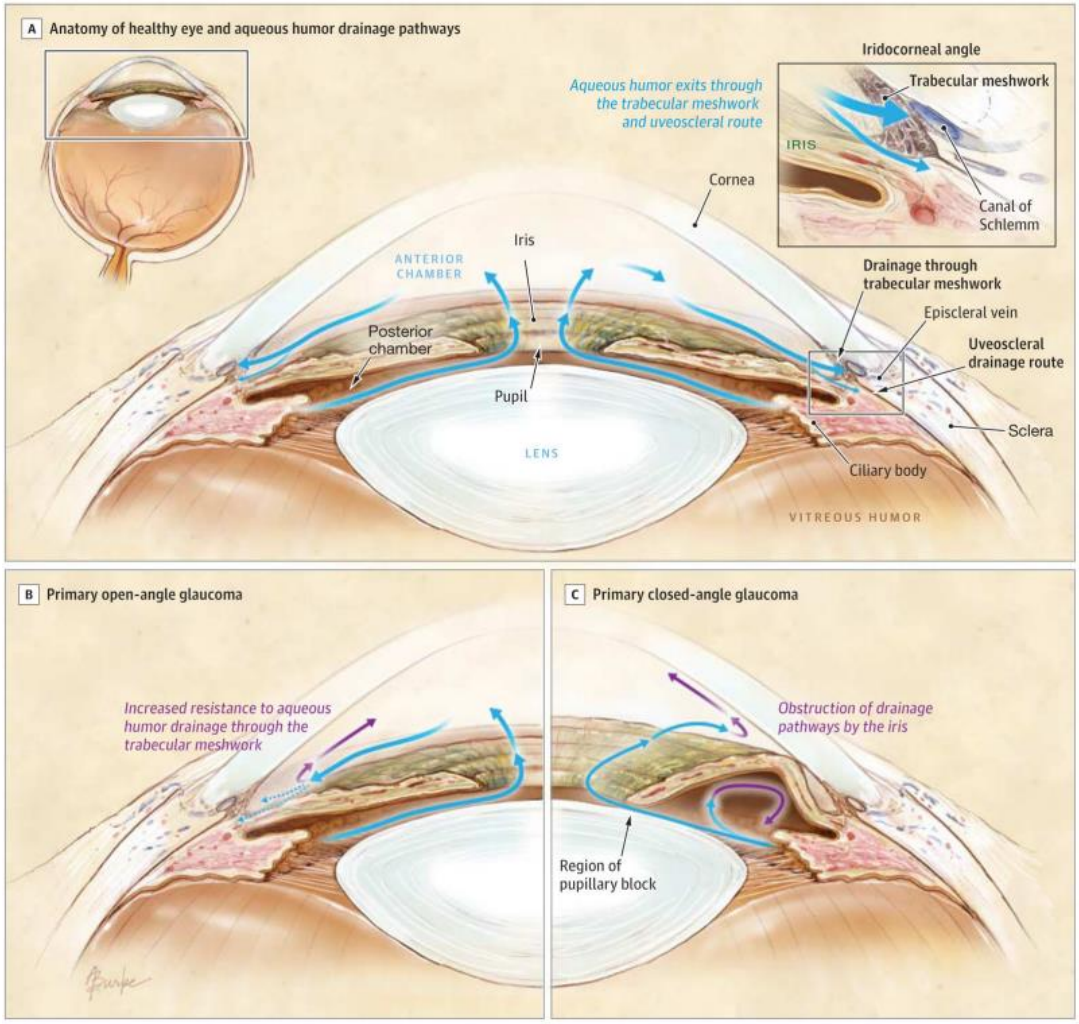


Figure 13: Mechanisms responsible for primary open and closed angle glaucoma. (Weinreb et al., 2014).

aqueous humour from the epithelial cells of the ciliary body, and the excretion flowrate of it from the eye through the trabecular meshwork and uveoscleral outflow pathway. The path that aqueous humour takes from the ciliary body to the two drainage pathways in a healthy and correctly functioning eye is illustrated in the top cell of Figure 13. It can be seen that the fluid passes through the opening in the iris (i.e., the pupil) in order to travel from the posterior chamber to the anterior chamber.

There are two main categories of glaucoma. These are closed-angle and open-angle glaucoma. In closed-angle glaucoma, intraocular pressure is heightened due to an obstruction of the two drainage pathways by the iris. A blockage (pupillary block) preventing aqueous humour from traveling from the posterior chamber to the anterior chamber results in a build-up of pressure in the posterior chamber. This build-up of pressure causes the iris to bulge towards the cornea, causing the angle between the iris and cornea (iridocorneal

angle) to decrease. Closure of this angle leads to the blocking effect. This mechanism is illustrated in the bottom right cell of Figure 13. This category of glaucoma is diagnosed by observation of the iridocorneal angle.

This work is focused on the other category of glaucoma: open-angle glaucoma. In open-angle glaucoma, the drainage pathways of aqueous humour are not obstructed but instead the trabecular meshwork shows increased resistance to flow of the fluid through it, as illustrated in the bottom left cell of Figure 13, leading to increased pressure. This form is chronic and progresses slowly. Increased pressure on the lamina cribrosa is theorised to result in RGC death due to ischemia and mechanical damage (Albon et al., 2000). The progressive death of these cells causes progressive and irreversible visual loss. Such visual loss typically goes unnoticed until it is severe enough that central vision begins to be affected (Jonas et al., 2017). The death of the RGCs causes the retinal nerve fibre layer to atrophy. Since all of the axons meet at the ONH, degradation is particularly noticeable there and causes the ONH to look more and more excavated over time. In order to give an indication of what this change can look like, OCT images of a normal ONH and one with quite significant glaucomatous damage are provided in Figure 14. Open-angle glaucoma can be further sub-divided into primary or secondary open-angle glaucoma. The difference between POAG and secondary open-angle glaucoma is that with secondary open angle glaucoma the root cause is a separate underlying eye condition (such as inflammation, trauma, etc.).

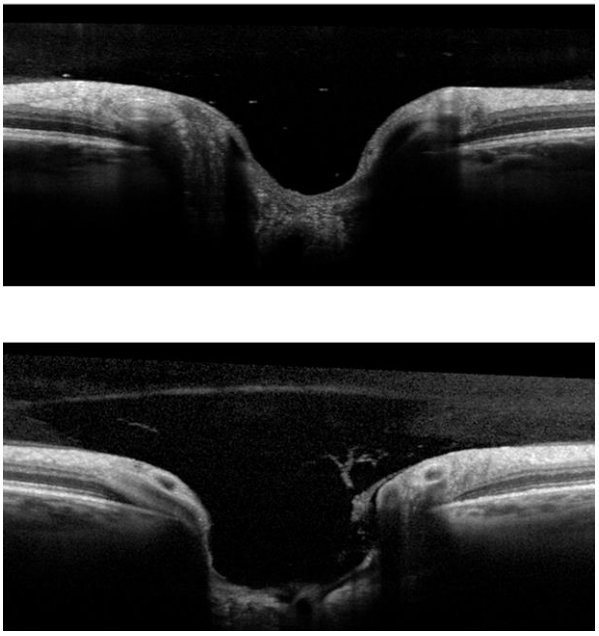


Figure 14: OCT images of a normal ONH (TOP) and an ONH showing significant glaucomatous damage (BOTTOM). (Heidelberg Engineering, n.d.).

2.2.4 Treatment

POAG is an irreversible condition, and treatment is only able to halt or slow further progression of the disease. The only treatments that have been proven effective in doing so, are those which lower the intraocular pressure (IOP; Yoo et al., 2018). Available treatments include surgical interventions and those which act pharmacologically. The medication-based treatments, which are administered either systemically or topically, either increase the output flowrate of aqueous humour or decrease its rate of production (Killer & Pircher, 2020). The majority of the surgical procedures lead to changes that increase the output flow rate of aqueous humour and include procedures such as laser trabeculoplasty (Gazzard et al., 2003), incisional trabeculotomy or the insertion of a glaucoma drainage device (Wang & Barton, 2017).

2.2.5 Diagnosis

Since visual loss caused by glaucoma is irreversible, but its progression can be halted or slowed with treatment, early detection is vital to preventing or mitigating visual loss. Early detection, for a number of reasons, is not straightforward. Despite the fact that high IOP, which is typically measured clinically using a procedure known as tonometry (Bader & Havens, 2020), is thought to cause neural damage, many individuals who have glaucoma actually have an IOP that is within the normal range (Song and Caprioli, 2014; considered to be roughly 10 to 21 mmHg [Khaw et al., 2004]). Some studies have shown this to be the case for 30 to 40% of all individuals showing a glaucoma induced defect in their vision (Song & Caprioli, 2014). Moreover, some individuals with IOP levels considered to be high do not develop glaucoma (Jiang et al., 2018). It is not entirely clear why this is the case however it is thought that the pressure gradient across the lamina cribrosa, determined by the balance of IOP and cerebrospinal fluid pressure, which would determine the deformation of the lamina cribrosa, could be more telling (Abe et al., 2015). IOP can fluctuate throughout the day and corneal thickness, which can be different between individuals, can affect IOP measurement accuracy (Susanna et al., 2015). Thus, IOP on its own is not a reliable diagnostic measure for the detection of POAG but is still measured clinically since it is an easily measurable risk factor (Matlach et al., 2018).

Another factor that makes POAG difficult to detect early is that it is typically asymptomatic until it has reached an advanced stage. Consequently, at the time of diagnosis, many patients, who do not see an eye specialist until symptoms develop, already have significant visual

defects (Susanna et al., 2015). Furthermore, signs of visual deterioration, assed clinically by a specialist, most commonly through a visual field evaluation (Alencar & Medeiros, 2011), are usually only detectable after a significant number (as much as 30 to 50%) of RGCs have died (Harwerth et al., 1999). Since a significant amount of neural damage can occur before visual defects are even detectable, evaluation of the ONH, where such damage is most apparent, resulting in a change in its 3D structure, is a vital indicator in early detection.

Although practice can vary depending on the location and setting, typically an eye test will involve testing of the IOP, visual field and ONH. Where there is suspicion that the patient may have POAG, they are then referred to a hospital eye service where an extensive examination is performed by an expert, again involving testing of the same three aspects, and a decision is made by considering all of the information in conjunction. In the follow-up of patients diagnosed with the condition, they are again subject to testing of the same three aspects. IOP is monitored to gauge the response to IOP lowering medication, and visual function and ONH structure are followed over time to look for any deterioration. These three pieces of information are again considered in conjunction to guide the treatment plan of the patient.

The ONH is typically examined clinically using either direct ophthalmoscopy, indirect ophthalmoscopy or a slit lamp biomicroscope, or briefly “slit lamp” (Giaconi et al., 2016b). Examination using the slit lamp, which provides a magnified and stereoscopic view, is usually the preferred means to assess the ONH (Giaconi et al., 2016a). Slit lamps are mainstays within eye care practices, allowing for visualisation of a number of different structures within the eye (useful in the diagnosis of many conditions), and as such are available in the majority of eye care practices. An image of a slit lamp is provided in Figure 15. The instrument contains an illumination system, biomicroscope and patient chin rest. The illumination system is used to illuminate the structures of the eye to be visualised, and the biomicroscope to view those structures. The biomicroscope contains two independent optical paths, for the left and right eye of the examiner, to allow for the depth of those structures to be perceived. It also contains a dial which can be used to control the magnification of the view. In order to view the retina with the biomicroscope, the principles of indirect ophthalmoscopy are exploited. Indirect ophthalmoscopy, as is illustrated in Figure 16, makes use of an indirect ophthalmoscopy lens, briefly “indirect lens”, to form an aerial image of the retina outside the eye, which is observed, in this case using a biomicroscope. Commonly used indirect lenses for ONH evaluation are those with a power of 78 or 90 dioptres (Shaarawy et al., 2014).

Figure 15: Image of a slit lamp. (Optics Incorporated, n.d.).

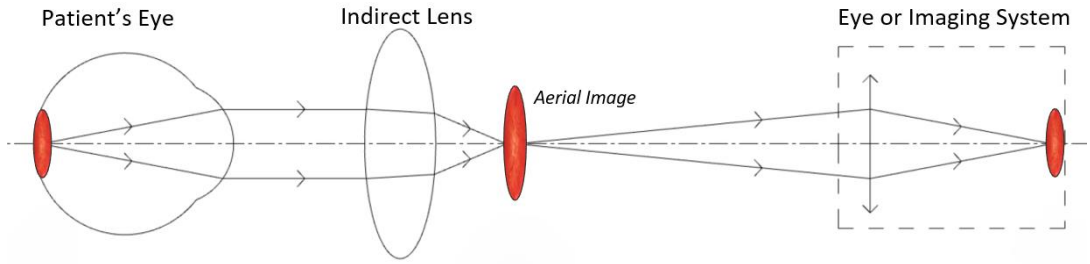


Figure 16: Diagram to illustrate indirect ophthalmoscopy.

Dilation of the pupil, which can be induced by dilating eye drops, is usually necessary to obtain a stereoscopic view (Giaconi et al., 2016a).

The evaluation of the ONH involves looking for signs of damage such as haemorrhages but also taking measurements of its morphology, since damage leads to changes to its structure. Regarding its morphology, one important parameter that clinicians rely on, which can indicate that glaucoma may be present, is the cup-to-disc ratio. This parameter tries to quantify the amount of neural tissue (axons of the RGCs) exiting the eye. It is defined as the ratio between the size of the cup (the inside margin of the neural tissue) and the size of the disc (the outer margin of the neural tissue). A simplified illustration of the cross-section of an ONH, showing the cup and disc dimensions, is provided at the top of Figure 17. Below this illustration is an image of the ONH with markings overlaid to roughly indicate the cup and disc margins. The parameter is often determined in the vertical or horizontal directions, but

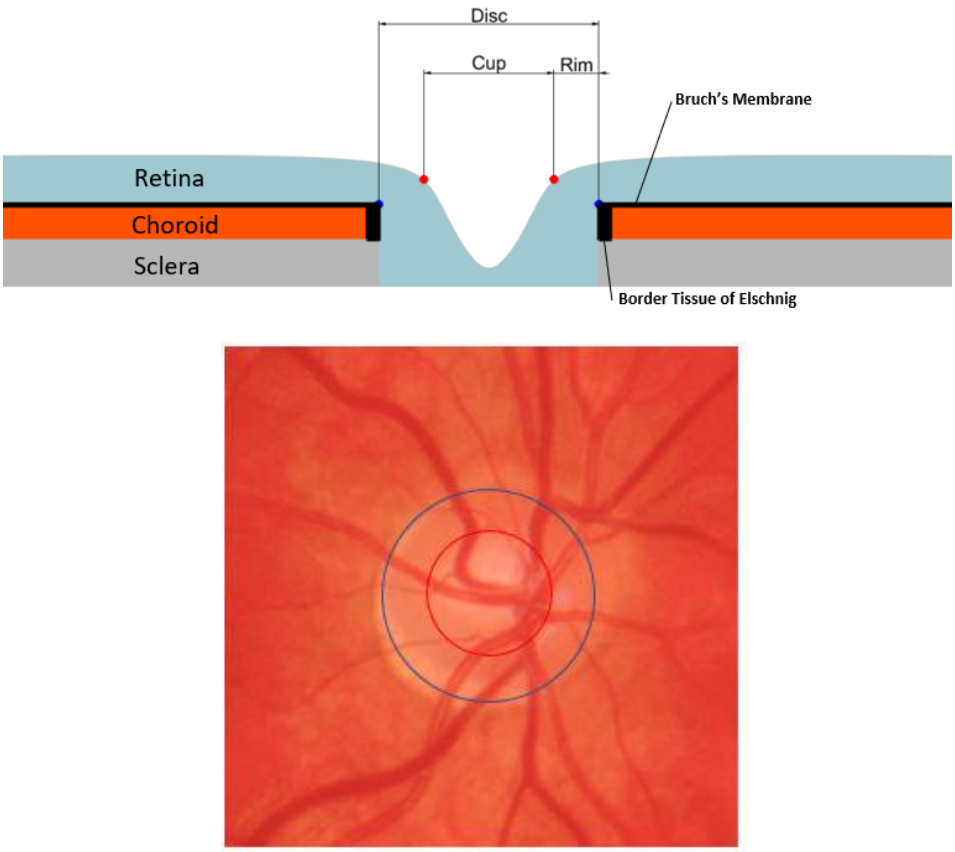


Figure 17: Illustration of the cup, disc, and rim of the optic nerve head. ***TOP****: Cross-sectional illustration of the ONH showing the cup, disc, and rim dimensions.* ***BOTTOM****: ONH image (Menolotto et al., 2019) with approximate cup and disc margins overlaid.*

sometimes the narrowest rim-to-disc ratio (as is recommended in the Scottish glaucoma referral guidelines: SIGN 144 [Scottish Intercollegiate Guidelines Network (SIGN), 2015]) is used or even the area ratio. Since there is a wide variation of disc size within the population (Quigley et al., 1990), and the ratio is tightly related to optic disc size (Tatham et al., 2013), it is often interpreted in conjunction with the absolute disc size (Scottish Intercollegiate Guidelines Network (SIGN), 2015). Regarding the typical disc sizes seen, the average values for the vertical and horizontal disc diameters are 1.88 mm and 1.77 mm, respectively (Arora et al., 2015). In order to give an idea on the range of values possible, they can be categorised into discs which are large (> 2.0 mm), medium (1.5 mm to 2.0 mm) and small (< 1.5 mm; Scottish Intercollegiate Guidelines Network (SIGN), 2015). It must be noted that different studies or guidelines may use slightly different bands for each of the categories. In order to also give an idea of the typical depth dimensions of the ONH, one study reported a mean value of 0.619 mm for the distance between the lowest point of the ONHs' cups to their flatter upper surfaces.

While ONH measurements can be useful clinically (Garway-Heath et al., 1998), they cannot be totally relied upon for a number of reasons. These include high intra- and inter-observer variability in determining such measurements (Tatham et al., 2013), that the clinically visible disc rarely corresponds to the true outer border of the neural tissue (Reis et al., 2012) and the orientation of neural tissue is not taken into account (Reis et al., 2012).

Since there is wide variability in disc appearance, evaluation of the ONH over time is hugely valuable. Cameras that can capture high resolution images of the retina, called fundus cameras, are commonly used where they are available to document the ONH appearance so that it can be assessed over time. Monoscopic and stereoscopic fundus cameras exist, and although it is not entirely clear if stereo photography brings any additional benefit, at least when used by glaucoma experts (Chan et al., 2014), expert interpretation of stereo fundus images is often regarded as the gold standard for POAG diagnosis (Christopher et al., 2018). Despite this, the detection of glaucoma through subjective evaluation of the ONH, even by experts, is not always accurate (Giaconi et al., 2016a). Moreover, the ability to detect changes over time, as assessed in serial fundus images, varies relatively widely among clinicians (Giaconi et al., 2016a).

Technologies that can instead obtain quantitative information on the structure of the ONH or RNFL, quantitative parameters that relate to the amount of neural tissue present, have been introduced relatively recently and are being increasingly adopted. The three most commonly used devices use the techniques of OCT, confocal scanning laser ophthalmoscopy (CSLO) and SLP. OCT, which has been said to have revolutionised ophthalmic practice (Adhi & Duker, 2013), is akin to ultrasound but rather than using sound it uses light. The use of light permits a much higher imaging resolution than is possible with ultrasound. One of the top-of-the-range OCT machines, the Zeiss Cirrus 6000 (Carl Zeiss Meditec AG, Jena, Germany), is capable of imaging with an axial resolution of 5 μm and transverse of 15 μm.

As is the case with ultrasound, the echo time delay is used to determine the distance of the part of tissue that the light (or sound in the case of ultrasound) reflected from, to the device. Since light travels much faster than sound, the echo time delay is much smaller (a resolution on the scale of tens of microns requires a time resolution of approximately 30 femtoseconds) and cannot be measured purely electronically. The principle of interferometry, which is illustrated in Figure 18, is therefore used instead to determine echo delay times (Fujimoto et al., 2000). In this, the light from the source is split into two beams: a reference beam and a

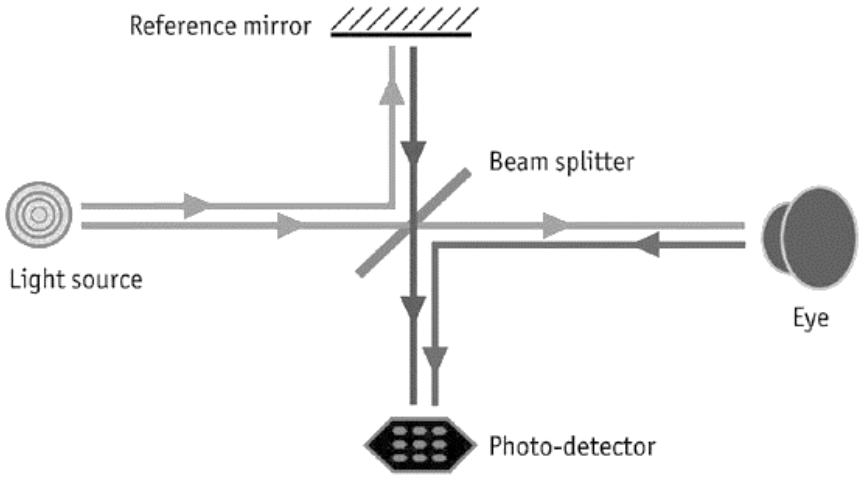


Figure 18: Illustration of the principle of interferometry. (Thomas & Duguid, 2004).

sample beam. The reference beam travels to, and is reflected from, a reference mirror, while the sample beam travels to the structures of the eye and is reflected back. The two beams are then recombined on their return. If the waveforms of both beams return in phase, then constructive interference is produced, and is detected by the photodetector. By varying the distance of the reference mirror over the scan depth, a range of reflecting structures over that scan depth can be picked up, producing an axial scan. It must be noted that this is the principle used in time-domain OCT systems. Newer systems, capable of faster scan speeds, operate in the spectral-domain and are able to scan in a similar way but without the need to mechanically move the reference mirror. OCT machines perform many of these axial scans (A-scans) along a transverse direction to yield cross sectional images of the tissue being scanned. Typically, the scan is performed over an area, so the device actually acquires multiple cross sections over this area, allowing for a full 3D picture to be obtained. The images obtained show the structures within a certain depth of the tissue scanned and not just the surface topography. Cross sectional images obtained using OCT were provided earlier in the chapter, in Figure 4 and Figure 14. As well as the cross-sectional slice images, the devices can also output a map of the RNFL thickness, and parameters which define the structure of the ONH, such as rim area, disc area, cup-to-disc ratio and cup volume. A normative database is used for comparison of key parameters to provide a classification of either within or outside of the normal ranges.

CSLO is the technique used in the commercially available HRT. Illustrations have been provided in Figure 19 to complement the description of its operation that follows. In the HRT, CSLO is used to image the ONH at a series of different planes. CSLO, as is illustrated in the

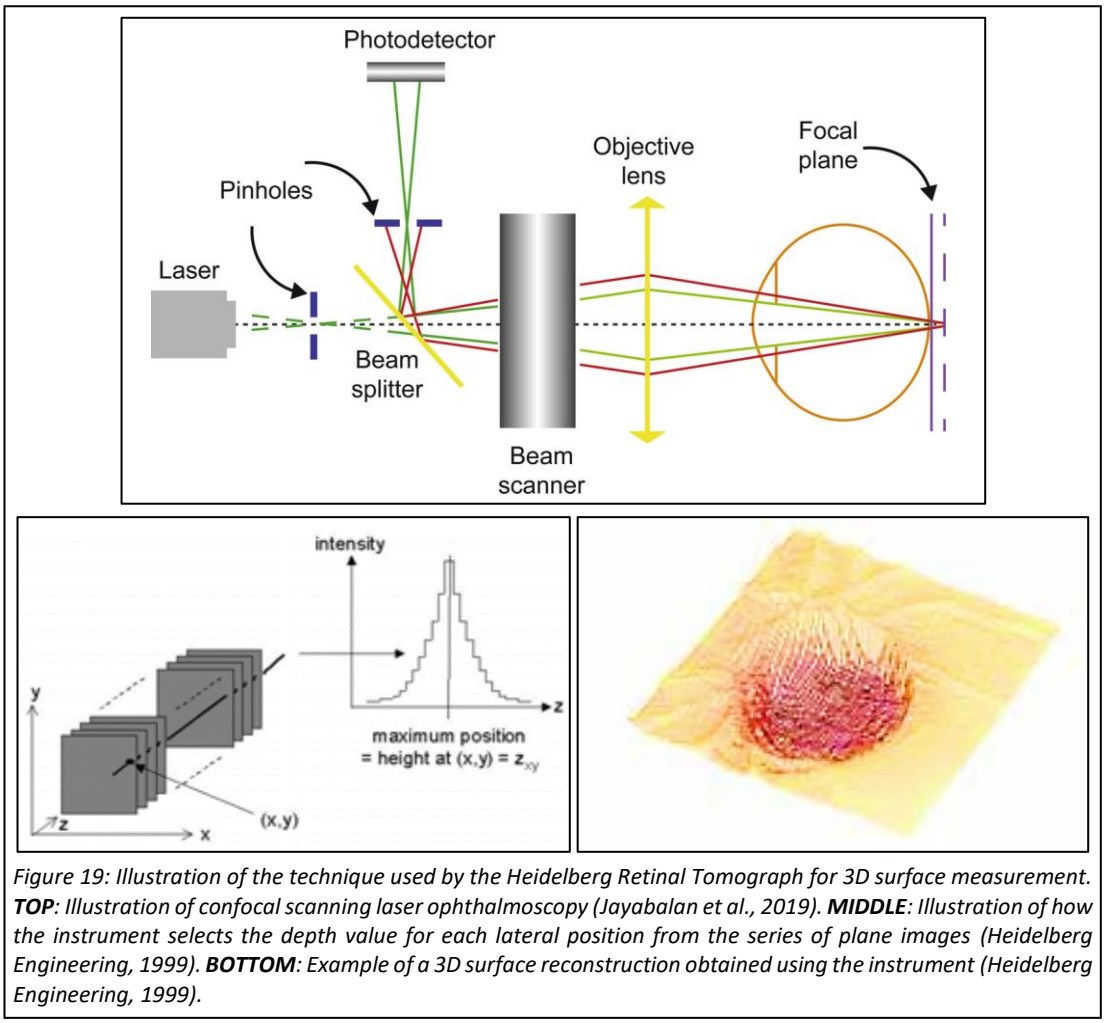


Figure 19: Illustration of the technique used by the Heidelberg Retinal Tomograph for 3D surface measurement. ***TOP****: Illustration of confocal scanning laser ophthalmoscopy (Jayabalan et al., 2019).* ***MIDDLE****: Illustration of how the instrument selects the depth value for each lateral position from the series of plane images (Heidelberg Engineering, 1999).* ***BOTTOM****: Example of a 3D surface reconstruction obtained using the instrument (Heidelberg Engineering, 1999).*

top image of Figure 19, is a technique whereby a laser is used to illuminate a spot on the sample, in this case the ONH, and the light which is reflected at the focal plane passes through a pinhole to a detector. The same pinhole prevents out of focus reflected light, not from the plane of focus, from reaching the detector. A grid of 384 x 384 of these spots is imaged in a raster fashion, at a single focal plane, to form each plane image. The HRT collects 64 of such images. The depth values at each location, as is illustrated in the bottom left image of Figure 19, is taken as the depth of the plane from which the intensity is the highest for that location. The full 3D surface image is obtained by determining the depth value for all locations. An example of such a 3D surface image is shown in the bottom right image of Figure 19. Aside from producing a topographical map of the ONH, the HRT also calculates parameters which describe its structure (similar to those reported by OCT devices – disc area, cup area, cup volume, cup-to-disc ratio, etc). The latest version of the HRT, the HRT III, reports a glaucoma probability score based on a normative database to indicate whether an ONH is within normal limits or not (Maslin et al., 2015).

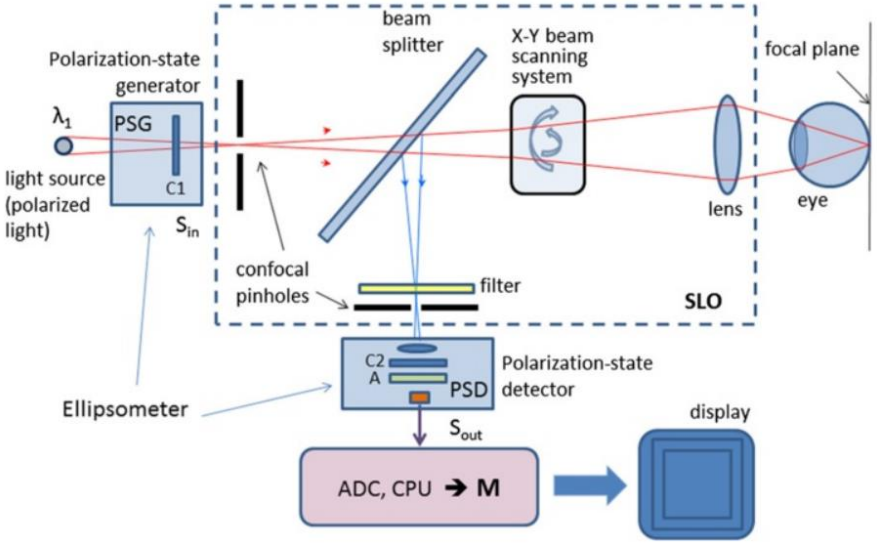


Figure 20: Illustration of a scanning laser polarimeter. (Gramatikov, 2014).

SLP does not provide 3D images or surface topography images of the ONH structure but instead is able to obtain a thickness map of the RNFL. SLP is used in the commercially available device known as the GDx (Carl Zeiss Meditec AG, Jena, Germany). SLP relies on the fact that the RNFL is birefringent (its refractive index is polarisation dependent). The polarisation dependency causes the vertical and horizontal components of the incident light to travel at different speeds within the RNFL, resulting in a phase shift, known as retardation. The thickness, on which the amount of retardation depends, can be determined based on the recorded retardation. The GDx shares similarities with the HRT in terms of its configuration, having also a confocal scanning laser ophthalmoscope. The key difference is that the GDx also features a polarisation-state generator and detector, as can be seen in the simplified diagram of an SLP system in Figure 20, enabling it to measure the amount of retardation. The device generates an RNFL thickness map of the ONH with a lateral resolution of 256 x 256 pixels (Lin et al., 2007). The device also reports the probability of abnormality based on comparison with a normative database.

2.3 Theory and Practice of Computer Stereo Vision

2.3.1 Introduction

This section explores the theoretical and practical aspects of computer stereo vision which are relevant to the work of this thesis. Computer stereo vision is a technique whereby, by exploiting parallax (the differing displacement of objects, at different distances apart, as seen

by a camera when translating horizontally), a 3D reconstruction of a scene can be estimated based on two images taken of that scene at different horizontal locations. Capturing such images, known as stereo image pairs, is commonly done using a stereo camera rig – where two, normally identical, cameras are setup in a configuration where they are displaced horizontally to each other. The rigs can be set up in either a parallel (non-convergent) or convergent configuration, as are illustrated in Figure 21. The processes of performing quantitative 3D reconstructions with these setups are similar but the equations used to calculate scene coordinates based on the obtained images differs. In this section, the geometry of the configuration used within the work of this thesis, the parallel configuration, is explored first. The subsections which follow detail the steps involved in performing reconstruction using this configuration. The chapter ends with a brief look at the geometry of the convergent configuration.

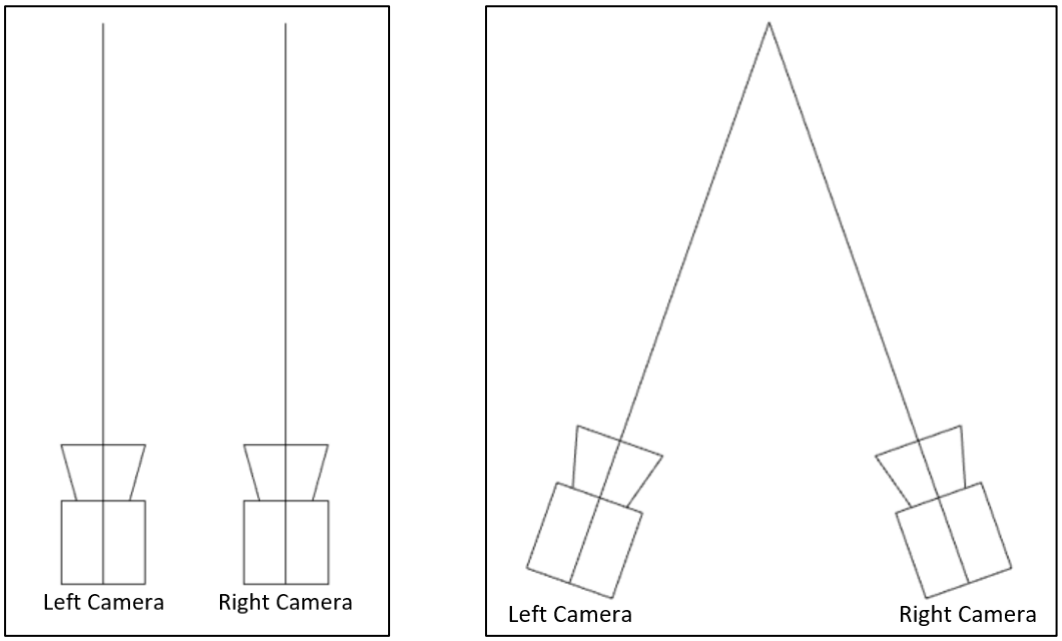


*Figure 21: Typical stereo camera configurations. **LEFT**: Parallel stereo configuration. **RIGHT**: Convergent stereo configuration.*

2.3.2 Parallel Stereo Camera Geometry

The process of extracting three-dimensional information from stereo image pairs, taken using a parallel stereo camera rig, can be simplified by mathematically mapping the physical system to an ideal pinhole stereo camera model. This model will, from now on, be referred to as the ideal stereo model. A diagram to illustrate the ideal stereo model is shown in Figure 22. The model is composed of two identical pinhole camera models that are perfectly aligned,

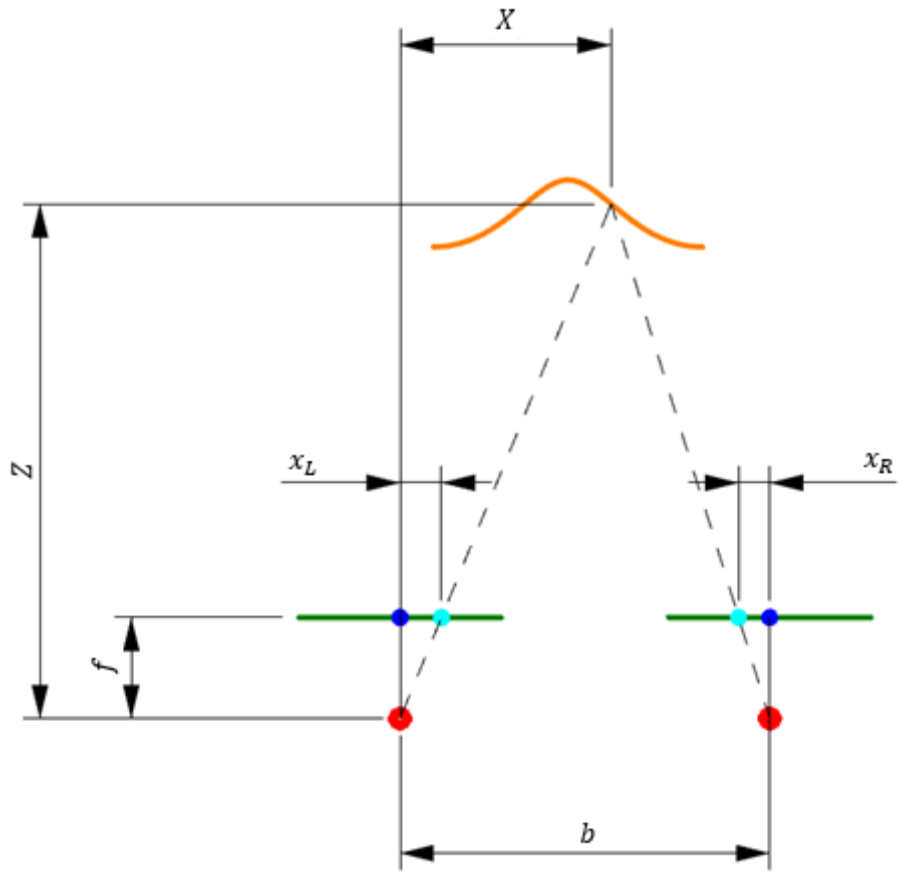


Figure 22: Schematic of the ideal stereo model. ***Orange*** *– Object in the scene,* ***Red*** *– Projection centres,* ***Blue*** *– Principal points,* ***Green*** *– Image sensors,* ***Cyan*** *– Sensor intersection points of rays emanating from a point on the object in the scene.*

separated by a distance, termed the baseline distance, and whose optical axes are parallel. In the pinhole camera model, and in physical pinhole cameras, light travels in a straight line from points on objects in the scene, through the camera's pinhole, to the image sensor. In this configuration, the image of the scene cast on the sensor is inverted horizontally and vertically. In order to correct for such inversions, cameras process the image seen on the sensor to reinvert it. In the pinhole camera model, the position of the sensor and pinhole are often flipped for convenience. In this flipped representation, the pinhole is instead called the projection centre, and light will travel from points on objects in the scene to the projection centre, intersecting the image sensor on the way. The effect of this is that the image cast on the sensor in the new position will be the same as the image obtained in the old position after correcting for the inversions. The focal length of a pinhole camera is the distance from the pinhole to the image sensor, or in this representation, the distance from the image sensor to the projection centre.

Once a system is mapped to the ideal stereo model, the 3D coordinates of points (X, Y, Z) in the scene can be estimated, from the obtained stereo images, using the following equations (derived from similar triangles):

$$Z = \frac{fb}{x_L - x_R} \qquad \textit{Eq. 2.1}$$

$$X = \frac{x_L Z}{f} \qquad \textit{Eq. 2.2}$$

$$Y = \frac{y_L Z}{f} \qquad \textit{Eq. 2.3}$$

where f is the focal length of the cameras and b is the baseline distance. For a particular point in the scene, the term x_L refers to the horizontal distance between the principal point of the left camera (the point on the sensor where the optical axis intersects it) and the location where the ray emanating from that particular point intersects the left sensor. The term x_R refers to the same but on the image sensor of the right camera. The difference in position of that particular point on the left and right image sensors $(x_L - x_R)$ but in units of pixels is known as the disparity and is an important quantity that is discussed more in subsection 2.3.5. The terms y_L and y_R refer to the same but for the vertical distance rather than horizontal. The terms y_L and y_R are not shown in Figure 22 since the y-axis is perpendicular to the page. Locating scene points identified in one image, in the other image, is known as the stereo correspondence problem, and is explored in subsection 2.3.5.

2.3.3 Distortion Coefficients and Intrinsic Parameters

The first step in mapping a physical camera to a pinhole camera model is to remove the distortion created by the non-ideal behaviour of the camera. Physical cameras use lenses instead of pinholes since they allow for a greater amount of light to reach the image sensor. While this allows for faster image capture, the use of lenses causes distortion. The two main types of distortion that are seen with cameras are radial and tangential distortion. Radial distortion is caused when the projection of points in the scene to the sensor deviate from the rectilinear projection seen in the pinhole camera model, where the rays from points in the scene follow a straight path to the image sensor through the pinhole. This deviation is illustrated in the bottom left cell of Figure 23. It can be seen in this illustration that the result is a displacement of the points to a different location on the image sensor than would the

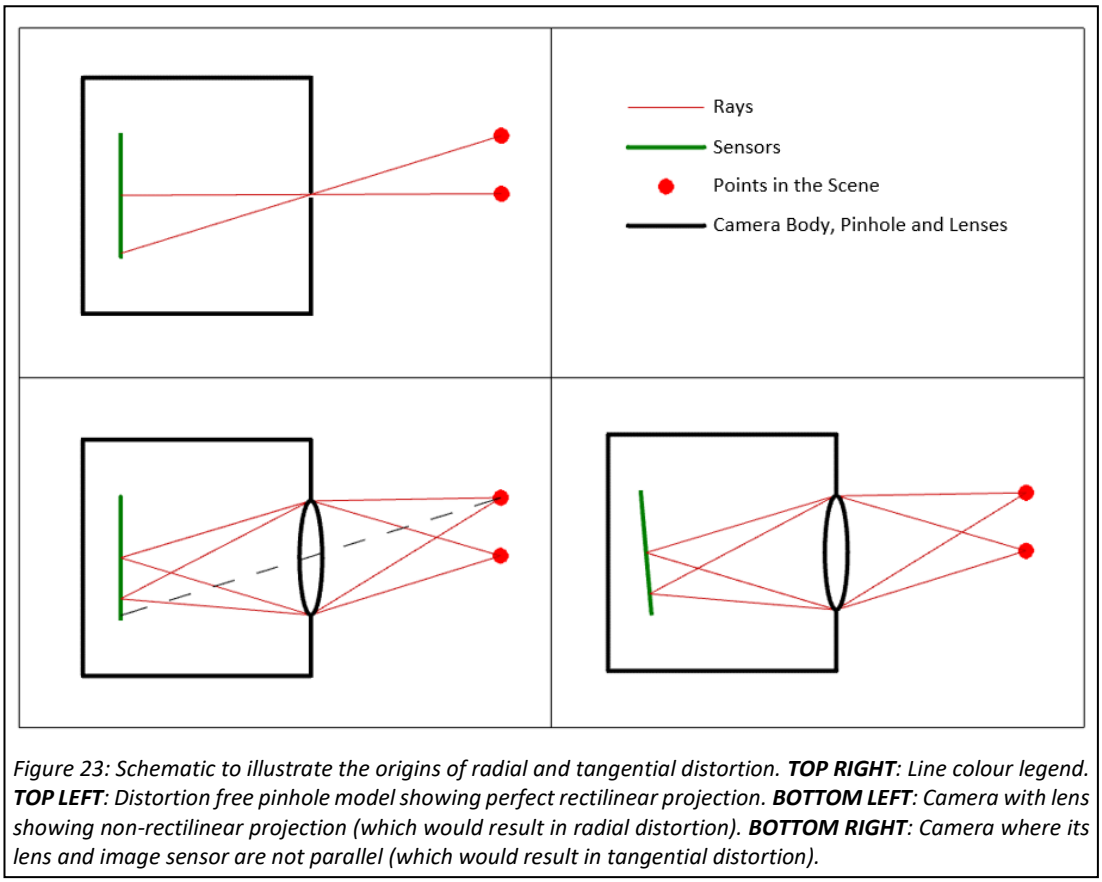


Figure 23: Schematic to illustrate the origins of radial and tangential distortion. ***TOP RIGHT****: Line colour legend.* ***TOP LEFT****: Distortion free pinhole model showing perfect rectilinear projection.* ***BOTTOM LEFT****: Camera with lens showing non-rectilinear projection (which would result in radial distortion).* ***BOTTOM RIGHT****: Camera where its lens and image sensor are not parallel (which would result in tangential distortion).*

case with the pinhole model, as illustrated in the top left cell of Figure 23. The amount of displacement increases with increasing distance from the principal point. Moreover, the direction of displacement can be in either of two directions, resulting in the two types of radial distortion: barrel and pincushion. The effects of these are illustrated in Figure 24, showing their effects when imaging a grid pattern.

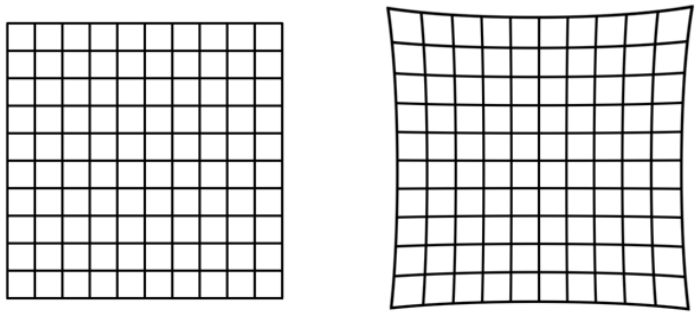
Radial distortion can be corrected for using the following equations:

$$x_{new} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \qquad \textit{Eq. 2.4}$$

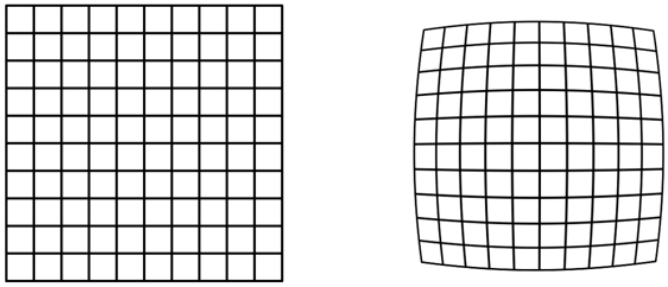
$$y_{new} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \qquad \textit{Eq. 2.5}$$

where r is the distance of a particular point (x, y) from the principal point, and x_{new} and y_{new} are the x and y pixel coordinates of that point after distortion removal, respectively. The k_i terms are the radial distortion coefficients.

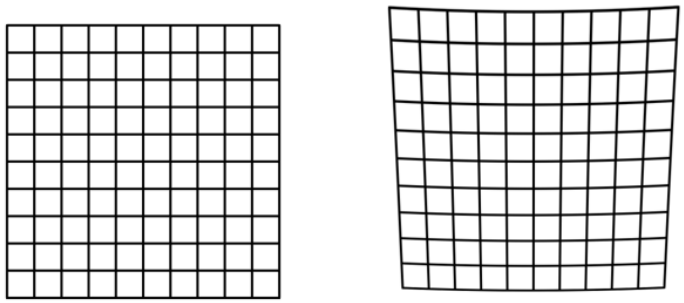
Another main type of distortion that is seen in camera systems is that of tangential distortion. Tangential distortion occurs as a result of imperfect assembly where the camera's lens and



Pincushion Distortion



Barrel Distortion



Tangential Distortion

Figure 24: The effect of pincushion, barrel, and tangential distortion of the imaging of a grid pattern. ***LEFT COLUMN****: Grid pattern imaged.* ***RIGHT COLUMN****: Images that would be obtained of the grid pattern with the different types of distortion in the imaging system.*

sensor are not perfectly parallel, as is illustrated in the bottom right cell of Figure 24. The effect of this kind of distortion is illustrated in Figure 24, showing its effect when imaging a grid pattern. Tangential distortion can be removed from images using the following equations:

$$x_{new} = x + (2p_1 y + p_2(r^2 + 2x^2)) \qquad \textit{Eq. 2.6}$$

$$y_{new} = y + (p_1(r^2 + 2y^2) + 2p_2 x) \qquad \textit{Eq. 2.7}$$

where the p_i terms are the tangential distortion coefficients. All distortion coefficients, as well as the parameters that map cameras to the pinhole camera model, can be estimated through a calibration process. The parameters that are needed to describe the camera's corresponding pinhole camera model are known as its intrinsic parameters. The intrinsic parameters, focal length and principal point coordinates, are commonly held in a matrix of the following form:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \qquad \textit{Eq. 2.8}$$

where c_x and c_y are the x and y coordinates of the principal point, respectively. While the principal point is the location where the optical axis intersects the image sensor, it is commonly not equal to the centre of the sensor due to imperfect alignment. The term f_x refers to the product of the focal length and number of pixels per millimetre of the sensor in the x direction. Since in some cases the pixels of image sensors may not be perfectly square, the same parameter but for the y direction (f_y) is included. On top of the parameters mentioned, camera calibration protocols for stereo cameras also estimate the rotation matrix and translation vector, the extrinsic parameters, describing the positional and orientational relationship between the two cameras in a stereo camera rig. There are many tools available for camera calibration, such as that within the OpenCV library (Bradski & Kaehler, 2000) and in MATLAB (Mathworks, Natick, Massachusetts, US). They all require a number of stereo images to be obtained of a pattern with a known size. Typically, the pattern used is a checkerboard calibration pattern, where the size of the squares is known. The libraries analyse the set of stereo images taken of the pattern and provide estimates of the distortion coefficients, and intrinsic and extrinsic parameters.

2.3.4 RECTIFICATION

After distortion is removed from the input images, the process of rectification performs the mathematical mapping of the physical stereo camera rig to the ideal stereo model, which contains pinhole cameras of identical focal length and principal point locations, and with their sensors horizontally aligned and coplanar. The process of rectification returns images, known as rectified images, that appear as if they were taken by a stereo camera rig that perfectly complies with the ideal stereo model.

While mapping the systems to the ideal stereo model allows for easy triangulation of corresponding points to estimate their 3D coordinates in the scene, it also makes the process of finding corresponding points (stereo correspondence) less computationally expensive and more reliable. The reason is that, after mapping, the images are horizontally aligned and each point in the left image can be found at some location on the same horizontal scanline in the right image, unless it is occluded or out with the camera's field of view. This means that stereo correspondence becomes a search along a single dimension instead of two. The aspect of the rectification process that aligns the images can be better understood by looking at the geometry that governs stereo camera systems, known as the epipolar geometry. This concept is better illustrated with a system where the cameras are angled towards some point in the scene rather than at approximately at infinity. This is illustrated in Figure 25 and will be used in the description of the epipolar geometry that follows. Looking at the left cell of this figure, point C in the left image could be from a scene point located anywhere along the green projection line. In projecting this line onto the right image, it can be seen that a straight line is formed (also shown as a green line). This means that the search in the right image for the point that corresponds to point C, only needs to be carried out along the green line shown on the right image. Looking at the right cell in this figure, now consider the possible locations in the scene of many points along the green line in the right image (shown as blue projection lines). These lines all project onto the same line in the left image (also shown as a blue line). This means that for any point on the blue line in the left image, the corresponding point can be found on the green line in the right image. These lines are known as epipolar lines. Points in a different plane from point C (plane formed by points A, B and D), will have different epipolar lines. In the process of rectification, the images from both cameras are transformed

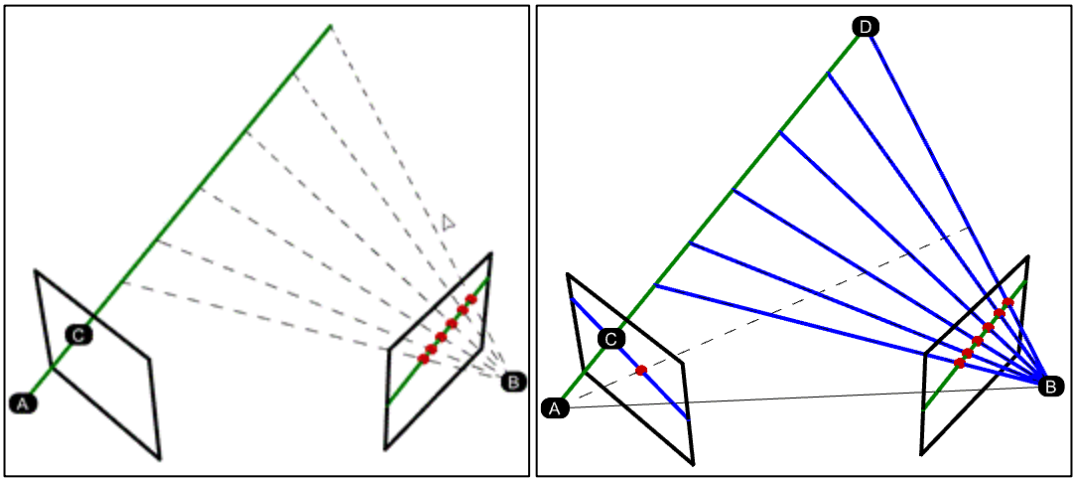


Figure 25: Illustrations to aid the description of epipolar geometry.

such that all epipolar lines are horizontal and aligned between the left and right images. The effect of rectification, and also distortion removal, is illustrated in Figure 26.

Rectification can also be performed without knowing the calibration parameters of the system. This process requires matches to be made between the input images, from which the transformations that need to be applied to rectify the images can be determined. More detail on this is provided in Bradski and Kaehler's book, *Learning OpenCV: Computer Vision with the OpenCV Library* (Bradski & Kaehler, 2008). The drawbacks of rectifying in this way are that sometimes the rectified images can appear more distorted than those obtained though the calibration-dependent rectification route, making the stereo correspondence difficult; the images still need to be undistorted prior to rectification, which requires a calibration procedure; and any reconstructions obtained would only be proportionally accurate but not to scale (their true size would not be known; Bradski and Kaehler, 2008).

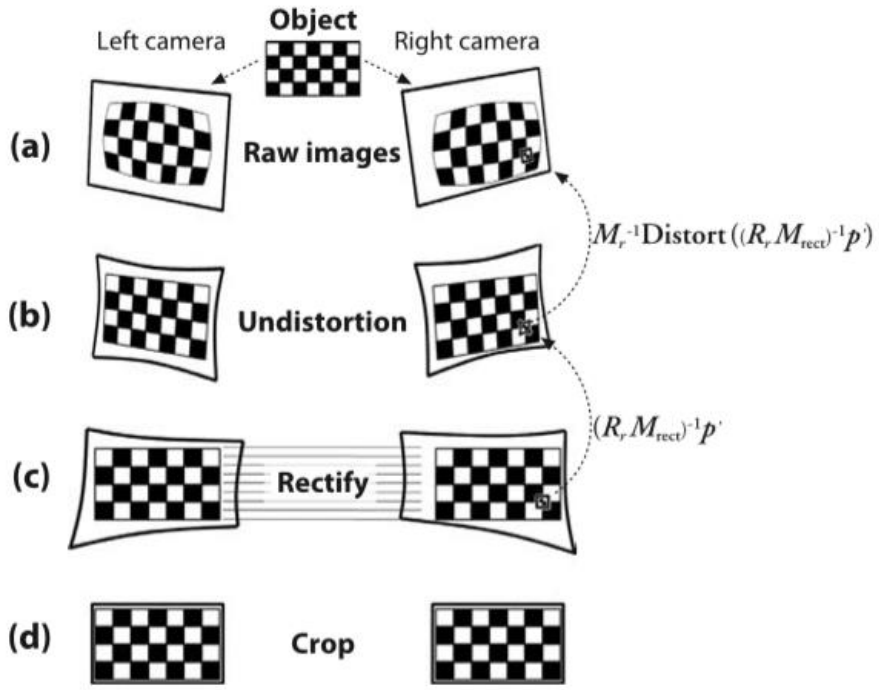


Figure 26: Illustration of the effect that distortion removal and rectification have on input stereo images of a checkerboard pattern. (Bradski and Kaehler, 2008).

2.3.5 Stereo Matching

Following distortion removal and rectification of the images, the only remaining information that is required to estimate 3D positions (using equations 2.1 to 2.3) of points in the scene are their corresponding locations in the right and left images. As mentioned previously, the

disparity of a particular point in the scene is the difference in location of that point in the left (the reference image) and right images in units of pixels. The problem of identifying particular points in both images, the stereo correspondence problem, is challenging and an extremely active area of research (Kai et al., 2006). The algorithms used to find correspondences, or perform stereo matching, can be broadly categorised as sparse and dense methods. Sparse methods produce point clouds that are not as densely populated as those produced using dense algorithms. Sparse methods operate by detecting features in the reference image and then matching patches that contain those features to patches of the same size in the right image. Though these methods produce sparse point clouds, densification is possible through interpolation. The use of these methods was more common before computer systems became powerful enough to more rapidly run the more computationally expensive dense stereo matching algorithms (Szeliski, 2011). On the other hand, they are sometimes used when sparse but highly reliable matching is desired. Dense methods are subdivided into local and global algorithms. Local algorithms attempt to find a match in the right image for each pixel in the reference image by computing the similarity, by some metric, and selecting the location with the greatest similarity. Since individual pixels do not contain enough information to enable reliable matching, matching for a pixel is carried out using a patch-based method, whereby a patch (sometimes termed support window) centred at that pixel is matched to a patch of the same size in the right image. Figure 27 has been provided to complement the description that follows for how matching is performed. Matching operates by computing the similarity between patches in the right image (one example shown with a blue dashed line), within a defined search range (green), and the support window (blue) for a particular pixel (red) in the reference image and selecting that which is most similar. Commonly used similarity metrics include the sum of squared differences, sum of absolute

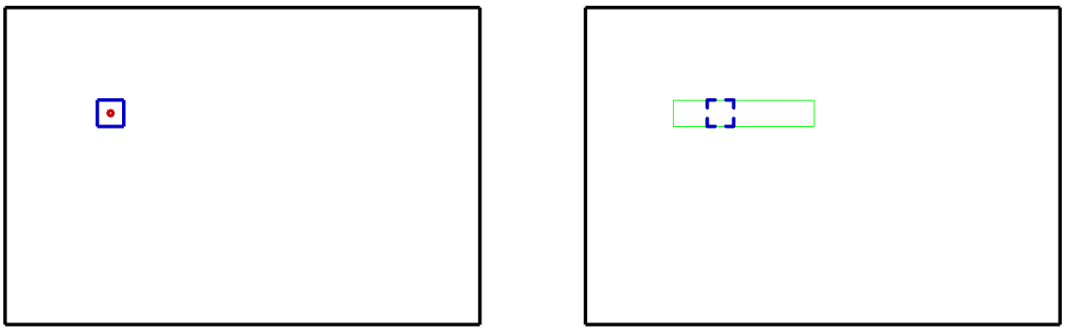


Figure 27: Illustration of patch based stereo matching. ***Red Dot*** *– Point to be matched,* ***Solid Blue Rectangle*** *– Support window,* ***Green Rectangle*** *– Search window,* ***Dashed Blue Rectangle*** *– Example of a patch in the right image that will be tested for similarity.*

differences and cross correlation (Szeliski, 2011). Local methods, while less computationally expensive, are less accurate than global methods, particularly where the texture density is low (Geiger et al., 2011; Yang et al., 2016). Global methods solve the correspondence problem through a global optimisation which assumes some degree of smoothness. They operate by minimising a cost function containing data and smoothness terms.

Another matching strategy to mention, since it has been used by researchers, such as Corona and colleagues (Corona, Mitra, Wilson, & Soliz, 2002), in ONH stereo reconstruction work, is the coarse-to-fine method – a type of hybrid local-global method. The coarse-to-fine method is based on matching patches but does so in a hierarchical way. Matches are first found for few but large patches. The patches are then subdivided, and the resulting new patches are matched, using the disparity information from the previous step to constrain the search. This process is repeated until some end criteria is met (e.g., minimum window size). This method is therefore conducted like a local method, but by constraining the search using information obtained in the previous step, some degree of smoothness is enforced.

2.3.6 Convergent Stereo Camera Geometry

In the case of convergent stereo rigs, the calculation of the 3D coordinates of scene points differs to that for parallel stereo camera rigs. Figure 27 illustrates a convergent stereo camera model. The relationship that relates the positions of a particular scene point in both images to its Z coordinate in the scene is described by equations 2.9 and 2.10. Equations 2.2 and 2.3, mentioned previously, can also be used in this case to estimate the X and Y scene coordinates of that scene point.

$$x_R = f\frac{(fsin(\theta) + x_L cos(\theta))Z - fbcos\left(\frac{\theta}{2}\right)}{(fcos(\theta) - x_L sin(\theta))Z + fbsin\left(\frac{\theta}{2}\right)} \qquad \textit{Eq. 2.9}$$

$$y_R = f\frac{y_L Z}{(fcos(\theta) - x_L sin(\theta))Z + fbsin\left(\frac{\theta}{2}\right)} \qquad \textit{Eq. 2.10}$$

The parameters used in equations 2.9 and 2.10 have the same definitions as those detailed for the parallel stereo configuration but a new parameter, Θ, has been introduced, which denotes the angle of convergence of the cameras.

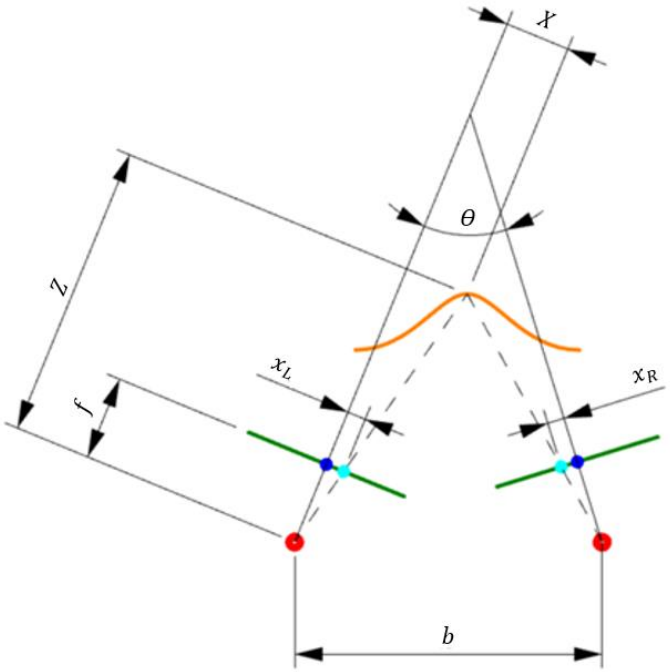


Figure 28: Schematic of the convergent stereo model. ***Orange*** *– Object in the scene,* ***Red*** *– Projection centres,* ***Blue*** *– Principal points,* ***Green*** *– Image sensors,* ***Cyan*** *– Sensor intersection points of rays emanating from a point on the object in the scene.*

CHAPTER 3

A SCOPING REVIEW ON COMPUTER STEREO VISION-BASED 3D SURFACE RECONSTRUCTION OF THE OPTIC NERVE HEAD

INTRODUCTION

In order to provide a solid grounding for this work, a scoping review of the work carried out by others on the application of computer stereo vision to the task of ONH 3D reconstruction, and the various imaging devices used, was conducted. This review is also aimed to be the basis of an appropriate publication on the matter. The Preferred Reporting Items for Systematic Reviews and Meta-Analyses – Scoping Reviews (PRISMA-ScR) guidelines and checklist (Tricco et al., 2018) were used to guide this review. The sections of this chapter are numbered to correspond to the numbered essential reporting items in the PRISMA-ScR checklist.

3.2 STRUCTURED SUMMARY

3.2.1 BACKGROUND AND OBJECTIVES

Computer stereo vision remains an attractive technique for quantitative imaging of the ONH at low-cost. A number of works have explored the idea but the full extent of what has been done until now is unclear. The objective of this review was to identify the different computer stereo vision methods which have been developed, and imaging devices used, for this task and gauge the degree to which they have been validated.

3.2.2 DESIGN

The Scopus, Web of Science and PubMed databases were systematically searched. The main eligibility criteria were to select articles written in the English language, which focus on computer stereo vision techniques for the task of 3D reconstruction of the ONH from stereo images. Data were charted on the imaging device used, stereo matching algorithm presented, and testing performed.

3.2.3 RESULTS

Through the screening process, 34 eligible articles were identified. The majority of articles detailed either a local or hybrid stereo matching method, while only 3 detailed a global method. Most works performed testing on the human ONH, but few carried out solid quantitative validation. A number of different imaging devices were identified in the different works and included simultaneous stereo fundus cameras and monoscopic fundus cameras.

One device was identified which was developed specifically to be used for ONH 3D surface imaging using computer stereo vision and intended to be low-cost.

3.2.4 CONCLUSIONS

While a great number of different approaches have been explored and developed, solid quantitative validation is lacking. On top of this, not all works used images from a simultaneous stereo imaging device, which is required for quantitative work.

3.3 RATIONALE

The idea of applying stereo vision techniques to stereo images of the ONH to produce depth maps seems to have been first documented in 1969 by Crock and Parel (Crock & Parel, 1969), but may have been documented earlier. The researchers in such work generated depth maps of an ONH from stereo images by using a stereoplotter – an instrument which is typically used to create contour lines for terrain maps. This was a very time-consuming process and the equipment required was not inexpensive, prohibiting it from use clinically. In 1973, David G. Falconer (Falconer, 1973) documented the use of computers to, more rapidly and inexpensively, perform the task of ONH reconstruction from stereo images. This technique involved using the computer to match regions in monochromatic stereo images (128 x 128 pixels) by using correlation as the similarity metric.

In the time between Falconer's work and the introduction of OCT, HRT and GDx, the application of computer stereo vision for ONH topographical reconstruction was realised commercially in a number of quantitative ONH imaging devices (Lorenz & Borruat, 2007). These include the Topcon IMAGEnet (Topcon Instruments, Paramus, NJ, US), Humphrey Retinal Analyzer (Humphrey, San Leandro, CA, US), Optic Nerve Head Analyzer (Rodenstock Instruments, Danbury, CT, US) and Glaucoma-Scope (Ophthalmic Imaging Systems, Sacramento, CA, US). The Topcon IMAGEnet and Humphrey Retinal Analyzer were systems which generated topographical reconstructions of the ONH based on images supplied to them, taken using a fundus camera. The Optic Nerve Head Analyzer and Glaucoma-Scope were devices which performed imaging and reconstruction. The Optic Nerve Head Analyzer contained a projection system that projected 14 stripes onto the ONH, and a stereo camera which captured stereo images. The depth of points on the ONH were inferred from the amount of deflection of the stripes. The Glaucoma-Scope consisted of a projection system

and a single camera. It projected 25 lines onto the ONH at an angle, so that the lines would appear deflected to the camera which acquired the image. This system also calculated depth based on the degree of deflection of the lines.

Even though these devices are no longer used today (Lorenz & Borruat, 2007), work by researchers on the application of computer stereo vision for ONH surface reconstruction continues as its low-cost nature remains appealing and ever improving fundus camera capture quality (Bansal et al., 2013), together with improvements in stereo vision algorithm performance and increasing computational power (Bhowmik & Appiah, 2018), permit improvements to the technique. In fact, one stereo fundus camera currently on the market, the Kowa nonmyd WX^{3D} (Kowa Company Ltd., Tokyo, Japan), has an ONH 3D reconstruction function which exploits computer stereo vision. A number of contributions on this topic have been detailed in the literature, showing different approaches, but the extent of such work is not clear and to the author's knowledge, no review has been conducted previously to synthesise such information. Thus, a scoping review was conducted to systematically map such research. In order to conduct this scoping review in a highly transparent and repeatable manner, the PRISMA-ScR protocol (an extension of PRISMA for scoping reviews; Tricco et al., 2018) was followed. The preferred reporting items checklist associated with this protocol can be found in Appendix 3A.

3.4 Objectives

The following research question was formulated: What are the different computer stereo vision techniques, including the imaging devices used, that have been applied to the task of 3D surface reconstruction of the ONH, and to what extent has their performance been evaluated?

3.5 Protocol and Registration

Given the nature of the study, no formal protocol registration was required.

3.6 Eligibility Criteria

In order to be included in the review, manuscripts needed to:

- Detail a computer stereo vision technique, including the stereo matching algorithm employed, that was used for 3D surface reconstruction of the ONH from stereo images.
- Have been written in the English language (since it is the only language that can be read by the author).

Manuscripts were excluded if they:

- Detail the use of the systems or devices that are/were commercially available for 3D surface reconstruction of the ONH (e.g., the Optic Nerve Head Analyzer) and do not provide information on the algorithms used by them.
- Are short (1-2 page) conference abstracts (since they lack in technical detail).
- Lack enough detail to permit categorisation of the stereo matching method used.

No restriction was placed on the year of publication.

3.7 Information Sources

In order to identify potentially relevant scientific manuscripts, the Scopus (Ballew, 2009), Web of Science (Clarivate Analytics, 1997) and PubMed (U.S. National Library of Medicine, 1988) databases were searched for all years available. The latest search was conducted on the 31st of August 2020. The final search results were exported into Microsoft Excel (Microsoft Corp, Redmond, Washington, US) and duplicates were removed by the author.

3.8 Search

The search strategy, shown in Table 1, was developed, by the author, to include papers mentioning terms in the title relating to stereophotogrammetry or 3D reconstruction, and terms relating to the ONH or retina. The search strategy was not peer-reviewed. The corresponding search strings used for the three different databases are provided in Appendix 3B. A filter was applied on the searches to exclude articles not written in the English language.

Table 1: Search strategy employed in the stereo vision scoping review.

	AND	
OR	stereo	optic **AND** nerve **AND** head
	stereo-matching	optic **AND** nerve
	stereo **AND** matching	optic-nerve
	stereophotogrammetry	optic **AND** disk
	stereo-photogrammetry	optic **AND** disks
	stereo **AND** photogrammetry	optic-disk
	stereoscopic	optic **AND** disc
	3D **AND** reconstruction	optic **AND** discs
	3-D **AND** reconstruction	optic-disc
	3-dimensional **AND** reconstruction	optic **AND** cup
	three-dimensional **AND** reconstruction	optic **AND** cups
	three dimensional **AND** reconstruction	optic-cup
	3D **AND** visualization	fundus
	3-D **AND** visualization	retina
	3-dimensional **AND** visualization	retinal
	three-dimensional **AND** visualization	
	three dimensional **AND** visualization	
	3D **AND** visualisation	
	3-D **AND** visualisation	
	3-dimensional **AND** visualisation	
	three-dimensional **AND** visualisation	
	three dimensional **AND** visualisation	
	depth **AND** analysis	
	topographic **AND** measurements	
	topographic **AND** measurement	

3.9 Selection of Sources of Evidence

The author evaluated, in order, the titles, abstracts and full texts of the articles, eliminating articles at each stage which did not meet the eligibility criteria, to identify the relevant publications.

3.10 Data Charting Process

A data-charting form was developed by the author and implemented in Microsoft Excel (Microsoft Corp, Redmond, Washington, US). Data charting was performed by the author. No form of peer review was performed for the data-charting form.

3.11 Data Items

Where present, data were abstracted on:

1. The imaging device used to capture the stereo images, including:
    a. The name of the device.

    b. Whether it was commercially available or a research prototype.
    c. Whether it can capture simultaneous stereo images or not.
2. The stereo matching algorithm, including:
    a. Whether the stereo matching method appears to be global, hybrid or local.
    b. A description of the stereo matching method.
    c. The similarity metric or metrics used.
    d. If any consistency checks were used, to remove unreliable matches.
    e. Any novel aspects.
3. The testing performed, including:
    a. The subject (e.g., eye phantom, in vivo).
    b. How ground truth data was acquired.
    c. If validation was performed, and if it was qualitative or quantitative in nature, and what it entailed.

3.12 Critical Appraisal of Individual Sources of Evidence

A critical appraisal of the individual sources of evidence was not conducted since it is not relevant to the objectives of this scoping review.

3.13 Synthesis of Results

Articles were categorised by the type of stereo matching algorithm detailed (i.e., local, hybrid or global) and maps were created for each, displaying the different articles, key aspects of the stereo matching algorithms used, and testing performed. The maps were created in order to provide some insight into how the articles are related and how they fit into the bigger picture. On top of this, the imaging devices used were summarised narratively.

3.14 Selection of Sources of Evidence

The majority of the records were identified through Web of Science (n = 398), followed by Scopus (n = 295) and PubMed (n=193). A total of 418 articles were screened using the eligibility criteria. Finally, 34 articles were included in the final qualitative synthesis. A flow chart of the selection process, showing the number of articles at each stage, is provided in Appendix 3C.

3.15 Characteristics of Sources of Evidence

The majority of the articles detail the use of either a local or hybrid stereo matching algorithm, 16 and 15 respectively. Only 3 articles detail a global algorithm. In the work detailed in 16 articles, a simultaneous, or approximately simultaneous, stereo fundus camera was used for image acquisition. The use of a monoscopic fundus camera, only permitting asynchronous capture, was used in 5 works, while the imaging device used was not detailed in 15 records. 5 works performed testing of their method on ONH-like models. Of these, 4 performed some form of quantitative validation. Practically all, 32, works performed testing in vivo but only 18 included some form of quantitative validation. In 9 articles, relevant ONH parameters, such as the cup-to-disc ratio, were estimated from cup and disc margins delineated on the reconstructions. 2 articles delineated the margins using a semi-automated procedure, and 7 using a fully automated procedure.

3.16 Critical Appraisal within Sources of Evidence

A critical appraisal within sources of evidence was not conducted since it is not relevant to the objectives of this scoping review.

3.17 Results of Individual Sources of Evidence

The key details of the imaging devices used, stereo matching algorithms implemented, and testing performed in the articles included in this synthesis are presented in the table in Appendix 3D.

3.18 Synthesis of Results

3.18.1 Stereo Matching Algorithms and Performance Evaluation

Maps of the research conducted in this area where a local, hybrid and global stereo matching algorithm was implemented are provided in Figure 29, Figure 30 and Figure 31, respectively. The maps detail the stereo matching algorithms used and testing performed in each of the works, and how each relates to other works.

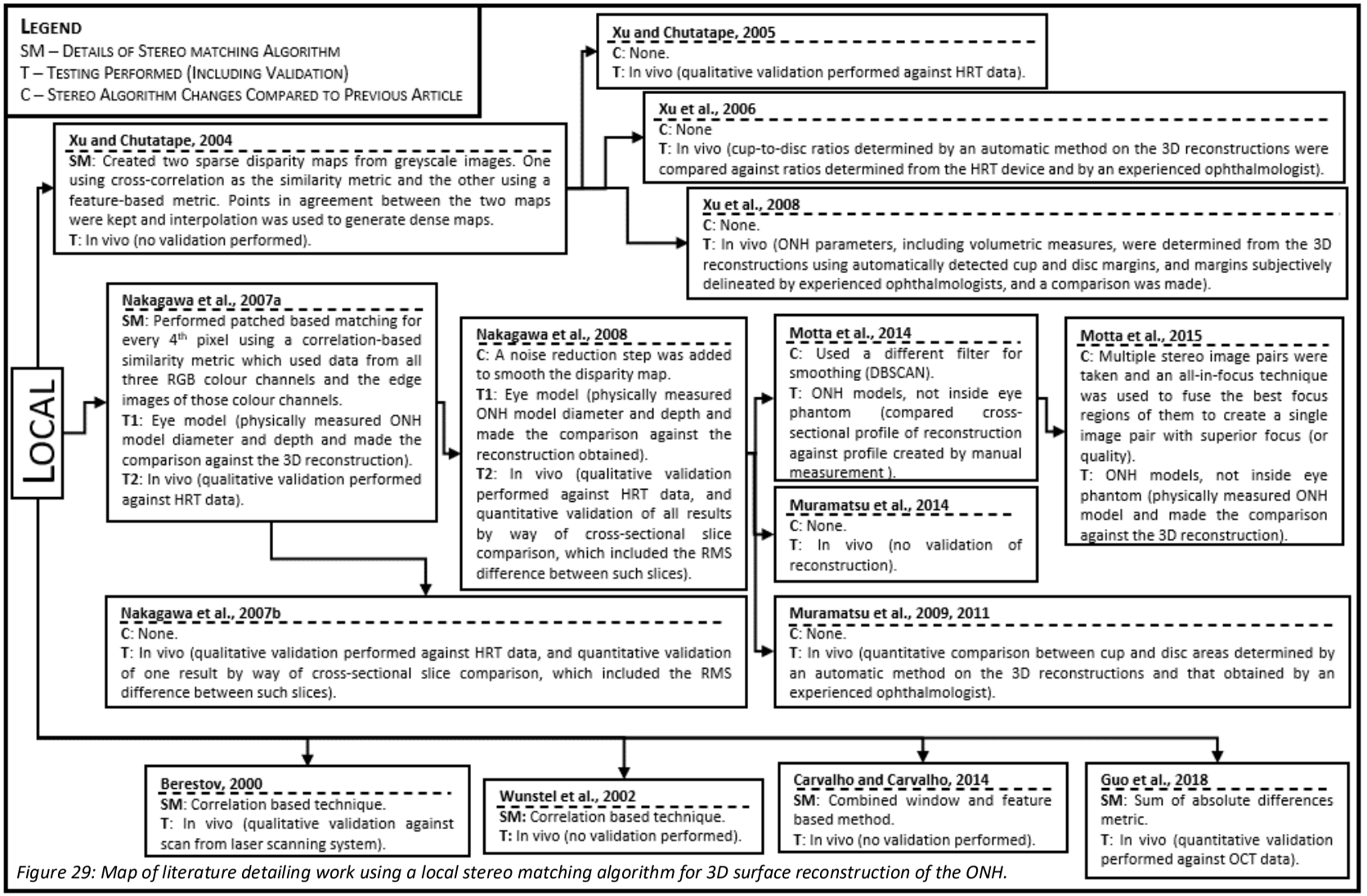


Figure 29: Map of literature detailing work using a local stereo matching algorithm for 3D surface reconstruction of the ONH.

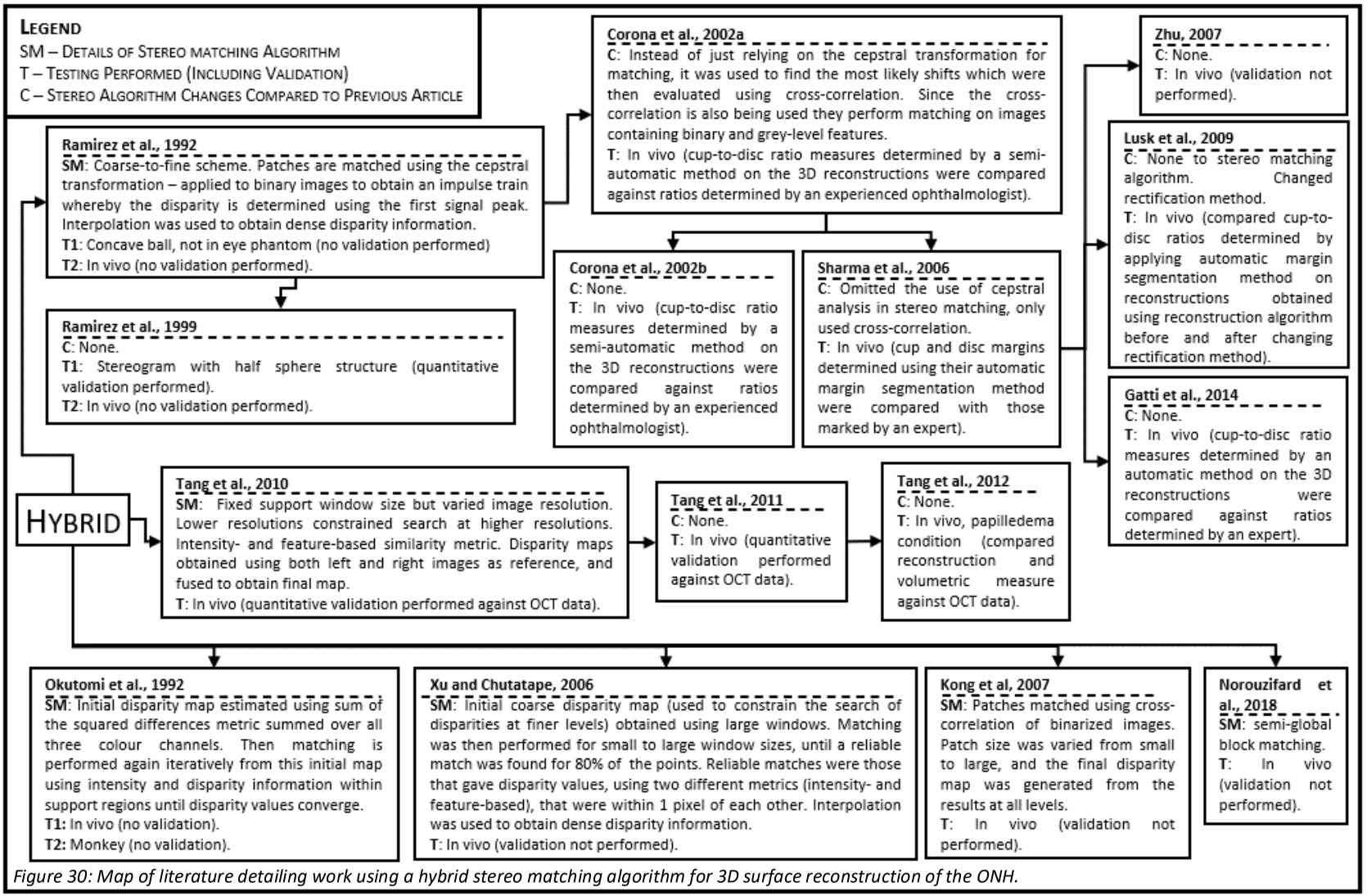


Figure 30: Map of literature detailing work using a hybrid stereo matching algorithm for 3D surface reconstruction of the ONH.

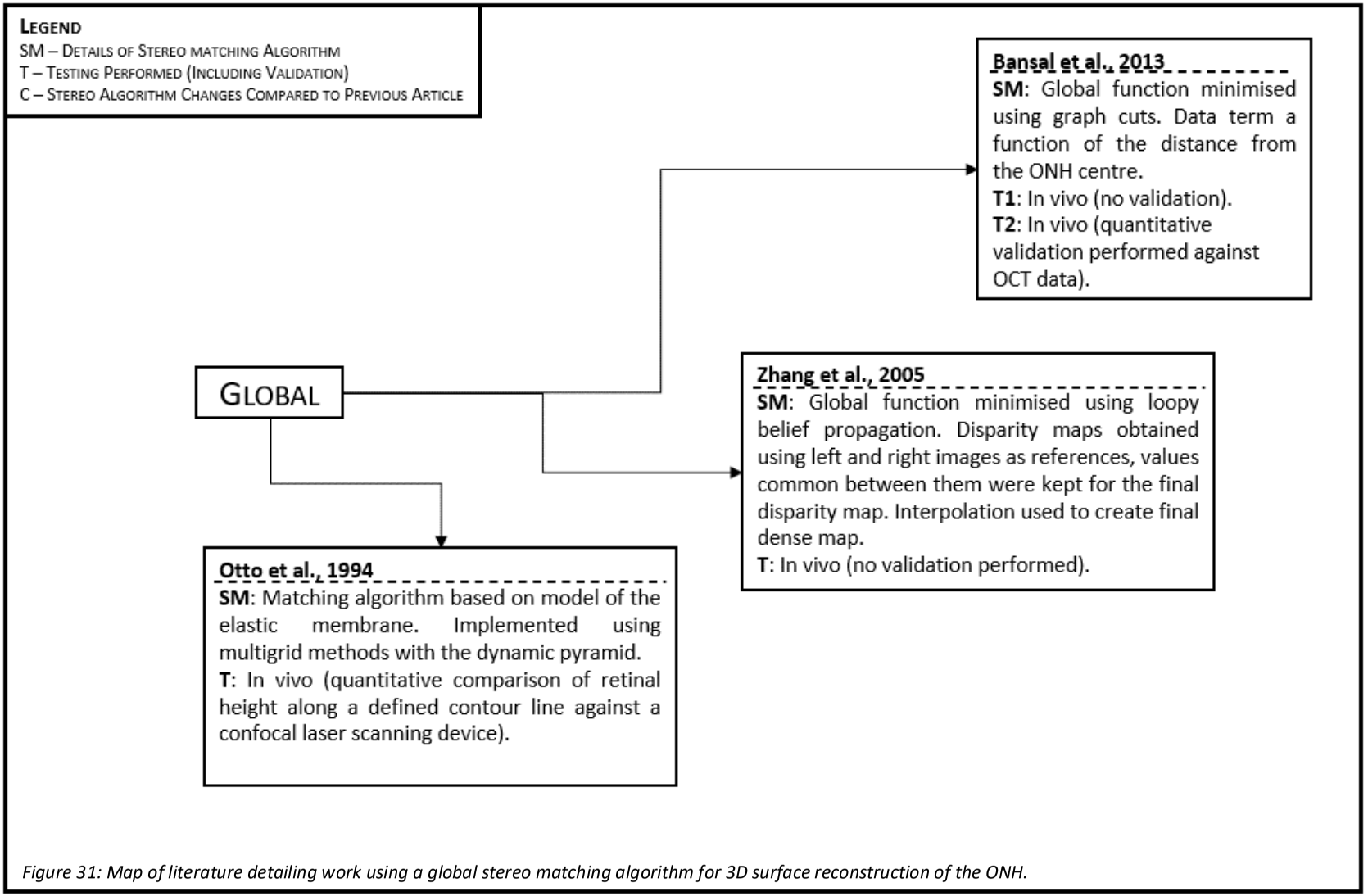


Figure 31: Map of literature detailing work using a global stereo matching algorithm for 3D surface reconstruction of the ONH.

Regarding performance evaluation, only the work by Nakagawa and colleagues (Nakagawa et al., 2008; Nakagawa, Hayashi, Hatanaka, Aoyama, et al., 2007; Nakagawa, Hayashi, Hatanaka, M.d, et al., 2007) involved quantitative validation where the reconstruction obtained by their technique was not scaled to match the size of the ground truth scan. In the work by Bansal and colleagues (Bansal et al., 2013) and Tang and colleagues (Tang et al., 2010, 2011), the disparity maps obtained were scaled along each dimension to match the OCT data.

3.18.2 Imaging Systems

The simultaneous stereo fundus cameras seen within the articles were the Nidek 3Dx stereo fundus camera (Nidek, Gamagori, Aichi, Japan), the Topcon TRC-SS2 (Topcon, Tokyo, Japan) and a low-cost research prototype stereo fundus camera. The former two devices were previously commercially available but are no longer. These capture the stereo images as split frame images, either onto a 35 mm slide or an image sensor (Schacknow & Samples, 2010). The latter device, detailed in three articles (Carvalho & Carvalho, 2014; D. Motta et al., 2015; D. A. Motta et al., 2014), was designed to be specifically used for 3D surface reconstruction of the ONH region and low-cost such that it could be widely adopted for glaucoma diagnostics. The optical design of the device was presented by Motta and colleagues in 2014 (D. A. Motta et al., 2014), and the constructed system was detailed by Carvalho and Carvalho in the same year (Carvalho & Carvalho, 2014). The 3D computer-aided design (CAD), with the shell omitted, and a photograph of the system are shown in Figure 32.

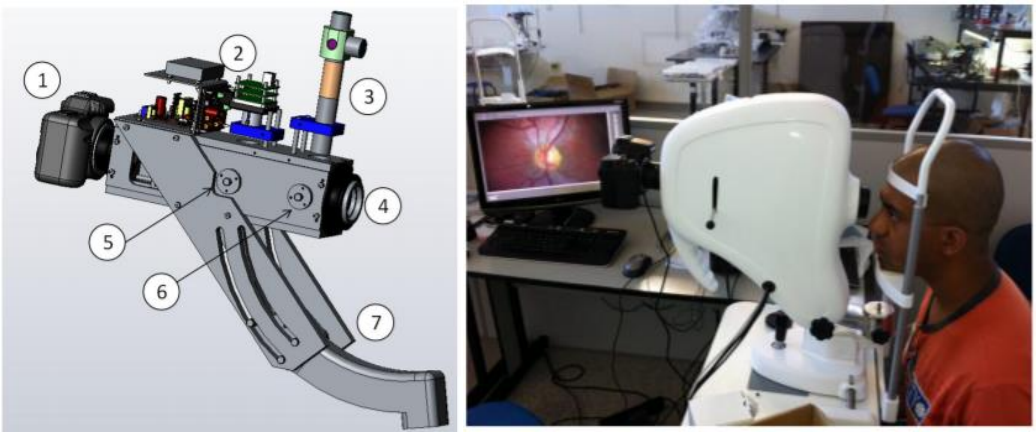


*Figure 32: CAD design (with shell omitted; **LEFT**) and photograph (**RIGHT**) of the low-cost stereo fundus camera presented by Carvalho and Carvalho (Carvalho and Carvalho, 2014). 1 – DSLR camera, 2 – Capture and flash control board, 3 – White and infrared LEDs, 4 – Objective.*

Another stereo fundus camera that was seen within the articles is the Kowa WX-1 (Kowa Company Ltd., Tokyo, Japan), or a prototype of it. It captures stereo images in a not totally, but practically, simultaneous manner, by shifting of an aperture in a short time period – stated to be 0.14 s by Nakagawa and colleagues (Nakagawa et al., 2008).

Other works used images obtained using monoscopic fundus cameras where the stereo separation was created by shifting the whole device to the left and right, and sequentially capturing an image at each position. One of the works (Tang et al., 2012) used a Topcon TRC-50DX (Topcon, Tokyo, Japan), which contains a mechanism that constrains the left and right movement such that stereo images can be obtained with a similar baseline each time (Schacknow & Samples, 2010).

3.19 Summary of Evidence

In this scoping review, it was revealed that a number of different approaches have been explored for 3D surface reconstruction of the ONH from stereo images. The findings indicate that while plenty of different approaches have been developed, solid validation of them is lacking. Many articles presented techniques and obtained reconstructions in vivo but did not perform quantitative validation. Others scaled the reconstructions obtained in all dimensions to match ground truth data, permitting only validation of the shape of the reconstruction. Only Nakagawa and colleagues quantitatively validated the reconstructions obtained against ground truth data without such scaling. Since absolute size of the ONH is important in the evaluation of the ONH (Giaconi et al., 2016a), solid validation like this is likely important. Some of the works compared ONH parameters, such as the cup-to-disc ratio, obtained using a semi- or fully automated method on the reconstructions, against that determined subjectively by experts, without first performing solid quantitative validation of the reconstructions obtained.

Many of the works used imaging systems that do not permit simultaneous capture of the ONH, which is necessary to obtain quantitative reconstructions of a non-stationary target (as the eye is), using calibration data.

Only one low-cost imaging system, developed specifically for the purpose of obtaining 3D reconstructions of the ONH (Carvalho & Carvalho, 2014; D. Motta et al., 2015; D. A. Motta et al., 2014), was identified, and the level of validation of its performance was poor. In the ONH

model tests, the models used were not placed within an eye phantom and the ground truth data was acquired by manually measuring the concave ball-shaped objects. The in vivo work involved no form of validation.

3.20 Limitations

One limitation of this scoping review is that only articles written in English and indexed in the Scopus, Web of Science and PubMed databases were included, and therefore other relevant work, available in the grey or published literature, could have been missed. Another related limitation is that the search strings created are likely not optimal and some relevant texts could have been missed as a result. On top of this, the method used to conduct this review was not peer-reviewed. Moreover, article selection and data charting were performed solely by the author.

3.21 Conclusions

The objective of this scoping review was to identify the different works that have been carried out on the application of computer stereo vision for 3D surface reconstruction of the ONH from stereo images, and to understand to what extent the methods developed have been validated. On top of this, the review aimed to identify the different imaging devices which were used to capture the stereo images used. A number of different techniques, exploiting local, global and hybrid stereo matching algorithms, were identified but solid quantitative validation of their performance was lacking. The imaging devices used which were the most appropriate for quantitative work were commercially available, and likely expensive, simultaneous stereo fundus cameras. One device was identified which is a low-cost stereo fundus camera prototype.

3.22 Funding

This work was supported by the UK EPSRC Centre for Doctoral Training in Medical Devices and Health Technologies, University of Strathclyde (Grant Ref. EP/L015595/1).

CHAPTER 4

A SCOPING REVIEW OF FLUID-FILLED OPTICAL EYE PHANTOMS

INTRODUCTION

In order to provide some grounding for the development of a complex eye phantom, containing a cornea, lens and fluid-filled interior, which closely matches the human eye in terms of optical characteristics, a second scoping review on optical eye phantoms with such complexity was conducted. The PRISMA-ScR guidelines and checklist (Tricco et al., 2018) were used to guide this review. The sections of this chapter are numbered to correspond to the numbered essential reporting items in the PRISMA-ScR checklist.

4.2 STRUCTURED SUMMARY

4.2.1 BACKGROUND AND OBJECTIVES

Optical eye phantoms are valuable tools in work with ophthalmic instruments. Many of the more complex phantoms, featuring a cornea, lens and fluid-filled interior, have been detailed in the literature. The objective of this review was to identify the various phantoms of this kind which have been presented and briefly examine how they differ.

4.2.2 DESIGN

The Scopus, Web of Science and PubMed databases were systematically searched. The main eligibility criteria were to select peer-reviewed articles, written in the English language, which focus on optical eye phantoms with the above-mentioned optical complexity. Data were charted on their intended application, their optical design and retina implemented.

4.2.3 RESULTS

Through the screening process, 12 eligible articles were identified. The majority of the articles detailed eye phantoms which were specifically designed for OCT and refractive error/biometric studies. Phantoms designed for fundus imaging work and ultrasound studies were also identified. In most cases, the optical designs did not differ greatly, but their retinas did, being tailored for their intended application.

4.2.4 CONCLUSIONS

While quite a number of phantoms with similar optical designs have been detailed, the majority use costly custom optical components.

4.3 Rationale

Optical eye phantoms play a crucial role in the calibration and performance evaluation of various ophthalmic instruments as well as in studies such as those on refractive errors (Xie et al., 2014). Some feature a single lens for refractive power and are air-filled, while others are more complex, containing a cornea, lens, and fluid-filled interior. On top of this, a range of different retinas have been used, specific for their purpose. Regarding the more complex eye phantoms, which more accurately mimic the artefacts and aberrations seen in the human eye, a number of different versions have been presented/used in the literature. The range of those detailed in the literature, and how they differ, is unclear and to the author's knowledge, no review has been previously conducted to synthesise such information. Thus, a scoping review was conducted to systematically map this information. In order to conduct the review in a highly transparent and repeatable manner, the PRISMA-ScR protocol (an extension of PRISMA for scoping reviews; Tricco et al., 2018) was followed. The preferred reporting items checklist associated with this protocol can be found in Appendix 4A.

4.4 Objectives

The following research question was formulated: What are the different types of eye phantoms, which contain a cornea, lens and fluid-filled interior, to approximate the artefacts and aberrations of the human eye, that have been presented or used within the literature, and how do they differ?

4.5 Protocol and Registration

Given the nature of the study, no formal protocol registration was required.

4.6 Eligibility Criteria

In order to be included in the review, manuscripts needed to:

- Detail an eye phantom that had been constructed or used, which contained a cornea, lens and fluid-filled interior.

- Have been written in the English language (since it is the only language that can be read by the author).

Where the same eye phantom was detailed in two or more articles, the most recent article was kept, and the others excluded.

4.7 Information Sources

In order to identify potentially relevant scientific manuscripts, the Scopus (Ballew, 2009), Web of Science (Clarivate Analytics, 1997) and PubMed (U.S. National Library of Medicine, 1988) databases were searched for all years available. The latest search was conducted on the 2nd of September 2020. The final search results were exported into Microsoft Excel (Microsoft Corp, Redmond, Washington, US) and duplicates were removed by the author.

4.8 Search

The search strategy, shown in Table 2, was developed to include papers with eye phantom or eye model in the title, but in an attempt to restrict the results to those which detail physical eye phantoms and not computational or animal models, it was extended to only include those with construct (or synonym of) in the abstract and exclude those with the name of the common animal eye models used in research (e.g. rabbit) in the title. The search also contained elements to filter out papers related to gaze tracking. The corresponding search strings used for the three different databases are provided in Appendix 4B. A filter was applied to the searches to exclude articles not written in the English language.

Table 2: Search strategy employed in the eye phantom scoping review. ABS – Abstract.

	AND			**AND NOT**
OR	TITLE(**eye**)	TITLE(**phantom**)	ABS(**fabricate**)	TITLE(**rabbit**)
		TITLE(**model**)	ABS(**fabricated**)	TITLE(**mouse**)
			ABS(**construct**)	TITLE(**chick**)
			ABS(**constructed**)	TITLE(**rat**)
			ABS(**built**)	TITLE(**mathematical**)
			ABS(**build**)	TITLE(**gaze**)
			ABS(**assemble**)	TITLE(**tracking**)
			ABS(**assembled**)	TITLE(**movement**)

4.9 Selection of Sources of Evidence

The author evaluated, in order, the titles, abstracts and full texts of the articles, eliminating articles at each stage which did not meet the eligibility criteria, to identify the relevant publications.

4.10 Data Charting Process

A data-charting form was developed by the author and implemented in Microsoft Excel (Microsoft Corp, Redmond, Washington, US). Data charting was performed by the author. No form of peer review was performed for the data-charting form.

4.11 Data Items

Where present, data was abstracted on:

1. The cornea and lens:
    a. Materials
    b. Radii
    c. Thicknesses
2. The fluid interior:
    a. Materials
    b. Thicknesses
3. The retina

4.12 Critical Appraisal of Individual Sources of Evidence

A critical appraisal of the individual sources of evidence was not conducted since it is not relevant to the objectives of this scoping review.

4.13 Synthesis of Results

Articles were categorised by eye phantom application (e.g., intraocular lens [IOL] studies) and presented on a research map detailing the type of cornea, lens and retina used.

4.14 Selection of Sources of Evidence

The majority of the records were identified through Web of Science (n = 144), followed by Scopus (n = 138) and PubMed (n=50). A total of 177 articles were screened using the eligibility criteria. Finally, 12 articles were included in the final qualitative synthesis. A flow chart of the selection process, showing the number of articles at each stage, is provided in Appendix 4C.

4.15 Characteristics of Sources of Evidence

The majority of the articles (n = 6) detailed an eye phantom which was used for studies pertaining to refractive errors or biometric studies. 4 of the articles detailed eye phantoms used for studies with OCT systems. Few articles detailed eye phantoms developed specifically for fundus imaging and ultrasound studies, with only 1 article in each category. 1 article described the use of a commercially available eye phantom, modified for a specific purpose. 5 of the studies used an IOL to act as the crystalline lens while 6 used custom manufactured lenses. All but one article detailed a custom cornea, designed to closely match the dimensions of the human cornea.

4.16 Critical appraisal within sources of evidence

A critical appraisal within sources of evidence was not conducted since it is not relevant to the objectives of this scoping review.

4.17 Results of individual sources of evidence

The key details of the different eye phantoms (the optical materials, dimensions of optical components, and the retinas used), presented in the articles included in this synthesis, are presented in Table 3.

4.18 Synthesis of Results

A map showing the included articles, categorised in terms of the eye phantom's application, is provided in Figure 33. The optical designs of them do deviate to some degree but are not

Table 3: Design and application of various optical eye phantoms (containing 4 refractive surfaces) detailed in the literature. PMMA - Polymethyl Methacrylate.

Study	Application	Cornea				Lens				Anterior Chamber		Vitreous Humour		Retina
		Material (N)	R1 (Q)	R2 (Q)	T	Material (N)	R3 (Q)	R4 (Q)	T	Material (N)	T	Material (N)	T	
(Cao et al., 2018)	OCT Resolution Assessment	Glass (N/A)	7.6 (N/A)	6.8 (N/A)	N/A	N/A	N/A	N/A	N/A	Index-matching liquid (N = N/A)	N/A	Index-matching liquid (N = N/A)	N/A	$Ti-O_2$ PDMS multilayer spin-coated retina (R = -7.6)
(Hu et al., 2014)	OCT Resolution Assessment	Transparent Resin (N/A)	7.8 (N/A)	6.8 (N/A)	0.55	Transparent Resin (N/A)	12 (N/A)	-6 (N/A)	3.9	Deionsided Water (N = N/A)	2.95	Deionsided Water (N = N/A)	18.4	3D USAF 1951 test chart (R = -12)
(Xie et al., 2014)	Fundus Imaging Studies	PMMA (N = 1.492)	7.72 (-0.44)	7.41 (0)	0.55	N/A (IOL: Tecnis Z9000, N = 1.47)	12.7173 (-1.30613)	-15.7226 (0)	0.912	Balanced Saline Solution (N = 1.3342)	3.994	Balanced Saline Solution (N = 1.3342)	18.405	SLA created retina, R = -12. They created an angle scale on the retina.
(Corcoran et al., 2014)	Wide-Field OCT Studies	Fused Silica (N = 1.46)	7.72 (-0.26)	6.5 (0)	0.55	CaF_2 (N = 1.44)	11.22 (0)	-5.9 (-0.55)	3.93	Water (1.33)	3.05	Water (1.33)	16.32	Used SLA to create different retinas: embedded feature, axial layers and bullseye target.
(Arianpour et al., 2013)	Refractive Error Studies	Fused Silica (N = 1.458)	7.8 (-0.55)	6.7 (0)	1	Fused Silica (N = 1.458)	11.462 (0)	-11.019 (-3.25)	3.7	Distilled Water (1.334)	4.45	Distilled Water (1.334)	16.64	Curved fibre bundle (R=-11) - relayed to CMOS
(Agrawal et al., 2012)	OCT Point Spread Function Characterisation	PMMA (N = N/A)	7.82 (N/A)	4.14 (N/A)	0.55	PMMA (N = N/A)	11.99 (N/A)	-5.99 (N/A)	3.9	Water (N/A)	3.5	Water (N/A)	18.4	Transparent epoxy retina with sparsely distributed silica-gold nanoshells.
(Birkner et al., 2011)	Biometric Device Development	PMMA Rigid Contact lens (N = 1.484)	9.9 (N/A)	11.7 (N/A)	N/A	BK7 (N = 1.515)	11.7 (N/A)	-11.7 (N/A)	N/A	Water (1.328)	N/A	Water (1.328)	N/A	Reflective Surface (R = N/A)
(Taketani and Hara, 2011)	IOL Studies	PMMA (N = N/A)	7.67 (-0.13)	7.20 (0)	1	Various IOLs were placed inside for testing: Acryfold AC60AD, Acrysof IQ SN60WF and Tecnis ZA9003				5% Balanced Saline Solution (N = N/A)	5	5% Balanced Saline Solution (N = N/A)	Varied	Flat retina: painted dark grey matte
(Bakaraju et al., 2010)	IOL, Corrective glasses or Contact lens Studies	Fluiropolymer (N = 1.376)	7.75 (-0.20)	6.40 (-0.20)	0.55	Boston EO (N = 1.429)	11.85 (-5)	-6.35 (-2.75)	3.6	Deionsided Purified Water (N = 1.334)	3.45	Deionsided Purified Water (N = 1.334)	14.875	CMOS Sensor
(McKelvie et al., 2009)	IOL Studies	Boston XO (N = N/A)	N/A (N/A)	N/A (N/A)	N/A	Various IOLs were placed inside for testing: AcrySof IQ IOL SN60WF aspheric, Tecnis ZA9003 aspheric, and Adapt Advanced Optics.				Millipore-filtered water (N = N/A)	N/A	Millipore-filtered water (N = N/A)	N/A	Retinal back plate
(Ohnuma et al., 2000)	IOL Studies	Photographic lens (f = 35 mm)				IOL				Water		Water		CCD Sensor
(Brand et al., 1994)	Ultrasound Studies	KEL-F (N = N/A)	7.15 (N/A)	6.85 (N/A)		IOL				Distilled Water	N/A	Distilled Water	N/A	10mm thick polyethylene membrane

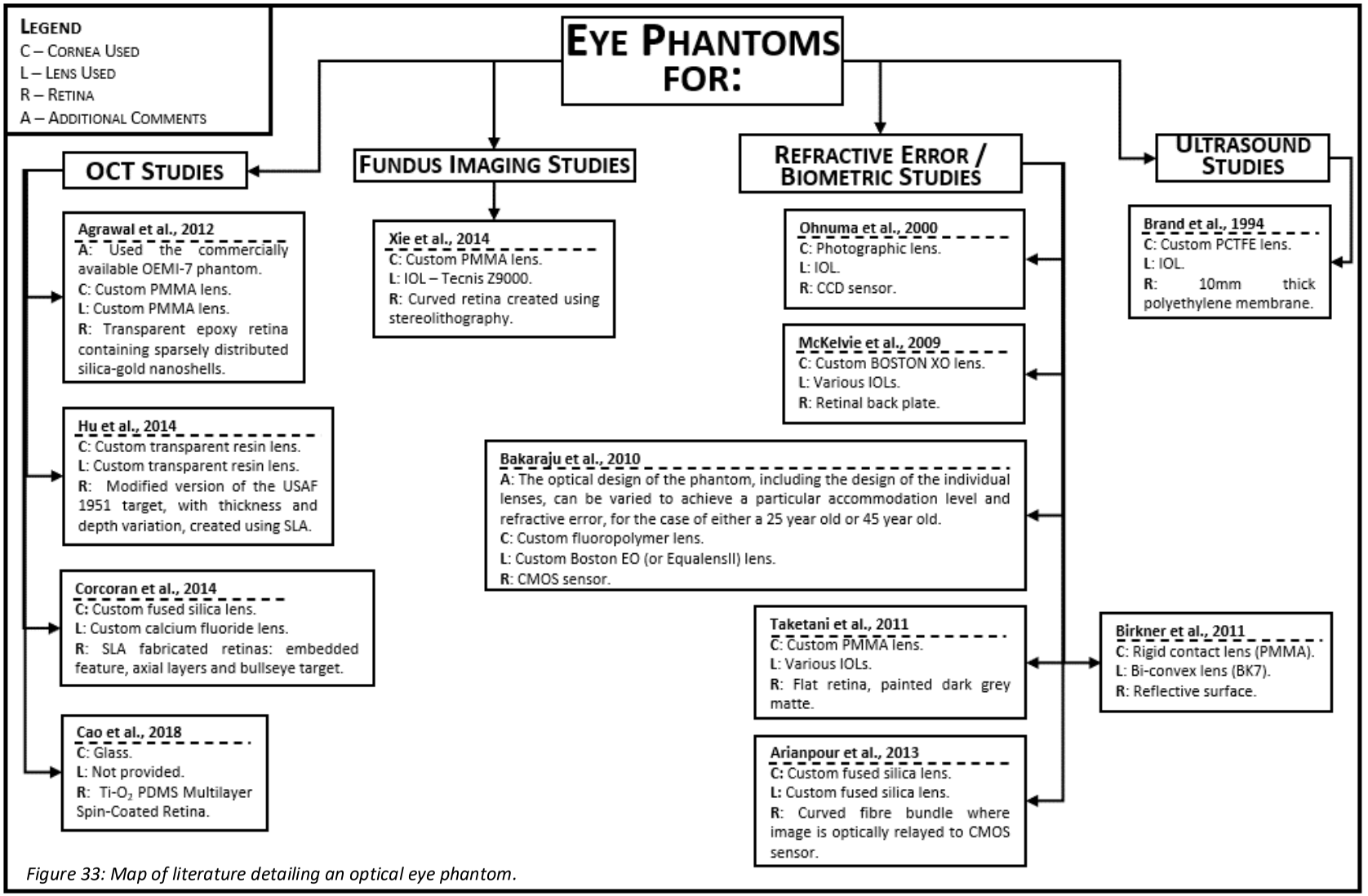


Figure 33: Map of literature detailing an optical eye phantom.

substantially different. Regarding their corneas, the majority have similar dimensions and were custom made. Some, such as that produced by Corcoran and colleagues (Corcoran et al., 2014), have an aspherical surface, which likely allowed for a closer match to the human eye in terms of aberrations. More variation was seen in the lens used in the different phantoms. Many groups used IOLs, one used a stock bi-convex lens, while others used custom fabricated lenses. In terms of the material used to fill the interiors of the eye phantoms (anterior and vitreous chambers), practically the same material was used in each. They all contained water but with slight variations. Some used water that was deionised, distilled, or filtered. Regarding the retinas used, a wide variety were seen, which were tailored specifically to the phantom's intended application.

4.19 Summary of Evidence

This scoping review revealed that a number of eye phantoms, of an optical complexity where they contain a cornea, lens and fluid-filled interior, have been detailed in the literature. Excluding some cases, the optical designs seen did not deviate greatly. This is particularly the case for the design of the corneas seen. The retinas used however did, which could be expected as they were each designed for different and specific purposes. The majority of the designs, especially the ones likely to be the most optically accurate, used custom made lenses, which likely makes these phantoms extremely expensive to fabricate.

4.20 Limitations

One limitation of this scoping review is that only articles written in English and indexed in the Scopus, Web of Science and PubMed databases were included, and therefore other relevant work, available in the grey or published literature, could have been missed. Another related limitation is that the search strings created are likely not optimal and some relevant texts could have been missed as a result. On top of this, the method used to conduct this review was not peer-reviewed. Moreover, article selection and data charting were performed solely by the author.

4.21 Conclusions

The objective of this scoping review was to identify the various complex eye phantoms that have been detailed in the literature and to examine how they differ. A number of different eye phantoms were identified. It was seen that for the majority of those identified, their optical designs did not differ by a great amount. Custom optics were a feature of most which likely makes these phantoms very expensive to fabricate.

4.22 Funding

This work was supported by the UK EPSRC Centre for Doctoral Training in Medical Devices and Health Technologies, University of Strathclyde (Grant Ref. EP/L015595/1).

CHAPTER 5

INITIAL FEASIBILITY ASSESSMENT OF SLIT LAMP-BASED RECONSTRUCTION BY PERFORMING UNCALIBRATED RECONSTRUCTION IN AN EYE PHANTOM

5.1 Introduction

Given that stereo vision-based reconstruction had not been documented before, a basic initial trial was needed to check if it was possible and that no fundamental flaw existed. This chapter presents such a trial, which was conducted at the beginning of the project. A simple setup of the imaging system, composed of webcams and an indirect lens mounted to a slit lamp, and basic stereo vision algorithms, were created and tested on simple ONH targets in a purpose built scaled-up simple eye phantom. In order to simplify matters further at this stage, the ONH targets had speckle patterns painted onto their surfaces to make stereo matching more straightforward. The algorithms used only permitted uncalibrated results to be obtained and therefore only a qualitative analysis could be done at this stage, however this was sufficient to assess initial feasibility.

5.2 Eye Phantom Development

5.2.1 Brief Overall Description

The eye phantom developed was a simplified and scaled-up version of a normal adult human eye and featured internal light sources so that illumination of the retina could be provided from inside. 3D CAD rendered views, including a colour-coded half-sectional view, and a photograph of the phantom are shown in Figure 34.

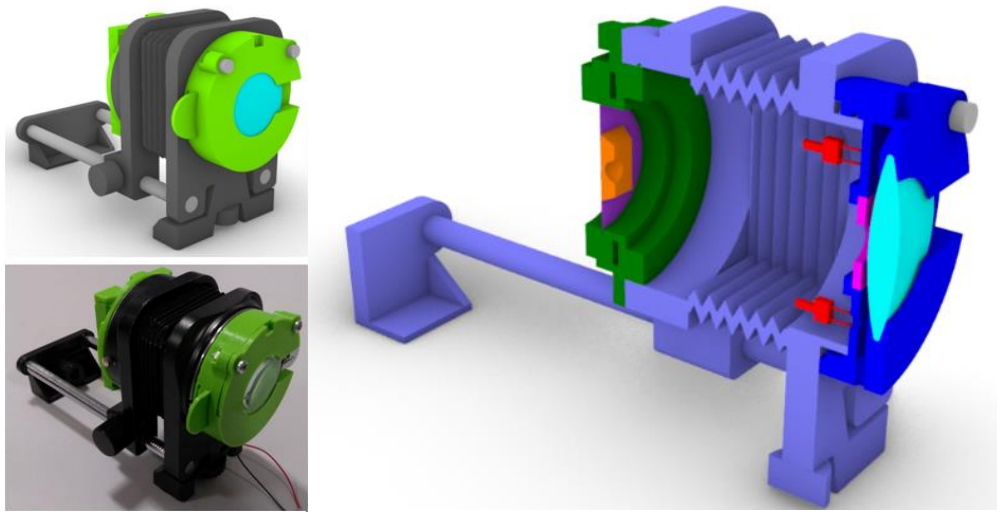


Figure 34: 3D CAD rendered views and photographs of the eye phantom. ***TOP LEFT:*** *3D CAD rendered view of eye phantom.* ***RIGHT:*** *3D CAD rendered half-sectional view of eye phantom with colour coding to allow identification of the different components:* ***cyan*** *– 14 D trial lens,* ***blue*** *– mount for lens, iris and LEDs,* ***grey*** *– M3 bolt,* ***magenta*** *– iris with 8 mm pupil,* ***red*** *– LEDs,* ***lavender*** *– camera macro-bellows,* ***green*** *– mount for ONH targets,* ***purple*** *– white card,* ***orange*** *– ONH target.* ***BOTTOM LEFT:*** *photograph of eye phantom.*

In the following description, the colour-coded half-sectional view is referred to in order to point out the various components of the phantom. It featured: a single 14 D bi-convex trial lens (BiB Ophthalmic Instruments, Stevenage, Hertfordshire, UK), shown in cyan, to provide refractive power; an air-filled interior; a 3D printed iris with an 8 mm pupil, shown in magenta; interchangeable 3D printed ONH targets, where one is shown in orange; and two 2 mm flat top white light-emitting diodes (LEDs; L-13PWC-Z, Kingbright, Issum, Germany), which are shown in red. These components were all held in place using 3D printed mounts, shown in green and blue, fitted to the back and front of a camera macro-bellows (Fotga, Shenzhen, China), which is shown in lavender.

5.2.2 Optical Design

5.2.2.1 Geometrical Optics

The optical configuration of the eye phantom is depicted in Figure 35. It shows the 14 D lens, iris with 8 mm pupil and an ONH target. It was intended to mimic only an emmetropic eye (i.e., one that is infinity focused) to simplify matters at this stage. The phantom was designed to be air-filled so that the issues associated with fluid-filled phantoms (i.e., leakages) could be avoided in this early work. The scale factor of the phantom was chosen such that the ONH targets would be sufficiently large such that they could be easily fabricated using a Fused Deposition Modelling (FDM) 3D printer, since manufacturing life-sized targets requires a more precise manufacturing technique and creating texture on targets of that scale is difficult. Moreover, it was chosen such that a calibration target would be able to be positioned in the full field of view of the imaging system and not be blocked by the rails of the bellows, which could have been the case if the calibration target needed to be a size

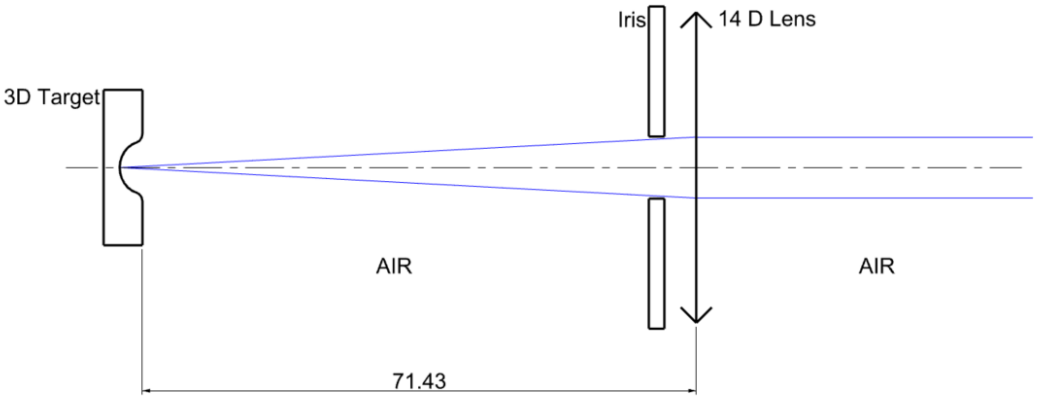


Figure 35: Diagram showing the optical configuration of the eye phantom. Dimensions are in millimetres.

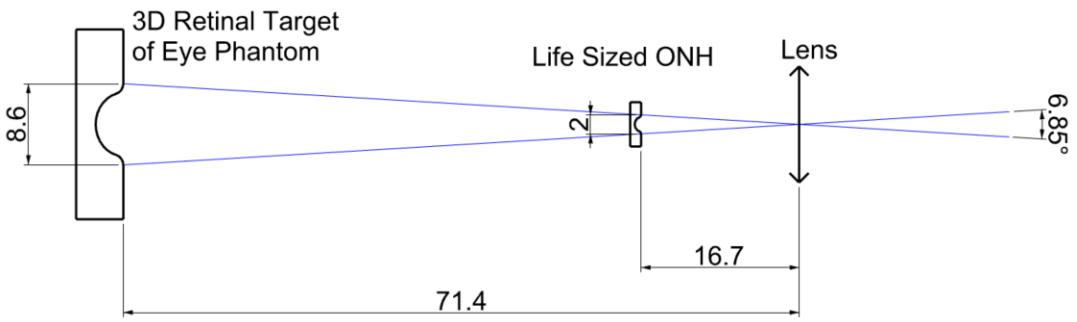


Figure 36: Diagram showing how the ONH targets of the eye phantom compare to the ONH within a human eye in terms of angular size. Dimensions are in millimetres.

which was too large. Regarding ONH target size, the dimensions of the features on them match that which is seen through the slit lamp in a life-sized eye, due to an appropriate size matching. As can be seen in Figure 36, the angular size of the structures of the ONH target matches that of a 2 mm ONH structure in a human eye, which typically has a focal length of approximately 16.7 mm (Atchison & Smith, 2000). The ONH targets of this phantom required scaling by a factor of 4.3 to achieve the correct angular size.

5.2.2.2 Aberrations

At this early stage, the optics of the eye phantom were chosen to allow for work to begin as soon as possible, provide refractive power but not mimic the more complex optical characteristics of the human eye, such as spherical aberration, coma, astigmatism, and chromatic aberration. Moreover, since the angular size of the structures being imaged is small, the field is likely to be approximately flat over size of the structure, and therefore field curvature was not considered. Distortion was also not considered at this stage, since the amount of distortion over the field of view of the structures, which have a small angular size, is likely not very large but more importantly, it can be removed using the results of a calibration process.

5.2.3 Mechanical Design and Fabrication

5.2.3.1 Design Software and 3D Printers Used

In all of the work within this thesis, all 3D CAD designs were generated using the CAD software Rhinoceros 6 (McNeel Europe, Barcelona, Spain). Moreover, all 3D CAD designs to be printed were processed for printing using the slicing software Ultimaker Cura (Ultimaker B.V., Utrecht, Netherlands).

In the work of this chapter, unless otherwise specified, 3D printed parts were printed using the Ultimaker 2+ 3D printer (Ultimaker B.V., Utrecht, Netherlands) using green polylactic acid (PLA; RS Components Ltd, Corby, Northamptonshire, UK).

5.2.3.2 Main Structure

As mentioned, a camera macro-bellows was used for the main body of the phantom. This item allowed for parts to be locked into position at its front and back openings, through bayonet attachments. The distance between the back and front openings could be adjusted, which was exploited to allow for focusing. The range required for focusing, on the order of a few millimetres, is very small compared to the focusing range possible with the bellows, which permitted the axial distance to be varied between around 40 mm to 150 mm. At the front of the bellows, the anterior part containing the lens, iris and white LEDs (for illumination of the ONH structures from the inside) was mounted. At the back, the posterior part which held interchangeable ONH targets was mounted.

The anterior part held the 14 D trial lens in place (this could be easily switched with another by releasing a mounting piece). The part also held the iris, which was 2 mm thick and had an 8 mm aperture, just behind the lens. The distance between the posterior surface of the lens and the iris was approximately 0.9 mm. The iris was held on a slide, that could be removed, the iris interchanged for one with a different aperture size and reinserted into the phantom. This part also held the two white LEDs, secured using Loctite 4305 UV Curable Adhesive (Loctite, Düsseldorf, Germany), pointing in the direction of the retina, for illumination. The posterior section clamped in place a square piece of white card, shown in purple in Figure 34, onto which the ONH targets were secured using Bostik Blu Tack (Bostik, Milwaukee, Wisconsin, US) – allowing the targets to be interchanged easily and rapidly.

5.2.3.3 ONH Targets

Four different ONH targets were created. They had a similar structure size to the human ONH but with different cross-sectional profiles. The cross-sectional profiles of these are shown in Figure 37. The 3D CAD designs for each were created by revolving their profile around a

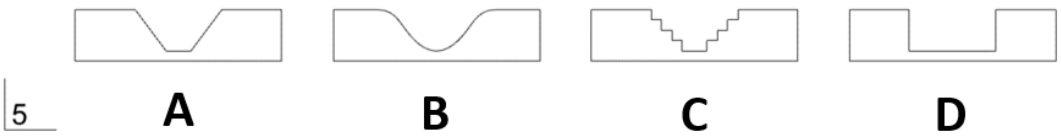


Figure 37: Cross-sections of the ONH targets.

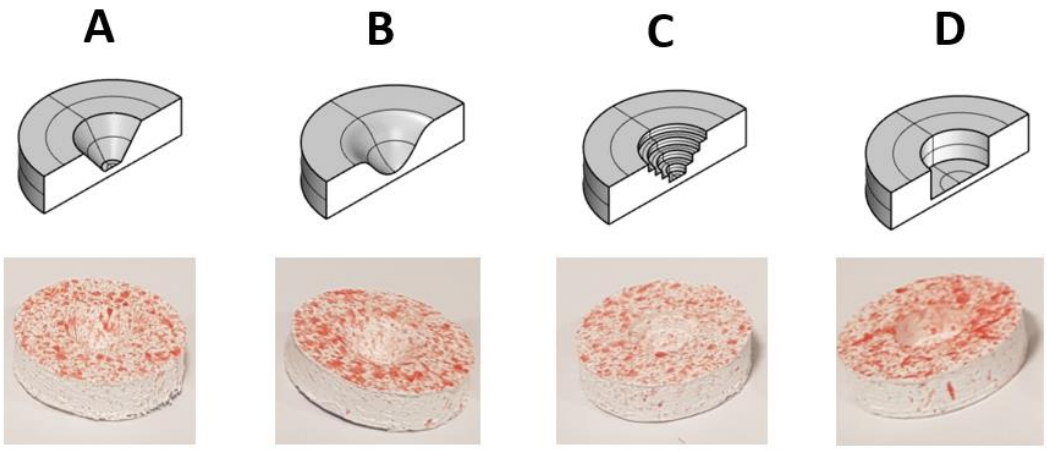


*Figure 38: Half sectional 3D CAD rendered views (**TOP**), and photographs (**BOTTOM**), of the ONH targets.*

vertical axis passing through their centre. Half-sectional 3D CAD rendered views of the targets are shown in Figure 38. Following scaling by the scale factor of the phantom, and 3D printing, the models were brush painted with titanium white heavy body acrylic paint (Golden Artist Colors, Inc., New Berlin, New York, US) and, after drying, painted with red acrylic paint (WHSmith, Swindon, Wiltshire, UK) using a brush flicking technique to create a speckled pattern. The pattern was added to provide texture to the stereo matching algorithm. Photographs of the fabricated targets are shown in Figure 38.

5.2.3.4 Calibration Target

In order to perform calibration of the imaging system, it was required to show to it a calibration pattern, held at the retinal location. The reason that it needed to be held at the retinal location is that the optics of the eye are essentially part of the imaging system. The calibration pattern used within this work was a checkerboard pattern, the required pattern of the calibration protocol used in this work. A photograph and exploded 3D CAD rendered half-sectional view of the calibration target, featuring the checkerboard pattern, is shown in Figure 39. In order to use the eye phantom for calibration, the posterior part needed to be removed and the bellows closed, to allow enough space for the calibration target to be held in many different positions at the retinal location. The 3D CAD rendered view in Figure 40 shows the eye phantom in its calibration configuration.

In order to have good calibration accuracy, the pattern had to be precisely made and flat. In order to meet this specification, a 6 x 5 checkerboard pattern, in the form of a patterned film photomask (JD Photo Data, Herts, UK), was laminated onto a standard glass microscope slide (with its length cut to approximately 26 mm, giving a size of 26 mm x 35 mm x 1 mm) using

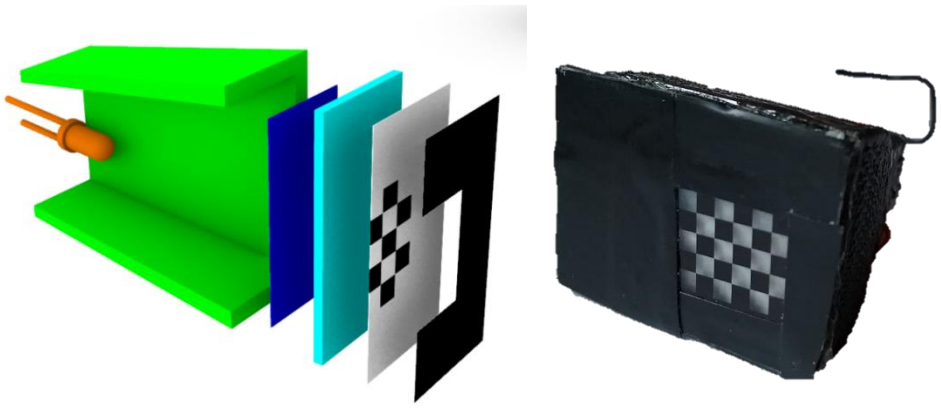


*Figure 39: Photograph (**RIGHT**) and exploded 3D CAD rendered half-sectional view (**LEFT**) of the calibration target. Colour coding for CAD rendered view: **black** – black electrical tape, **white** – photographic film featuring checkerboard pattern, **cyan** - microscope slide, **blue** – white plastic sheet, **green** – 3D printed LED mount, **orange** – white 3 mm through hole LED.*

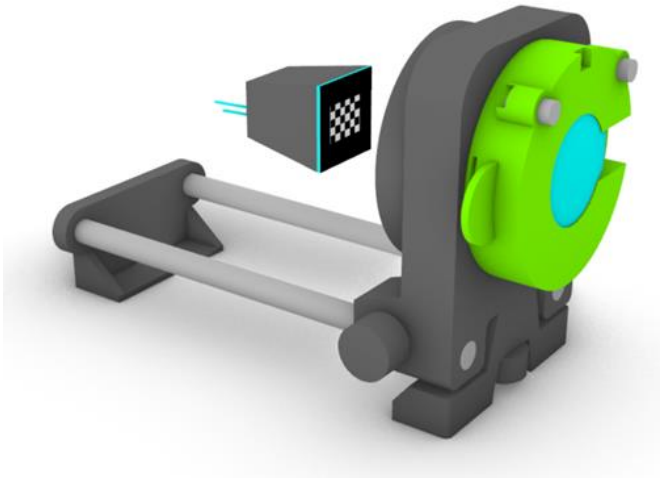


Figure 40: 3D CAD rendered view of eye phantom, set up for calibration.

Loctite 4305 UV Curable Adhesive. It was ensured that no bubbles were trapped between the film and the slide. It was determined, by measurement in a photograph of the length containing 6 squares, using the scale on a set of ABSOLUTE Digimatic Calipers (Mitutoyo Ltd., Andover, Hampshire, UK) as the photographic reference scale, and dividing by 6, that each square has an edge length of approximately 2.2 mm. Black electrical tape was applied on the surface of the film, around the checkerboard pattern, to create a border. The film containing the checkerboard pattern was in fact transparent and contained black squares only. A white backing was required in order to yield a black and white checkerboard pattern, and as such a thin white plastic film, cut from packaging, was taped to the other side of the microscope slide using black electrical tape. This sandwich of materials was glued, using Loctite 4305 UV Curable Adhesive onto a 3D printed mount (printed in black PLA+ filament [SUNLU, Los Angeles, California, US] on the Wanhao Duplicator i3 FDM 3D Printer [Wanhao, Jinhua,

China]) that, at the other end, held two white 3 mm through hole LEDs (Kingbright, Issum, Germany). The reason for the LEDs to be placed here was to retro-illuminate the calibration pattern. In doing so, it avoided specular reflections from the checkerboard film that would have interfered with the calibration process, which was previously seen to be an issue with frontal-illumination.

5.3 Slit Lamp System Development

5.3.1 Brief Overall Description

Photographs of the slit lamp, before and after components were fit to it, are provided in Figure 41. Also shown in this figure is a basic schematic of it with the components fitted. The assembled system was composed of a mount that essentially clamped onto the slit lamp to hold an indirect lens (also known as a slit lamp lens) in position in front of the slit lamp's objective lens, and two webcams, mounted at the eye pieces. The slit lamp used in this work was a Zeiss 30 SL-M (Carl Zeiss AG, Oberkochen, Germany).

5.3.2 Design and Fabrication

5.3.2.1 Mounting of the Indirect Lens

The slit lamp lens utilised in this set up was a 90 D SuperField non-contact slit lamp lens (Volk Optical, Mentor, OH, USA). It was chosen as it is one of the lenses which is commonly available within ophthalmology and optometry clinics (Sheehan and Goncharov, 2011). The mounts used to hold the lens in position can be seen in the middle image of Figure 41 as green 3D printed parts to the left of the image. The lens was press fit into a 3D printed ring part which was itself press fit into a 3D printed main mounting arm which was positioned at the left side of the slit lamp's main block. The main arm was bolted to a 3D printed part located on the right side of the slit lamp, effectively clamping it to the slit lamp. This lens mount held the slit lamp lens approximately 140 mm in front, and on the optical axis, of the slit lamp's objective lens. As the lens ring part was press fit and free to slide backwards and forwards, changes in this distance could be made, allowing focusing.

5.3.2.1 Mounting of the Webcams

Two 720p webcams (B07Q2T23CR, Amazon .com, Inc, Seattle, Washington, US) were used since they were low-cost and straightforward to control with a PC. One was mounted at each

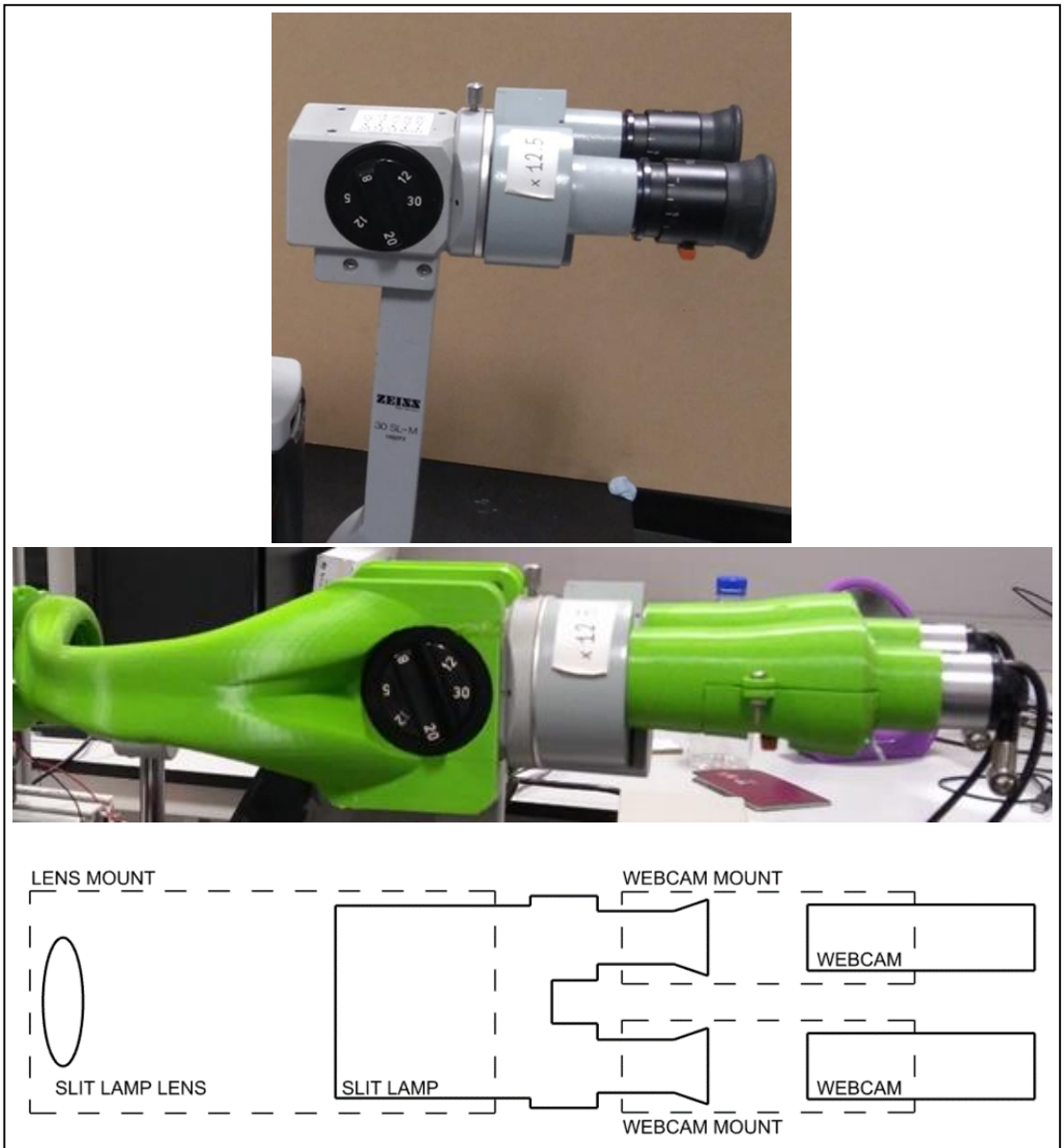


*Figure 41: Photograph of the slit lamp used in this work with (**MIDDLE**) and without (**TOP**) the add-on components mounted, and a schematic of it with the add-on components mounted (**BOTTOM**).*

eyepiece using 3D printed mounts. The mounts can be seen in the middle image of Figure 41, in green at the right of the image. The cameras were press fit into the mounts and the mounts were press fit onto the slightly tapered section of the slit lamp where the eye pieces insert into. The cameras were infinity focused – necessary as the focus of the eye pieces were set to their infinity (or zero-dioptre) setting. The cameras were connected to a laptop running Microsoft Windows 10 (Microsoft Corp, Redmond, Washington, US) and a script was written in C++, using functions of the OpenCV library (Bradski & Kaehler, 2008), to control image capture.

5.4 Performance Evaluation

5.4.1 Brief Overall Description

The slit lamp with fitted add-on components was used to capture the required images of the ONH targets and calibration target in the eye phantom. One stereo image pair was acquired of each target. The images were then processed using a computer stereo vision algorithm pipeline to yield disparity maps.

5.4.2 Trial Setup and Image Capture

The magnification setting on the slit lamp was set to 30 and fixed at this for the entirety of the trial. The focus of the cameras and the dioptre setting of the eye pieces were unchanged from the original setup – the cameras were already focused to infinity and the eye pieces at their zero-dioptre setting. The position of the slit lamp lens was adjusted such that the whole system was focused to infinity – allowing imaging of an emmetropic eye. The eye phantom was rigidly mounted in the appropriate position in front of the slit lamp lens, using a frame constructed of aluminium profile struts, such that it could be imaged. This setup is shown in Figure 42.

One of the ONH targets was placed on the retina of the phantom, and on its optical axis. The LEDs that are inside the phantom that provide illumination to the targets from inside, were connected to a power supply and powered. The axial length of the phantom was then varied, by adjusting the length of the bellows, until the targets surface came into focus in the images coming from the system's cameras, and that position set using the locking screw. The room

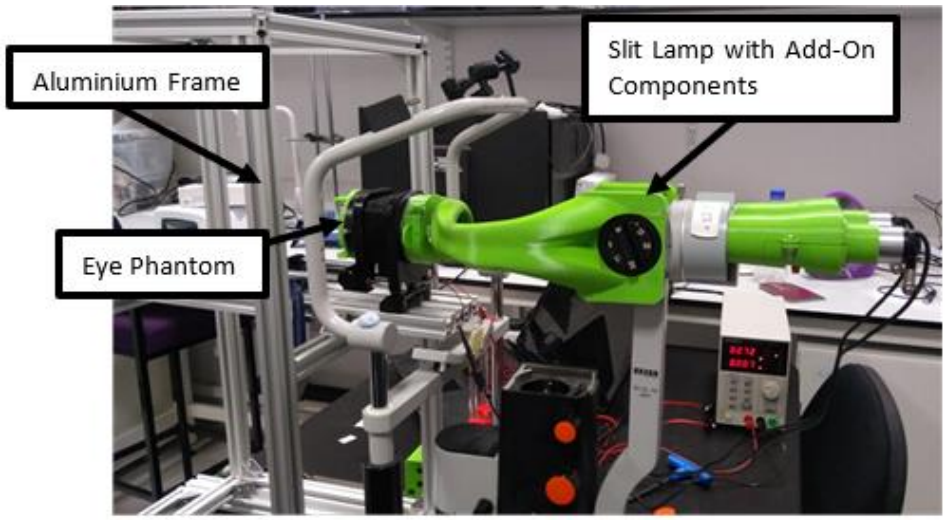


Figure 42: Photograph of the equipment setup for the performance trial.

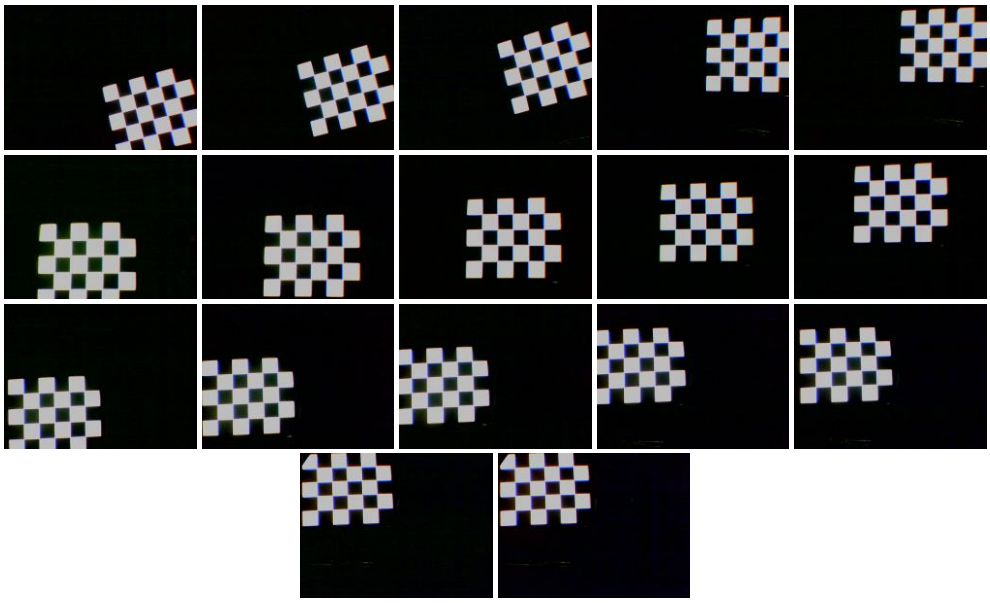


Figure 43: Calibration images taken with the left webcam.

was completely darkened before imaging began. Stereo images were then acquired of the target already inside the phantom and then of the others after interchanging them. Following this, the images necessary for calibration of the overall system (imaging system's plus the phantom's optics) were acquired. In order to do so, first the retinal target mount was removed from the back of the phantom, the bellows closed, and the LEDs of the phantom switched off. The calibration target was held in place using a clamp stand, to prevent movement during imaging. The LEDs of the calibration target were connected to a power supply to power them. The calibration target was then positioned in many different positions within the field of view of both cameras, while being held practically flat to them, and stereo images taken at each position. The target was held flat to the cameras since it was aimed to only use the distortion coefficients, which were thought to not require images of a tilted pattern to be estimated. In total, 17 calibration images were acquired. Those taken with the left webcam are shown in Figure 43. The images did not need to be cropped as the field of view was small and the images contained only the retina of the eye phantom and no parts/apertures of the system.

5.4.3 Stereo Vision Algorithm Pipeline

5.4.3.1 Overall Description of Pipeline

The computer stereo vision algorithm pipeline used in this trial was written in MATLAB (Mathworks, Natick, Massachusetts, US) on a laptop running Microsoft Windows 10. It involved the following steps:

1. Distortion removal. [Subsubsection 5.4.3.2].
2. Rectification. [Subsubsection 5.4.3.3].
3. Stereo Matching. [Subsubsection 5.4.3.4].

The MATLAB script written to perform rectification and stereo matching is provided in appendix 5A.

5.4.3.2 Distortion Removal

Using the acquired calibration images, the distortion coefficients of the system were determined by using the MATLAB Stereo Camera Calibrator App. The app is able to automatically detect the corners within the checkerboard pattern in the images and determine the distortion coefficients. In this work, the two tangential and three radial distortion coefficients, p_1, p_2, k_1, k_2 and k_3, which are detailed in subsection 2.3.3, were determined. The stereo images of the ONH targets were then undistorted using the determined distortion coefficients.

The overall mean reprojection error, a useful metric for evaluating calibration performance, which is calculated automatically by the app, was also recorded. Reprojection error is defined as the distance between where a point in the scene (a corner of the checkerboard) should have projected onto the image sensor, if the projection was perfectly rectilinear, and where it is in the undistorted image. The equation which is used to determine the overall mean reprojection error is as follows:

$$E = \frac{1}{NP}\sum_{i=1}^{N}\sum_{j=1}^{P}\sqrt{\left(X_{e_{i,j}}\right)^2 + \left(Y_{e_{i,j}}\right)^2} \qquad \textit{Eq. 5.1}$$

where E is the overall mean reprojection error, N the combined number of left and right calibration images taken, P is the number of points in the checkerboard pattern, $X_{e_{i,j}}$ is the x component of the reprojection error for point j in calibration image i, and $Y_{e_{i,j}}$ is the y component of the reprojection error for point j in calibration image i.

5.4.3.3 Rectification

All further mention of images refers to the images of the ONH targets. Prior to rectification, the images were converted to grayscale images. The transformation which needed to be applied to the images in order to rectify them was determined from the fundamental matrix. The fundamental matrix is a 3 by 3 matrix which encodes the epipolar geometry in an

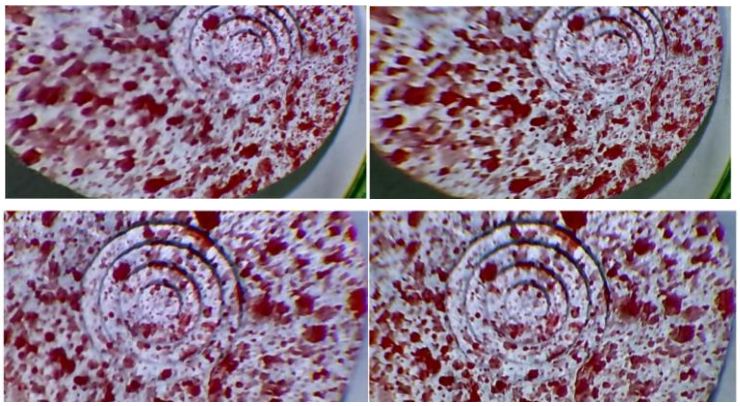


*Figure 44: Example of rectification result where resulting images have unwanted distortions and cropping (**TOP**) and where the images do not (**BOTTOM**).*

uncalibrated situation. It was determined using the normalised 8-point algorithm (Hartley & Zisserman, 2003) with the Random Sample Consensus (or RANSAC) Method (Fischler & Bolles, 1981) to remove outliers, using point correspondences found between the images in stereo pairs. In order to find point correspondences, the speeded up robust features (SURF) detector (Bay et al., 2006) was first used to find interest points which were then matched across the images in the stereo pairs using the sum of absolute differences metric on feature vectors determined for each interest point. The fundamental matrix and corresponding transformations, needed to rectify stereo image pairs, were determined for each stereo image pair, then applied to rectify them. In some cases, the rectification process yielded images which were rectified in such a way that erratic distortions were present, and the important areas of the images were cropped out to some degree. Figure 44 shows two different results of the rectification process on one of the acquired stereo image pairs: one where the rectification process introduced such unwanted features and the other where a more reasonable result was obtained. In these instances, and in cases where the algorithm failed to determine the fundamental matrix, the rectification procedure was repeated, until a reasonable rectification result was achieved (i.e., one in which the structures of interest were close to the centre of each image and obscure distortions, such as that present in the images in the top row of Figure 44, were not present).

5.4.3.4 Stereo Matching

Stereo matching was performed using semi-global block matching (Hirschmuller, 2005) with a support window size of 17. A circle crop was applied to the obtained reconstructions to keep only the structure of interest in the centre of each target plus some of the flat surface surrounding it.

5.5 Results

5.5.1 Calibration Performance

The overall mean reprojection error for the calibration of the cameras in the stereo system was 0.6 pixels.

5.5.2 Acquired Images of the ONH Targets

The stereo images acquired of the ONH targets are shown in Figure 45.

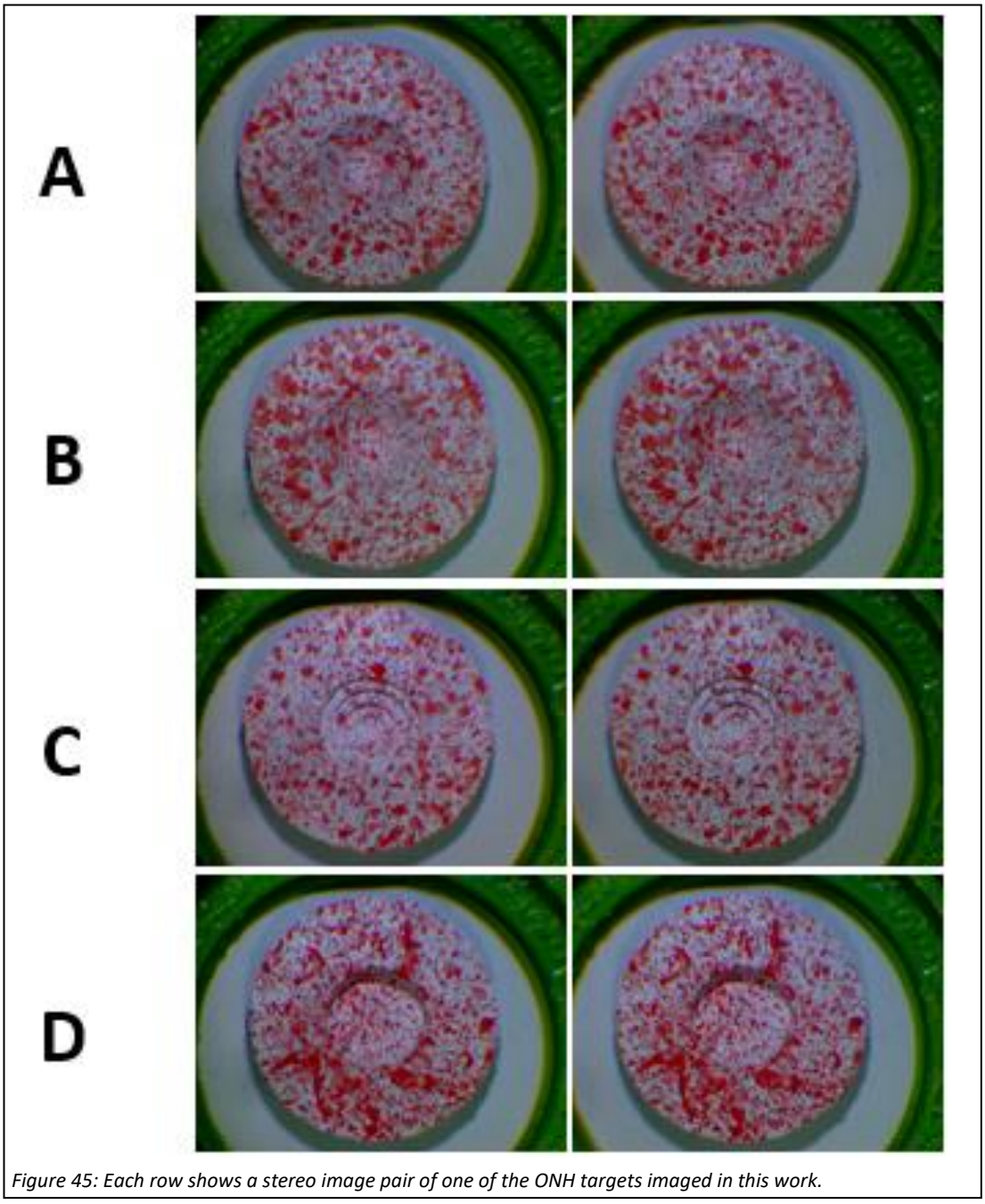


Figure 45: Each row shows a stereo image pair of one of the ONH targets imaged in this work.

5.5.3 Disparity Maps Obtained

Colour heightmap textured disparity maps obtained of each ONH target are provided in Figure 46. The figure also includes a cross-sectional profile, through the centre, of each reconstruction and its original CAD design (for ease of comparison).

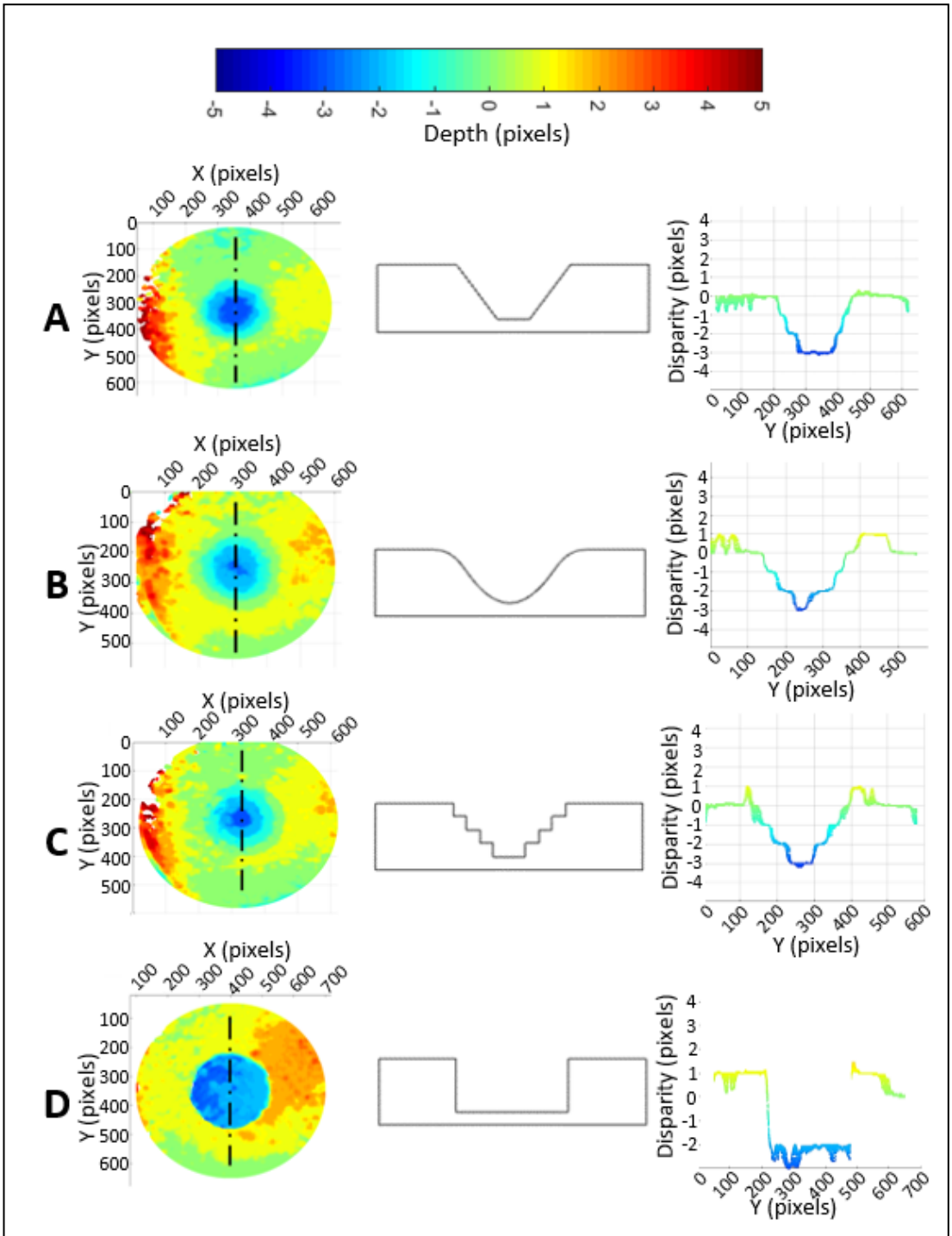


*Figure 46: Each row shows a colour heightmap textured disparity map of a particular target (**LEFT**), and a cross-sectional profile of its reconstruction (**RIGHT**) and CAD design (**MIDDLE**), taken through its centre.*

5.6 Discussion and Conclusion

5.6.1 Type of Validation Performed

Only disparity maps were obtained at this stage. Disparity maps are metrically uncalibrated and are also not correct in terms of shape up to a scale factor since they have not been reprojected. However, they roughly show the shape of the reconstruction that would be obtained after reprojection. Thus, the reconstructions could only be assessed roughly in terms of shape but not accuracy. Precision was also not assessed since the rectification process lacked repeatability.

5.6.2 Disparity Map Shape Validation

In comparing the cross-sections of the obtained disparity maps with the CAD designs, it can be seen that the general shape of the structures was approximately recovered in each case. In looking at the 2D disparity maps, it appears that in most cases there are raised sections to the right and left of the key structures at the centres of the targets. This pattern is not seen at the top and bottom of them.

The reconstructions appear to be affected by a quantisation-like effect. This is particularly evident when looking at the cross-sections obtained for cases A and B. Their sloped regions, which are supposed to be straight/curved, show this. In light of this, it is not clear as to whether the profile of target C was correctly recovered, since it is not possible to parse out whether the individual steps were actually detected or if they are in the disparity map as a result of this quantisation effect.

The disparity maps appear smooth and noise free in many locations, but some stochastic noise is present in others, particularly in the peripheral regions away from the structures of interest. It can be seen, in the cross-sectional slices, that little stochastic noise is present at the location of the structures of interest for targets A to C, but for target D, the stochastic noise is quite significant.

The observed quantisation-like effect looks to be the most significant error present. Stochastic noise is also an issue, but to a lesser extent.

5.6.3 Potential Sources of Error

5.6.3.1 Stereo Matching and Image Resolution

The stereo matching step is likely responsible for the majority of the errors seen. Stereo matching is a challenging problem due to aspects such as textureless areas, radiometric variations, nonplanar surfaces, and perspective distortions. Since the process relies on detecting very subtle differences between the images, no more than around 3 pixels in this case, small errors can result in large reconstruction errors.

The matching algorithm used here, the semi-global block matching algorithm, appeared to be able to obtain relatively smooth disparity maps with only some regions affected by noise. However, it appeared to result in a quantisation-like effect where the maps showed plateaued regions at integer disparity values, which strongly indicates that the error caused by the effect could be reduced by increasing image resolution so that there would be a larger disparity range in the disparity map. Norouzifard and colleagues (Norouzifard et al., 2018) also used the semi-global block matching algorithm in their work but the reconstruction they obtained did not appear to contain correct depth information and was not validated in any way. Others, using tailored algorithms, as was explored in the scoping review in Chapter 3, appeared to achieve better results, of which some were validated against ground truth data. It would therefore likely be beneficial to follow suit and implement a tailored algorithm, specific for this situation. One key advantage of doing so is that it permits more control over the algorithm and greater potential for improvements to be made, which is very difficult to do with MATLAB's built-in semi-global block matching function. It would still also likely be beneficial to use cameras with a higher resolution.

5.6.3.2 Distortion Removal

Calibration is relied upon for distortion removal. Poor calibration is very likely to have a negative impact on the accuracy of the reconstructions obtained. There are two aspects of the calibration that could have contributed to errors: how well the system can be described by the models that the calibration fits them to, and how accurately the calibration is conducted. The model which the calibration attempts to mathematically fit the system to is appropriate for conventional cameras in a stereo setup. Conventional cameras contain lens elements on a single optical axis, and their distortion can usually be accurately modelled with radial and tangential distortion terms. In this system however, where the lens elements for each of the cameras, as can be seen in the illustration provided in Figure 47, are not on a

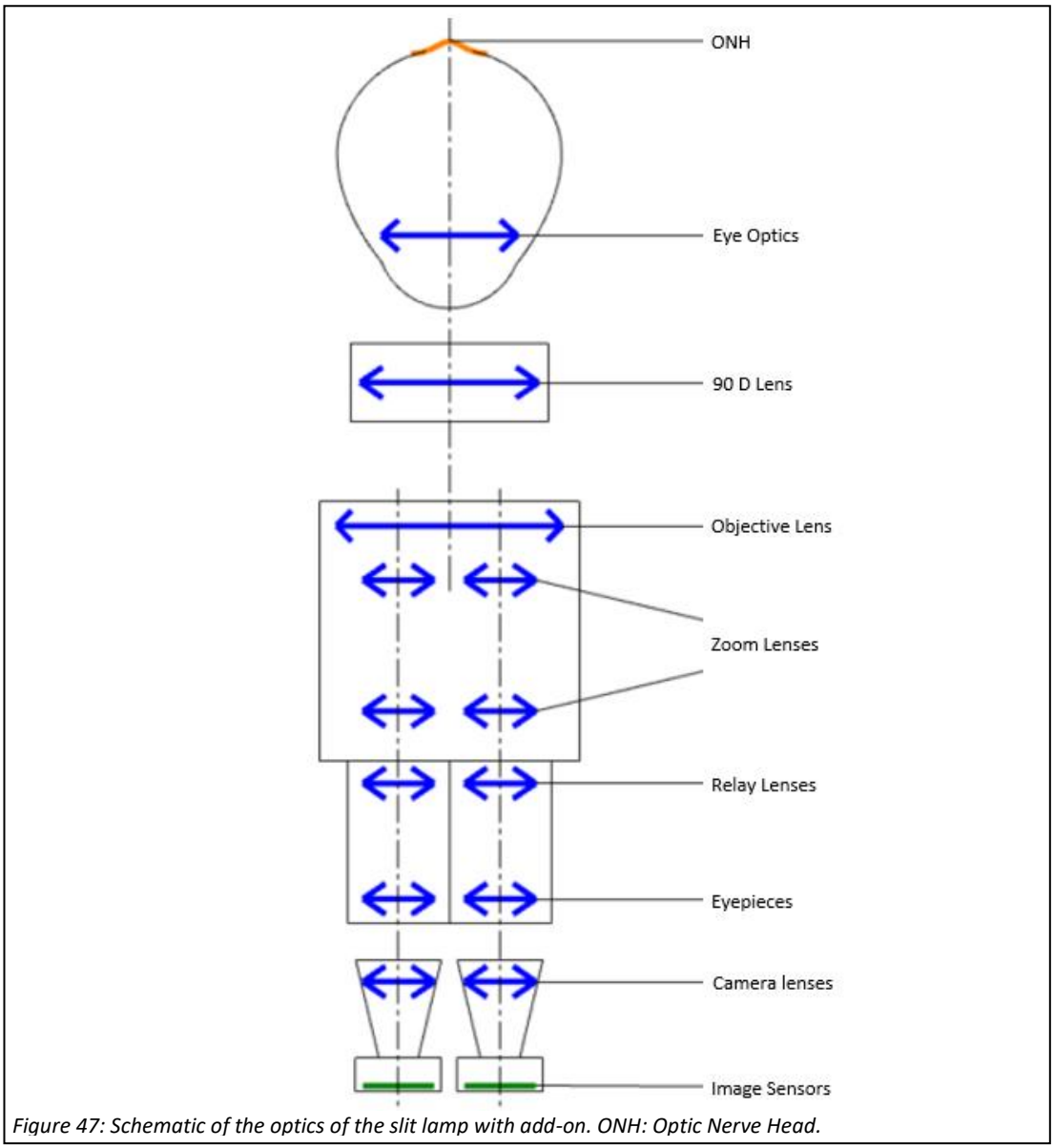


Figure 47: Schematic of the optics of the slit lamp with add-on. ONH: Optic Nerve Head.

common optical axis, it is not entirely clear how well only radial and tangential distortion can describe the distortion within this system. The mean reprojection error seen in the calibration can give some indication as to how well these distortion models can characterise this system, but it is also affected by how well the calibration was carried out. In this trial, the error value determined, 0.6 pixels, exceed the range of values which others have said to be typical for stereo camera systems (Vasudevan et al., 2011): in the range of 0.12 to 0.2 pixels. However, the mean reprojection error is affected by the pixel size, since for example a 0.2-pixel error for a low-resolution sensor would represent a greater distance error on the sensor than a 0.2-pixel error for a high-resolution sensor of the same size. Therefore, comparison to the values stated by others is not totally useful.

The calibration errors do however seem large when compared to the maximum range of disparity values seen within the ONH targets' structures in this work (approximately 4 pixels). It is therefore likely that error in distortion removal had some impact on reconstruction accuracy. It must be noted that for radial distortion, since it increases with distance from a camera's principal point, errors would likely have had a greater effect on the peripheral locations of the reconstructions, which may explain the noise seen in these locations. This fact, together with the fact that the lens elements of the system were not all-in alignment in the horizontal direction but were in the vertical direction, could explain the raised and noisy regions at the left and right peripheral locations of the disparity maps.

In light of this, future work to investigate the appropriateness of the distortion models for this system and how a more accurate calibration could be performed would likely be beneficial to this work.

5.6.3.3 Rectification

Rectification errors could have contributed to the errors seen, in particular the noise. The level of the rectification performance is not entirely clear and even how to assess its performance is not clear. However, if there were errors which led to object points not falling on the same horizontal scanlines in both images then this would have undoubtedly caused errors during the stereo matching process, leading to noise. Future work would benefit from considering and assessing the rectification performance. Moreover, an approach where rectification is carried out using the calibration parameters determined for the system would likely lead to more stable results.

5.6.3.4 Image Focus

Poor image quality likely contributed to the irregularities seen in the disparity maps. It can be seen in all of the stereo image pairs within Figure 45 that there are regions of the targets where the focus differs between the left and right images. The stereo images obtained of target A, where boxes have been added to indicate some regions where the focus differs between the left and right images, are shown in Figure 48. It can be seen with the red boxes that in this region the focus is better in the left image while the region indicated by the yellow boxes is in better focus in the right image. Obtaining images which are both in good focus, in all regions of the structure being imaged, would almost certainly improve stereo matching performance and therefore the results produced. Determination of the depth of focus would provide vital information that could help in making such improvements.

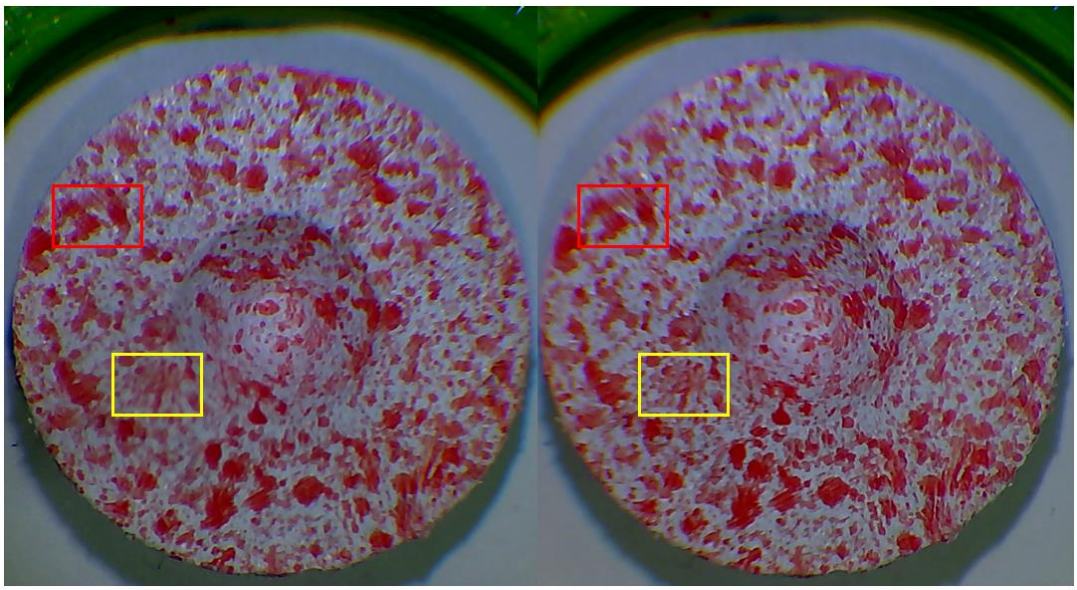


Figure 48: Stereo image pair of target A with boxes to indicate example regions where the focus differs between the two images.

5.6.4 Limitations of This Work

5.6.4.1 Type of Validation Performed

Since only disparity maps could be obtained at this stage, analysis of the system and algorithm's accuracy was not possible. Further development is required to permit calibrated reconstructions to be obtained so that they can be compared against ground truth data. Further work to assess precision of the disparity maps is not warranted since the rectification process causes very different results to be obtained each time with the same input images. A calibrated, and more stable, method is needed before precision can be usefully evaluated.

5.6.4.2 Surface Texture of ONH Targets

The ONH targets created in this work had a texture density which is much higher than that which is seen for the human ONH, which is practically textureless in many regions. This likely would have made the stereo matching algorithm perform much better than it would on a human ONH, meaning that this performance trial would have overestimated the performance that this system would have in vivo. Future work would benefit from using targets which have a texture density which is closer to that of the human ONH. This would pose challenges for the stereo matching algorithm and require some changes to enable it to operate successfully.

5.6.4.3 Optical Properties of ONH Targets

The superficial layers of the human ONH are semi-transparent, whereas the surfaces of the ONH targets used here were opaque. This aspect would have also likely led to an overestimation of the performance that this system would have in vivo. It would therefore likely be useful in future to create ONH targets which are semi-transparent. This would allow for performance to be more accurately characterised. It must be noted here, for future consideration, that optimal selection of illumination wavelength, corresponding to green for example, would likely allow for better imaging of the superficial layers than with longer wavelengths. Shorter wavelengths, corresponding to blue, highlight the superficial layers more strongly than green but is more toxic to the retina and RGB image sensors are less sensitive to it than green. The use of a single wavelength would also eliminate issues with chromatic aberrations.

5.6.4.4 Shape of ONH Targets

The shape of the ONH targets used, apart from target B, are not that similar to the human ONH. While these provided additional information on the performance of the stereo matching algorithm, it may be more useful to image only structures with a similar shape to the human ONH in future.

5.6.4.5 Illumination

Since the ONH targets were illuminated from the inside, it is likely that the quality of the images is better than that which would have been obtained if illumination was provided externally, as is the case in reality when imaging in vivo. It is likely that, when illuminating external to the eye, that corneal reflections would present difficulties. It is also not clear if the same light levels would be possible in vivo due to illumination safety limits.

5.6.4.6 Eye Phantom Optics

The eye phantom used in this trial likely did not accurately mimic the aberrations, and therefore optical performance, of the human eye and its reflection artefacts. It did mimic an emmetropic eye in terms of refractive properties – it allowed for the infinity focused system to focus onto the ONH targets. It was intended to create an eye phantom of this complexity at this stage since it allowed for system development to begin as soon as possible, since its construction was very straightforward. Future work would benefit from testing in an eye phantom that is more optically similar to the human eye. In particular, corneal curvature to

better mimic reflection artefacts. Reflection artefacts is something that will not have had an effect on the results of this work since the ONH targets were illuminated from inside the phantom meaning that no corneal reflections existed but will present challenges when illumination is provided externally. A further limitation related to the eye phantom is that the ONH targets were held on the phantom's optical axis, different to the situation inside the human ONH where it is off axis.

The amount of distortion in the human eye, over the angular size of the ONH, has not been considered yet but needs to be since if it is not negligible it will likely cause errors when imaging in vivo since the system will need to be calibrated on an eye phantom, on which the distortion coefficients estimated will depend. In the eye phantom work this is not an issue since the calibration takes place on the eye phantom that is also being using for ONH imaging, so distortion can be fully removed provided an accurate calibration is achieved. Future consideration of distortion, since it can affect accuracy in computer stereo vision (Ramírez-Hernández et al., 2020), would likely be useful.

5.6.4.7 Image Capture Synchronicity

It was not possible to obtain stereo images simultaneously using the USB webcams. While this did not affect the result in this trial since the eye was stationary, it will almost certainly present problems for in vivo work since the human eye will not be stationary, due to head and eye movements, such as microsaccades. Quantitative stereo vision requires simultaneous capture for moving targets. Future work will need to be done to implement cameras whereby simultaneous capture is possible.

5.6.5 CONCLUSION

The target of this work was to assess the initial feasibility of the idea of using a slit lamp, fit with some low-cost components, for 3D surface reconstruction of the ONH, to make sure that there was not a fundamental flaw in it. Thus, a simple system was set up, tested on a simple eye phantom, to see if results could be obtained (i.e., could retinal structures be detected and reconstructed in 3D).

This work showed that with this system it was possible to detect the retinal structures of the eye phantom, which warranted further work on this idea.

The issues (sources of error, limitations, etc.) seen/identified in this work are tabulated in Table 4.

Table 4: Issues present at this stage (chapter 5).

Issue Number	Issue
1	Stereo matching technique resulted in quantisation effect
2	Low image resolution
3	Calibration performance sub-optimal
4	Unstable rectification results
5	Poor focus in some areas of the images
6	Validation was only qualitative since reconstructions are uncalibrated
7	ONH target surface texture different to human ONH
8	ONH target optical properties different to human ONH
9	ONH target shape different to human ONH
10	Illumination not provided from outside the eye
11	Eye phantom optics different to human eye
12	Image capture is not synchronous

CHAPTER 6

SLIT LAMP SYSTEM IMPROVEMENTS AND CALIBRATED RECONSTRUCTION IN AN EYE PHANTOM

6.1 Introduction

The work of this chapter built on the work of the previous chapter through further development and testing. This involved making changes to the slit lamp add-on components, the stereo vision algorithms, eye phantom and testing methods, to address some of the issues that arose in the previous chapter (as are tabulated in Table 4). The core difference in this work was that the system was now capable of performing calibrated reconstruction, to obtain reconstructions in real-word coordinates, which permitted quantitative validation to take place (addressing issue 6: validation was only qualitative since reconstructions are uncalibrated). In order for this to be possible, the stereo vision algorithms were completely changed. This included the development of a tailored coarse-to-fine stereo matching algorithm (addressing issue 1: stereo matching technique resulted in quantisation effect) and the use of a rectification function which used calibration parameters, leading to repeatable results (addressing issue 4: unstable rectification results).

Other changes which were made in this work were the upgrading of webcams to higher resolution cameras (addressing issue 2: low image resolution) and the creation of ONH targets with shapes more similar to the human ONH (addressing issue 9: ONH target shape different to human ONH).

6.2 Eye Phantom Upgrades

6.2.1 Brief Overall Description

The eye phantom used in this work was the simple phantom which was described in the previous chapter (section 5.2), with the main difference being that new ONH targets were created. The calibration target used here was also different to the one that was used previously. It featured more squares in its pattern and was used instead of the previous one to increase the number of data points associated with each calibration image.

6.2.2 Design and Fabrication of the ONH Targets

6.2.2.1 Design

Two different ONH targets, representing different ONH morphologies, were created – one resembling an ONH without pathology and the other resembling an ONH with significant glaucomatous damage. Photographs of these, together with their corresponding 3D CAD

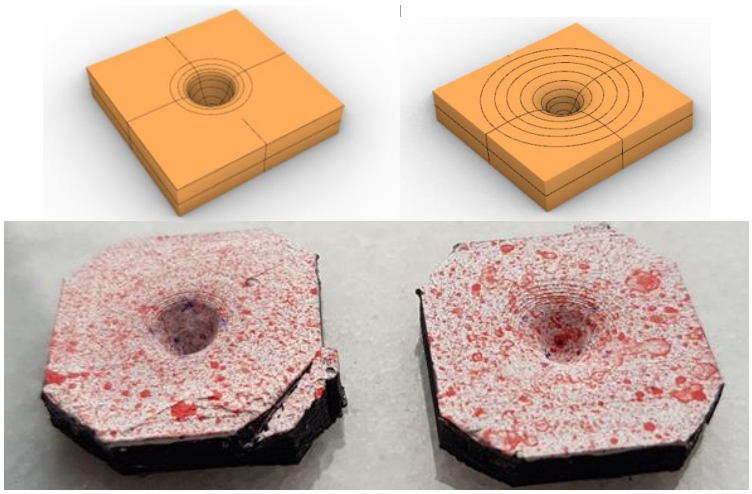


*Figure 49: Photographs of the ONH targets created (**BOTTOM ROW**) and their corresponding 3D CAD models (**TOP ROW**). **LEFT COLUMN**: ONH target representing ONH with significant glaucomatous damage. **RIGHT COLUMN**: ONH target representing ONH without pathology.*

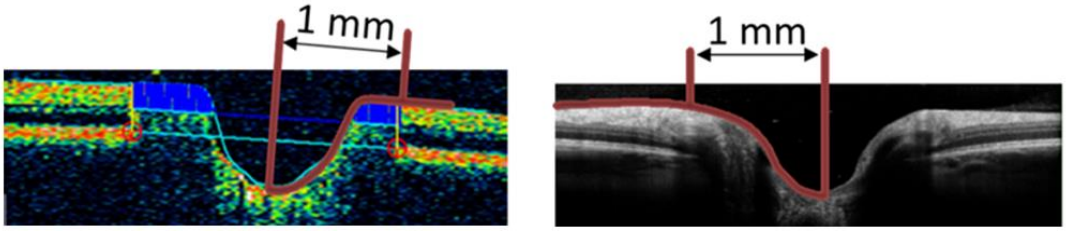


*Figure 50: 2D OCT slice images, with the ILM profiles traced (overlaid in red), used in the design of the ONH targets in the case of no pathology (**RIGHT**; Heidelberg Engineering, n.d.) and with significant pathology (**LEFT**; Agrawal et al., 2016).*

models, are shown in Figure 49. It must be noted that in order to fit at the back of the eye phantom, the corners of the ONH targets needed to be cut, but this was far from the ONH structures at the centres of the models and was thus not an issue.

The process of creating the targets involved 3D CAD design, 3D printing and painting. The CAD design of the normal and glaucomatous targets were based on OCT B-scans of a normal human ONH (Heidelberg Engineering, n.d.) and a phantom ONH (Agrawal et al., 2016), respectively, which are shown in Figure 50. The difference between these two structures is that the pathological ONH shows an increased excavation of the cup structure, compared to the normal ONH, due to thinning of the RNFL. Figure 51 illustrates the process used to develop the 3D CAD designs, demonstrated for the case of the ONH target without pathology. The first step in the design of the targets involved creating the dimple like morphologies. This was done by first tracing a section of the profile of the internal limiting membrane (ILM) in the relevant 2D OCT slice image (shown in red), and then revolving this profile, around a vertical axis passing through the deepest point of the ILM, to create a 3D structure. Before revolving, the profile traced was scaled uniformly in two dimensions by the amount

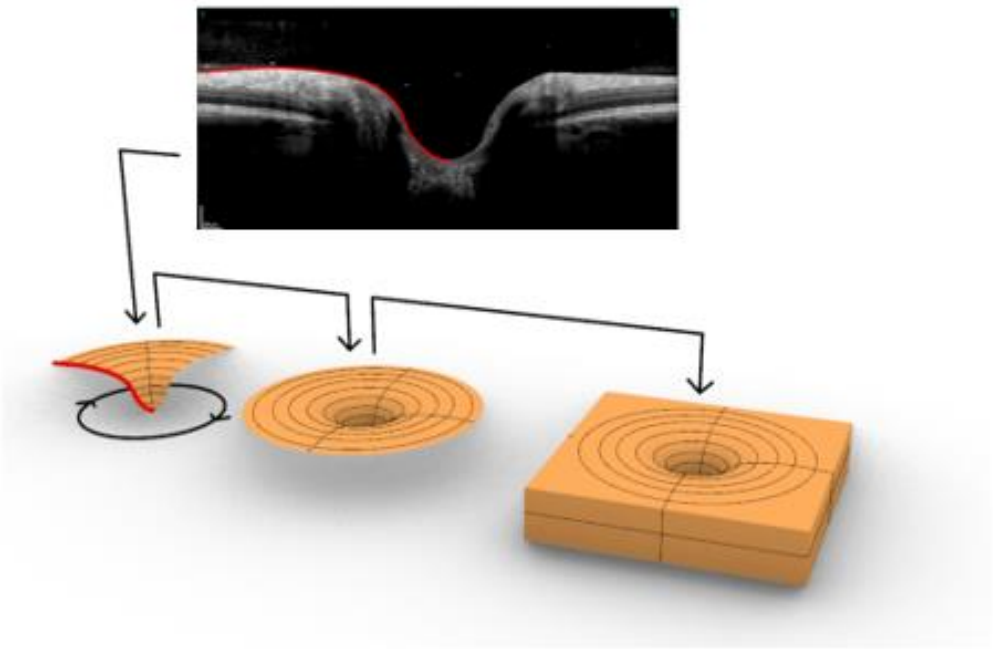


Figure 51: Illustration of the design process used to create the 3D CAD models of the ONH targets from 2D OCT slice images of the ONH.

necessary to give a disc size of 2 mm. The disc size was approximated as the distance between the two scleral ring edges in the OCT images.

The scaling resulted in a depth of 0.67 mm and 0.69 mm from the lowest point of the ONHs' cups to their flatter upper surfaces for the glaucomatous and normal cases, respectively. These depths were deemed reasonable when compared to HRT and OCT readings seen in the literature (Seymenoğlu et al., 2013). The revolved structure was then embedded into a square prism, and the whole part scaled up uniformly in three dimensions by a factor of 4.3 (the scale factor of this phantom).

6.2.2.2 Fabrication

Using the 3D CAD designs, the targets were printed with black PLA+ filament on the Wanhao Duplicator i3 3D printer. The targets were then painted in the same way as those detailed in the previous chapter (subsubsection 5.2.3.3): by painting with titanium white heavy body acrylic paint and, after drying, painting with red acrylic paint using a brush flicking technique to create a speckled pattern. In order to aid alignment of the reconstructions with ground truth data, 4 ink marks were made near the bottom of the dimple and 4 on the top surface, to act as anchor points for referencing the reconstructions to the ground truth data. It was aimed to place the marks 90° apart on the top surfaces at the approximate edges of the

dimple structures, and 90° apart on the bottom surfaces approximately where the structures flatten out at their bottom. These have been chosen to try to well align the important aspect of the ONH's structure when it comes to neuro-retinal rim thickness (i.e., the sloped sections).

6.2.3 Design and Fabrication of the Calibration Target

The calibration target created for this work is shown in Figure 52, in the form of a photograph and exploded 3D CAD rendered half-sectional view.

The target created in this work is much the same as the one used in the work detailed in the previous chapter (subsubsection 5.2.3.4), but features a different checkerboard pattern, 3D printed mount and light source. The checkerboard pattern used in this version had 10 x 7 squares and was created photographically on Rollei RPX 25 black and white film (Rollei, Hamburg, Germany) by photographing a Philips BDM4350UC 43" monitor (Philips, Amsterdam, Netherlands) with 4K resolution displaying the pattern, using a Yashica FX3+ camera (Yashica, Japan) with stock 50 mm lens, and then having the film developed by a commercial developing lab. The mount had a slightly different form and had only one 5 mm through hole white LED (NSPW515DS, RS Components Ltd, Corby, Northamptonshire, UK).

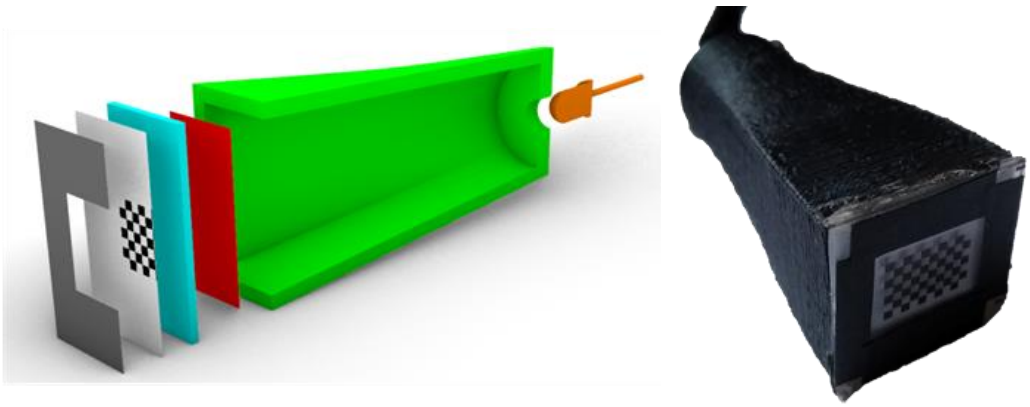


*Figure 52: Photograph (**RIGHT**) and exploded 3D CAD rendered half-sectional view (**LEFT**) of the calibration target. Colour coding for CAD rendered view: **grey** – black electrical tape, **white** – photographic film featuring checkerboard pattern, **cyan** - microscope slide, **red** – white plastic sheet, **green** – 3D printed LED mount, **orange** – white 3 mm through hole LED.*

6.3 Slit Lamp System Upgrades

A photograph, basic schematic and 3D CAD rendered view, with some parts sectioned to reveal internal components, of the add-ons, fitted to the slit lamp, are shown in Figure 53.

6.3.1 Design and Fabrication

6.3.1.1 Indirect Lens Mount

The indirect lens and mount used within this work were the same as was used in the work detailed in the previous chapter, apart from the fact that the mount was printed in black

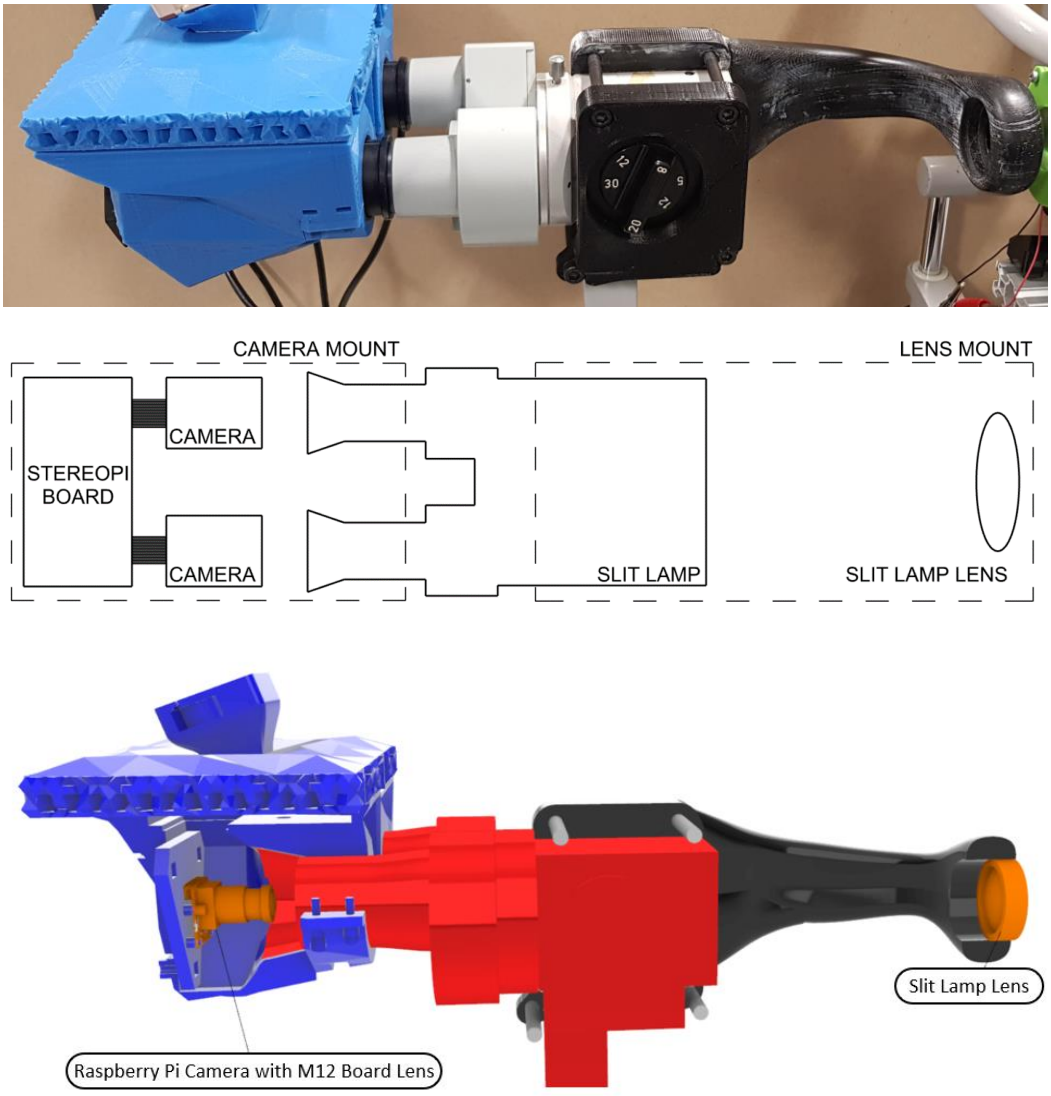


*Figure 53: Photograph (**TOP**) with corresponding schematic (**MIDDLE**) and colour-coded and labelled CAD rendered 3D model with some sections cut to reveal internal components (**BOTTOM**) of the slit lamp fitted with add-on components. **Blue** – 3D printed camera mount; **Red** – Zeiss 30 SL-M slit lamp; **Black** – Slit lamp lens mount; **Orange** – Imaging/optical components (excluding those of the slit lamp).*

PLA+ filament instead (on the Wanhao Duplicator i3 3D Printer). The reason for reprinting was simply that the previous version had become damaged.

6.3.1.2 Cameras

It was decided to use Raspberry Pi NoIR V2.1 Cameras (Raspberry Pi Foundation, Cambridge, UK), since they have a much higher resolution than the webcams which were used previously, at 8 megapixels, and permit the use of different lenses with them. The ability to select the lens allowed for the field of view to be narrowed, such that images contained even more data on the ONH. An image of one of the cameras used is shown in Figure 54.

In order to use M12 lenses with these cameras, some small modifications were required. A 3D CAD rendered view showing one of the cameras, modified to allow for an M12 lens to be used with it, is also shown in Figure 54. Modification involved first securing the cameras' sensor blocks to the printed circuit boards (PCBs) backing them to align the sensors better with the PCBs. In the stock camera, the block is attached to the PCB with a soft flexible disc material in between, leading to poor alignment with the PCB. This disc material was removed, and the blocks were secured to the PCBs directly using Loctite 4305 UV Curable Adhesive. Following this, the stock lenses were removed and M12 lens mounts, 3D printed in black PLA+ on the Wanhao Duplicator i3 3D printer, were glued onto the PCBs using the same glue, and M12 board lenses (Hongkong Ansice Industrial Co, Ltd, Hong Kong, China) with a focal length of 8 mm were screwed into place. The focal length of the M12 lens to be used was chosen such that the cameras would cover approximately a 15 mm vertical object height (corresponding to a field of view of approximately 12°) of the retina of the eye phantom, to sufficiently cover the structures of the ONH targets which were around 8.6 mm in diameter (which corresponds to an angular size of around 7°). The focal length was determined by first

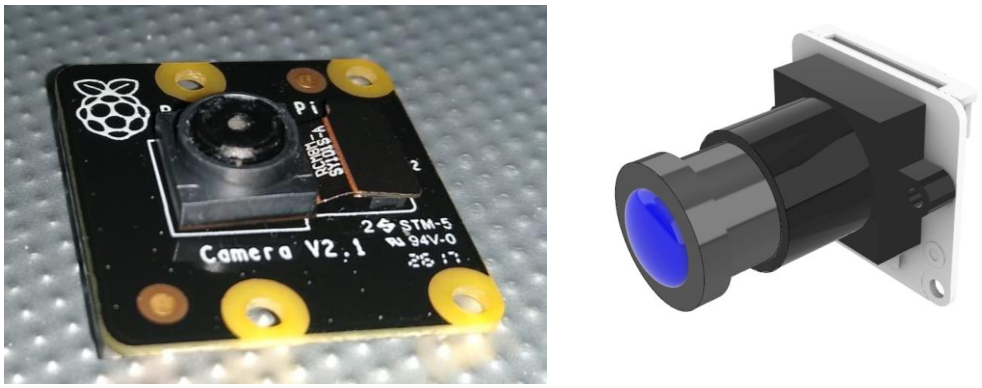


*Figure 54: Raspberry Pi Camera (**LEFT**) and CAD rendered view of a Raspberry Pi Camera with M12 lens, mounted using a 3D printed mount (**RIGHT**).*

imaging the ONH targets in the eye phantom with one of the cameras still containing its stock lens, through the system, then determining the scale up required from the obtained images, and therefore focal length required. Once the M12 lenses were fitted, their distance from the sensor was adjusted such that the cameras were infinity focused – necessary as the focus of the eye pieces were set to their infinity (or zero-dioptre) setting.

In order to control the cameras and collect images from them, they needed to be connected to a suitable Raspberry Pi Computer. In this case, a single StereoPi board (virt2real, St. Petersburg, Russia), containing a Raspberry Pi Compute Module 3 (Raspberry Pi Foundation, Cambridge, UK), was used since it allowed for both cameras to be connected to a single board. The Raspbian operating system (Raspberry Pi Foundation, Cambridge, UK) was installed. A script was created, in the Python programming language, to display the images coming from the cameras on a monitor connected using an HDMI cable and initiate stereo capture on a keyboard button press. The still images were captured at a resolution of 3280 x 2464 pixels for each camera. The cameras and StereoPi board were mounted to the eye pieces of the slit lamp using a mount that was 3D printed in blue PLA (RS Components Ltd, Corby, Northamptonshire, UK) on the Wanhao Duplicator i3 3D printer. The mount was designed to clamp onto the eye pieces, using clips, bolted on with M3 nuts and bolts, for tightening. The cameras were held on mounts that allowed for adjustment of the camera-to-eyepiece distance, allowing for the appropriate position to be achieved to give the maximum field of view, as well as some degree of tuning of their angle. A 40 x 40 x 10 mm Noctua cooling fan (Noctua, Vienna, Austria) was used to circulate air to provide cooling.

6.4 Performance Evaluation

6.4.1 Brief Overall Description

The slit lamp, fitted with add-on components, was used to capture images of the ONH targets and calibration target in the eye phantom. These images were then processed using a stereo vision pipeline which was developed in this work, before quantitative validation against ground truth data was performed.

6.4.2 Trial Setup and Image Capture

6.4.2.1 Setup

The experimental setup is shown in Figure 55. The magnification setting on the slit lamp was set to 30 and fixed at this for the entirety of the trial. The focus of the cameras and the dioptre setting of the eye pieces were unchanged from the original setup – the cameras were already focused to infinity and the eye pieces at their zero-dioptre setting. The axial position of the slit lamp lens was adjusted such that the system was infinity focused – allowing imaging of an emmetropic eye. The eye phantom was rigidly mounted in the appropriate position in front of the slit lamp lens, using a frame constructed of aluminium profile struts, such that it could be imaged.

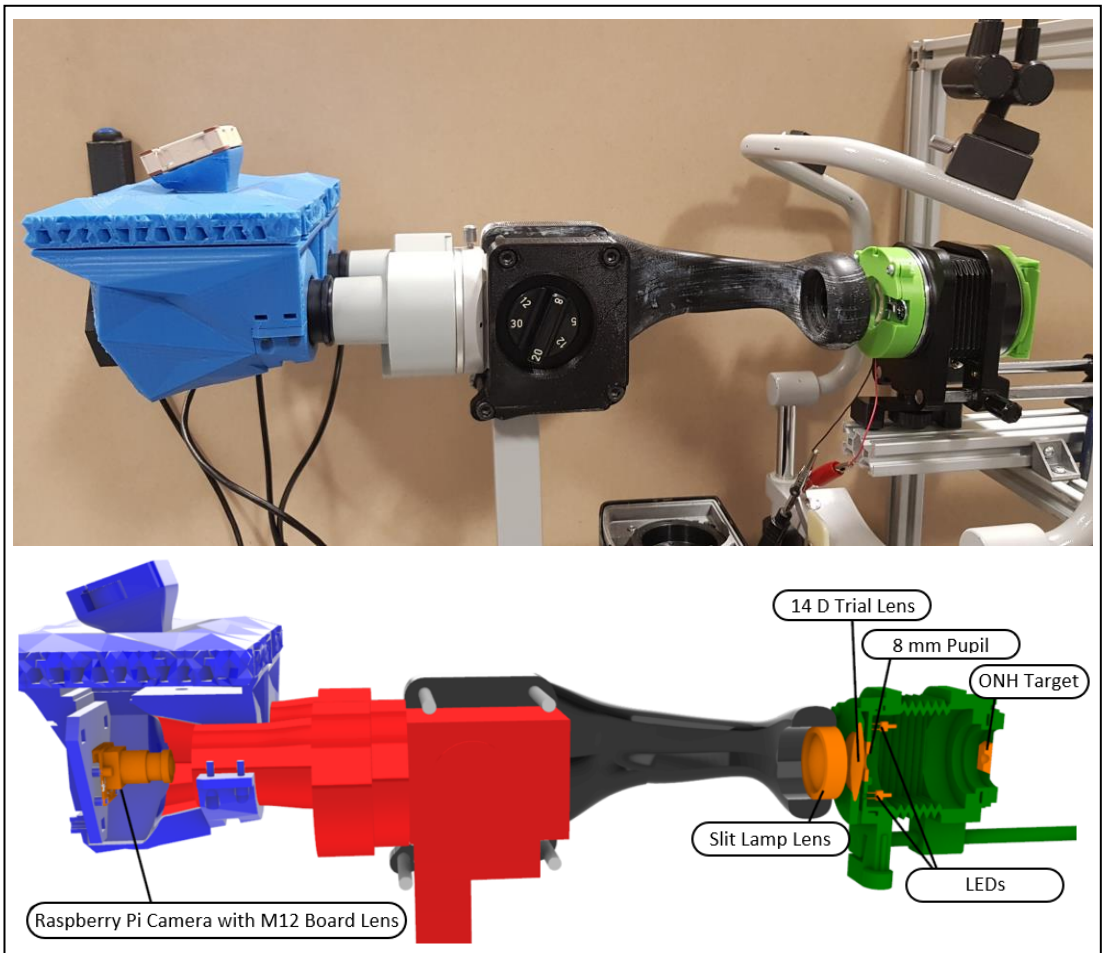


*Figure 55: Photograph (**TOP**) and corresponding colour-coded and labelled CAD rendered 3D model with some sections cut to reveal internal components (**BOTTOM**) of the experimental setup. **Blue** - 3D printed camera mount; **Red** – Zeiss 30 SL-M slit lamp; **Black** – Slit lamp lens mount; **Green** – Eye Phantom; **Orange** – Imaging/optical/illumination components (excluding those of the slit lamp) and optic nerve head (ONH) target.*

One of the ONH targets was placed on the retina of the phantom, and on its optical axis. The LEDs that are inside the phantom that provide illumination to the targets from inside, were connected to a power supply and powered. The axial length of the phantom was then varied until the targets surface came into focus in the images coming from the system's cameras, and that position set using the locking screw. The focus of each of the cameras was then finely adjusted so that the images from both cameras were in as good a focus as practically possible. The room was then completely darkened before imaging began.

6.4.2.2 Image Capture of the ONH Targets

Stereo images were acquired of the ONH target already inside the phantom. The target was then rotated 90° around the phantom's optical axis and again imaged. This was repeated until 4 images were obtained, each with the target at a different rotation, and then the same was done for the other target after interchanging them.

6.4.2.3 Image Capture of the Calibration Target

Following imaging of the targets, the images necessary for calibration of the overall system (imaging system's plus the phantom's optics) were acquired. In preparation for this, the LEDs of the eye phantom were switched off, the ONH target mount was removed and the macro-bellows closed. The LED of the calibration target was then connected to a power supply to power it. The calibration target, held using a flexible helping-hand support (B07PYP4V5S, Amazon.com, Inc, Seattle, Washington, US), was then positioned in many different positions within the field of view of both cameras, and images taken at each position, as was done in the work detailed in the previous chapter, but in this case the pattern was also tilted randomly at some positions, as is required by stereo camera calibration protocols. Such positioning and tilting can be seen through the acquired calibration images, of which 10, taken using the left camera, are shown in Figure 56. In total, 100 stereo images of the

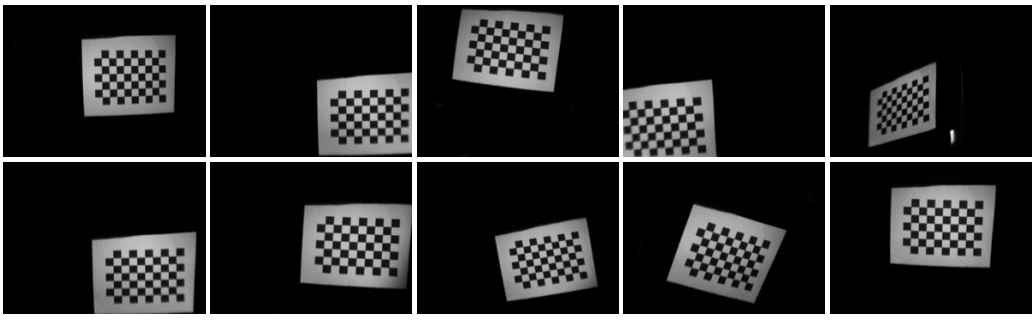


Figure 56: Calibration images from the left camera.

calibration target were acquired. A reasonably large number of images were taken as the exact number required to achieve a stable calibration result was not known.

6.4.3 Stereo Vision Pipeline

6.4.3.1 Overall Description of Stereo Vision Pipeline

The stereo vision pipeline developed involved the following steps:

1. Calibration of the imaging system and rectification, to mathematically map the system to the ideal parallel stereo model. [Subsubsection 6.4.3.2].
2. Stereo matching to yield a sparse disparity map. [Subsubsection 6.4.3.3].
3. Interpolation of the sparse disparity map to yield a dense disparity map [Subsubsection 6.4.3.4].
4. Texture addition and reprojection of the dense disparity map to 3D scene coordinates. [Subsubsection 6.4.3.5].

The calibration of the system, and interpolation of the sparse disparity map, were carried out using the MATLAB programming language and environment. The other parts of the pipeline used code written in the C++ programming language using Visual Studio 2017 Community Edition (Microsoft Corp, Redmond, Washington, US). The MATLAB and C++ codes are provided in Appendix 6A and Appendix 6B, respectively.

6.4.3.2 Calibration and Rectification

The camera calibration toolbox for MATLAB, developed by Bouguet (Bouguet, 2015), was used in this work to perform calibration – estimation of the distortion coefficients, and intrinsic and extrinsic parameters. Regarding distortion, the three radial and two tangential distortion coefficients, which were detailed in subsection 2.3.3, were estimated for each camera. The intrinsic parameters estimated, for each camera, were the two focal lengths, for the x and y axes, and the principal point. The extrinsic parameters estimated were the translation and rotation vectors, which describe the relationship between the two cameras. The toolbox requires that the left and right cameras be calibrated separately but then the results of which be fed into a stereo calibration function which estimates all of the calibration parameters through an optimisation of the entire model, which therefore involves recomputation of the intrinsic parameters. Prior to using the toolbox, a median filter was applied to the images to reduce noise. The first step in using the toolbox to calibrate each camera was to select, for each calibration image obtained, the inside corners of the four

squares at the corners of the checkerboard pattern. The toolbox then performs automatic subpixel estimation of all the corners within the pattern and then uses this information, together with the size of the squares that make up the checkerboard, to estimate the distortion coefficients and intrinsic parameters of the camera. Once this has been carried out for both cameras, the stereo calibration function can be used.

In order to roughly gauge how many image pairs are required before the calibration results stabilise, the calibration procedure was carried out for 10 to 100 image pairs, in increments of 10, and the coefficients, parameters, uncertainties and overall mean reprojection error recorded at each stage. It must be noted that the image pairs were randomly shuffled before such analysis took place as some sets of 10 may have otherwise been quite similar (e.g., all be of the pattern almost flat to the camera or be mostly tilted images).

The toolbox does not report the overall mean reprojection error after performing stereo calibration. Thus, a function was created to calculate it. The function created, was an implementation of equation 5.1 which is presented in subsubsection 5.4.3.2.

The square size of the checkerboard pattern was determined by measuring, in photographs, the length of a number of squares, using the distance between the legs of a pair of ABSOLUTE Digimatic Calipers set to 10 mm as the photographic reference scale, and dividing by the number of squares. This was carried out using 4 photographs (shown in Figure 57), and the final square size value was taken to be the mean of the square sizes determined. The images were taken using a Samsung Galaxy S8 smartphone (Samsung, Seoul, South Korea), set to capture at a resolution of 1952 x 4032 pixels and with its distortion correction setting activated. Measurement in the images was carried out using ImageJ (Schneider et al., 2012). The legs of the calipers were placed such that they were in contact with the patterns surface to mitigate parallax error. In the left two images in Figure 57, the measurements were taken

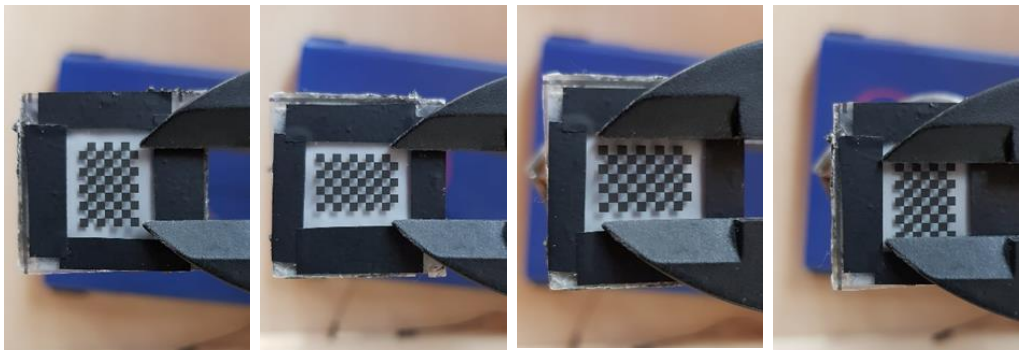


Figure 57: Photographs of the calibration target with calipers in place, used for measuring its square size.

of the squares nearest the legs of the calipers in order to minimise perspective errors. The square size was determined to be 1.165 mm (standard deviation [SD] = 0.005 mm).

The rectification functions of the OpenCV library, which are discussed in great detail in Bradski and Kaehler's book (Bradski & Kaehler, 2008), were used to correct for distortion and rectify the stereo images of the ONH targets, based on the calibration results obtained with 100 stereo image pairs. It must be noted that since the calibration toolbox reports a rotation vector whereas the rectification functions of the OpenCV library expect the rotation matrix, Rodrigues' rotation formula (Murray et al., 1994) was employed to perform the conversion.

6.4.3.3 Stereo Matching

The stereo matching algorithm developed in this work followed a coarse-to-fine scheme, with a left-right consistency check. Illustrations are provided in Figure 58 to complement the description of its operation that follows. Since the algorithm operated on greyscale images, the rectified RGB colour stereo images were first converted to intensity images (or greyscale images). The algorithm involved the following steps:

*A. Initial coarse disparity estimation (*Figure 58*, cell 1):*

A region of interest (shown in blue) is selected in the left image. The best matching patch of the same size (shown in red) is searched for in the right image. The template matching function of the OpenCV library was used to do so, where the search was carried out in one dimension (a horizontal search) over the entire width of the right image (the search range is shown in green). The function calculates a similarity metric between the region of interest in the reference image and all patches of the same size within the search range, and the patch with the greatest similarity is regarded as the best match. The similarity metric chosen to be used was the normalised cross correlation (NCC). In order to calculate the NCC (denoted r_n) between a patch in the left image (P_L) and a single patch in the right image (P_R), the match template function exploits the following equation (Bradski & Kaehler, 2008):

$$r_n = \frac{r_c}{Z} \qquad \textit{Eq. 6.1}$$

where

$$r_c = \sum_{x',y'} [{P_R}'(x',y') \cdot {P_L}'(x',y')]^2 \qquad \textit{Eq. 6.2}$$

in which

$${P_R}'(x',y') = P_R(x',y') - \frac{1}{(w \cdot h)} \sum_{x'',y''} P_R(x'',y'') \qquad \textit{Eq. 6.3}$$

$$P_L(x',y') = P_L(x',y') - \frac{1}{(w \cdot h)} \sum_{x'',y''} P_L(x'',y'') \qquad \textit{Eq. 6.4}$$

and

$$Z(x,y) = \sqrt{\sum_{x',y'} P_R(x',y')^2 \cdot \sum_{x',y'} P_L(x',y')^2} \qquad \textit{Eq. 6.5}$$

where w and h are the width and height of the patch, respectively, and the terms x', y', x'' and y'' are indices of summation. The match template function carries this calculation out for all of the patches within the search range, and returns a matrix containing the NCC values determined for each. Since this work involved only a horizontal search, a vector of NCC values was instead obtained. The position of the best matching patch corresponds to the location where the NCC is greatest. The disparity value can be determined by subtracting distance L from R (which are shown in cell 1 of Figure 58).

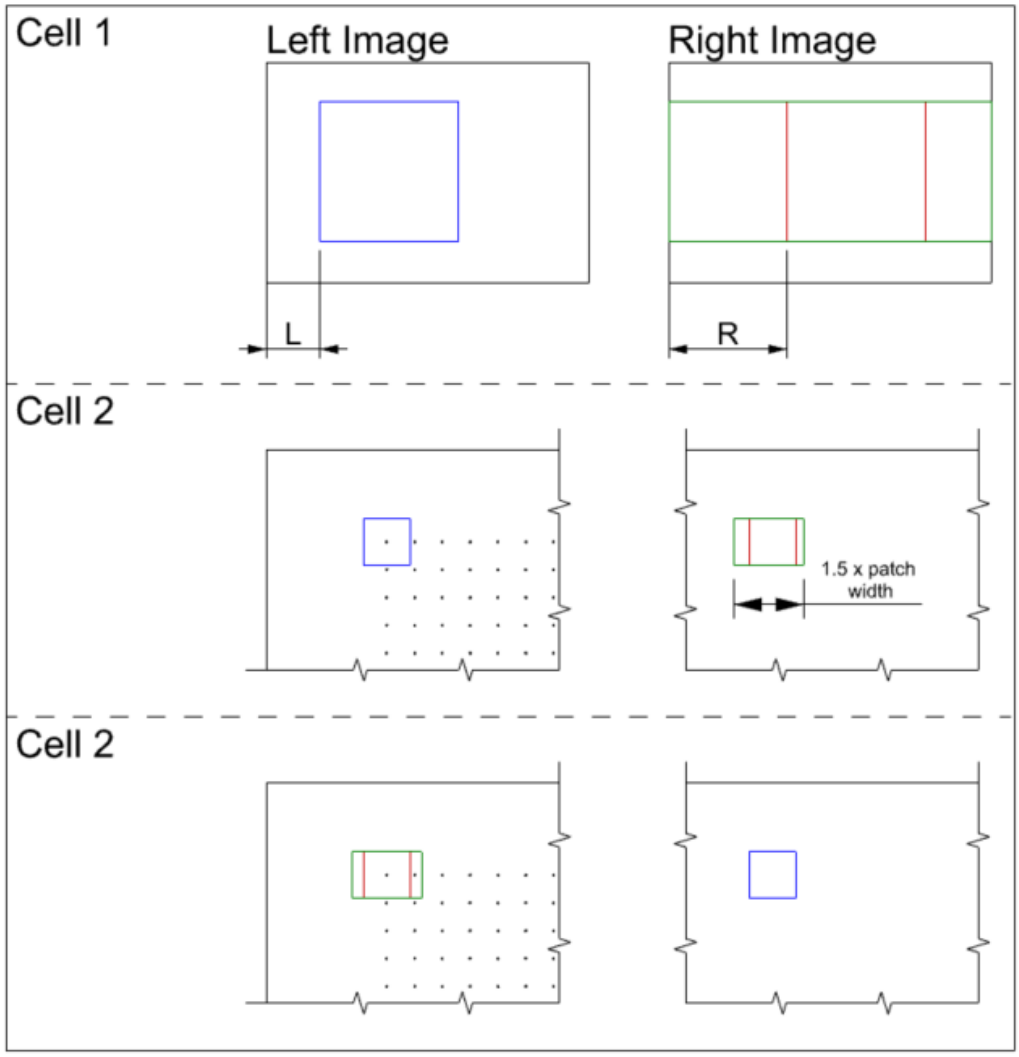


Figure 58: Illustrations to complement the description of the stereo matching algorithm developed in this work.

*B. Creation of a grid of points (*Figure 58*, cell 2):*

The region of interest is filled with a grid of points to be matched. In this work a grid of 40 by 40 points was used.

*C. Setting of initial patch size (*Figure 58*, cell 2):*

The initial height and width of the patch to be matched (shown in blue) were set to be 6 times the vertical and horizontal spacing of the points in the grid, respectively.

*D. Applying a texture threshold (*Figure 58*, cell 2):*

The SD of a normalised version of the patch to be matched is determined. If the value is below 0.2, the algorithm skips to stage I.

*E. Forward matching (*Figure 58*, cell 2):*

The best matching patch of the same size (shown in red) is searched for within a search range in the right image (shown in green) using the same matching method as in stage a. The search range is 1.5 times the width of the patch to be matched and is centred according to the disparity determined in either stage A or, after the first coarseness level, the disparity determined for the point at the previous coarseness level. To obtain a subpixel estimate of disparity, a quadratic equation was fitted to the NCC values corresponding to the disparity values in a ± 3-pixel range centred around the point of maximum NCC and the disparity at which the quadratic reached its maximum was deemed to be the subpixel accurate disparity.

*F. Backward matching (*Figure 58*, cell 3):*

The best matching patch in the left image (shown in red) for the patch in the right image (shown in blue) is searched for within a search range in the left image (shown in green) using the same matching method as in stage E, to obtain the sub-pixel disparity. The search range is also 1.5 times the width of the patch to be matched in this case and is centred at the initial starting grid point.

G. Apply a consistency threshold:

If the disparity values determined in the forward and backward matching stages differ by more than 1.2 pixels, then the algorithm skips to step I. Otherwise, the disparity value kept from this pass is the average of the values determined in the forward and backward matching steps.

H. Reduce patch size and repeat matching:

The height and width of the patch are reduced by a factor of 1.2. Stages D to G are repeated with the new size. Once the algorithm has processed 5 different patch sizes, the algorithm skips to step I.

I. Patch size threshold:

If matching was successful (i.e., it passed stages D and G) for 4 patch sizes, the disparity value determined in the last successful pass is stored for the final sparse disparity map.

J. Perform matching on next point:

Stages C to I are repeated for the next point in the grid. This is repeated until all points have been processed. The result is a sparse disparity map, an example of which is provided in Figure 59.

6.4.3.4 Interpolation

Smoothing splines were first fit to the rows of the 40 x 40 array of points in the disparity map and used to estimate the disparity value for each pixel along those rows. The fit function of the MATLAB programming language was used to do so, with the smoothing parameter set to 4×10^{-6}. An example of the output of this is shown in the top image of Figure 60. Following this, the same procedure is repeated for each column of the disparity map output from the

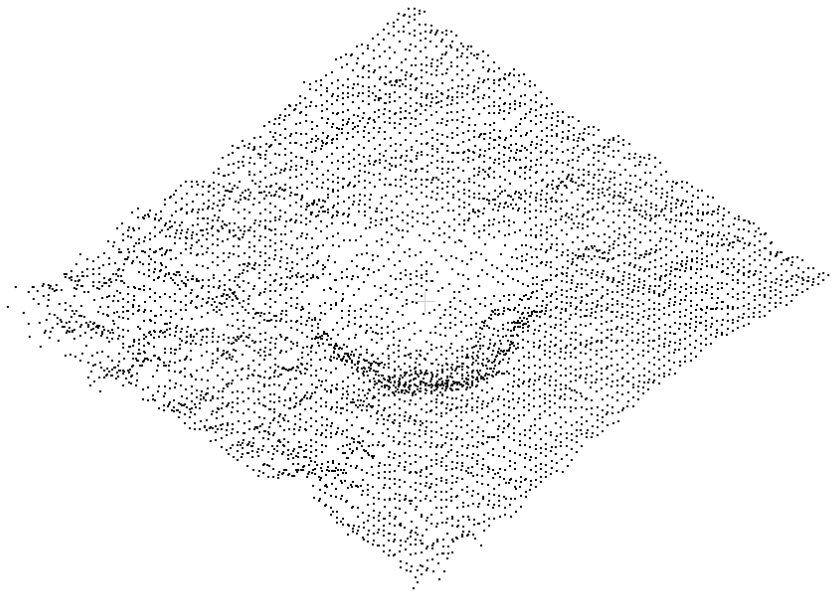


Figure 59: Sparse disparity map.

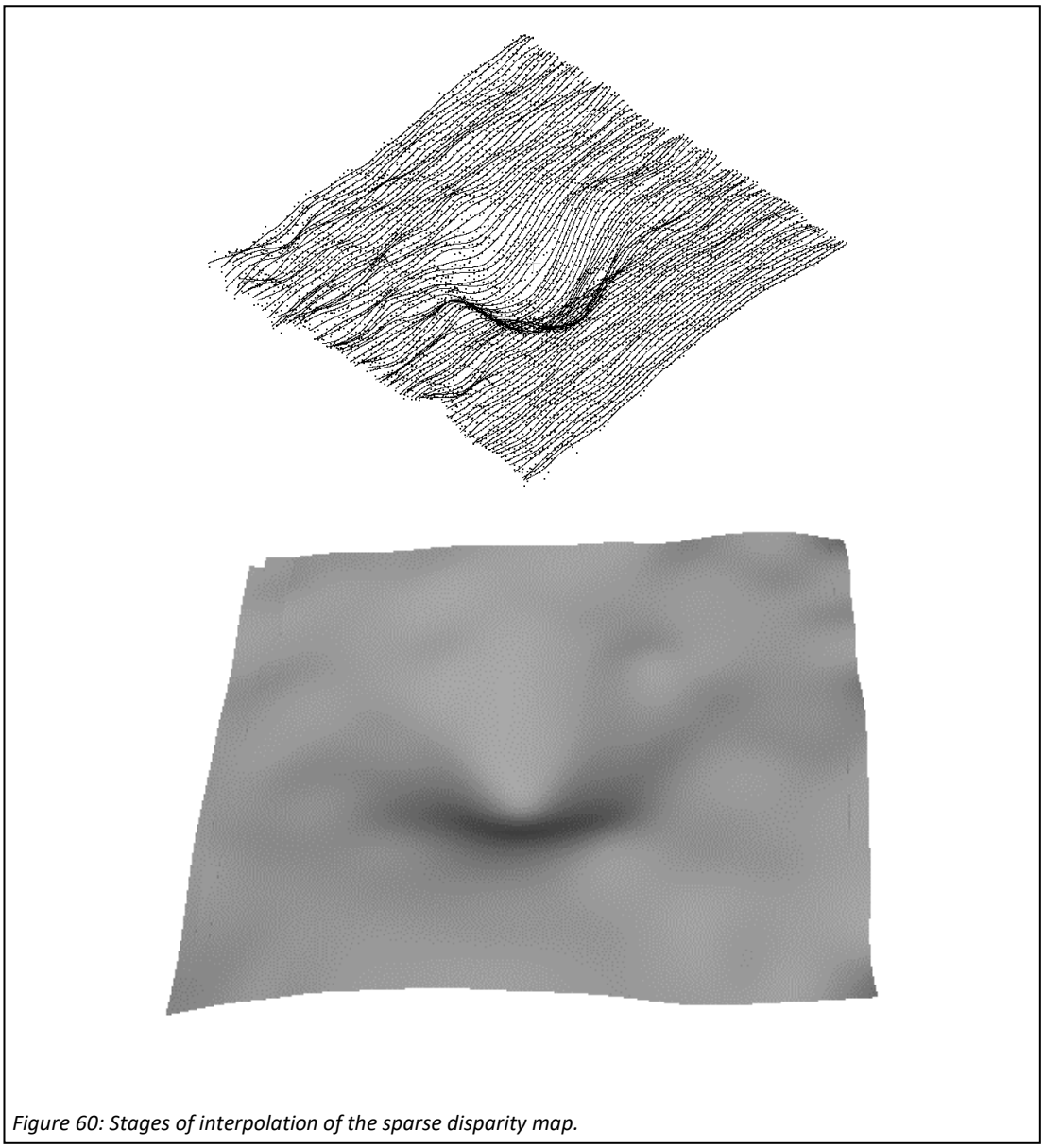


Figure 60: Stages of interpolation of the sparse disparity map.

previous stage, resulting in a dense disparity map. An example dense map is shown in the bottom image of Figure 60. It should be noted that the disparity values of the maps shown in Figure 60 were scaled so that the dimple like morphologies could be seen, as the map appears flat otherwise – the disparity values are much smaller than the range of image coordinate values (forming the x- and y- axes).

6.4.3.5 Reprojection and Texture Addition

The rectification functions of the OpenCV library output, as well as rectified images, a reprojection matrix (denoted Q), which can be used to reproject points from the rectified left image (x, y) back into 3D scene coordinates (X, Y, Z), if the disparity (d) of the point is known. The matrix Q takes the following form:

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & \frac{-1}{b} & 1 \end{bmatrix} \qquad \textit{Eq. 6.6}$$

The terms used in the matrix were described earlier in subsection 2.3.2. Note though that the focal lengths, in the x and y direction, of the cameras in the system were made to be equal to a single focal length, through the rectification procedure. Reprojection is governed by the following relationship:

$$Q \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = W \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad \textit{Eq. 6.7}$$

where the term W is a normalisation factor that scales the 4th element of the right-hand term to one. This equation was used to reproject the dense disparity map data (x, y, d), along with the RGB data of the pixels, into the 3D scene space, yielding a textured dense point cloud. An example of which is shown in Figure 61.

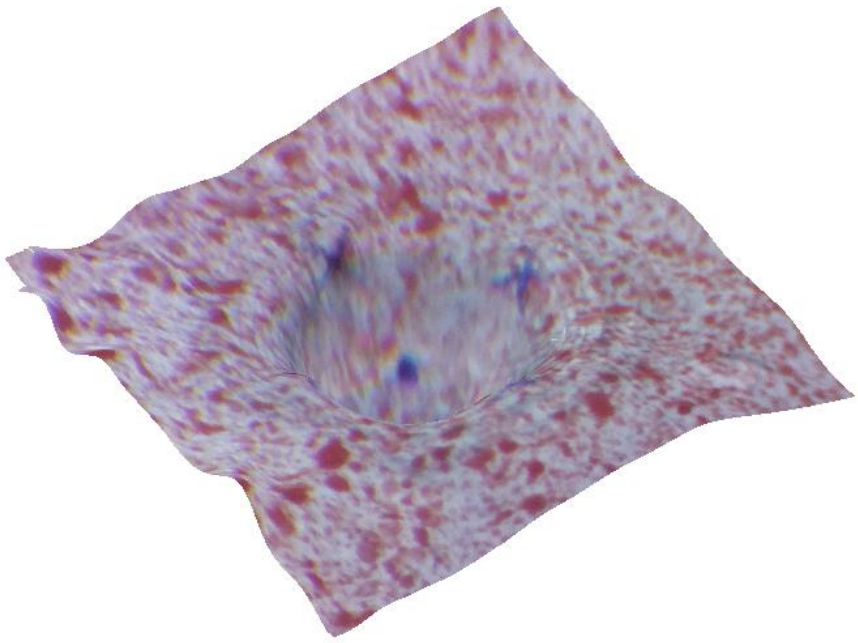


Figure 61: Final textured dense point cloud.

6.4.4 Quantitative Validation

6.4.4.1 Ground Truth Data Acquisition

Ground truth scans of each of the targets were obtained using the commercial photogrammetry software 3DF Zephyr Free (3Dflow, Verona, Italy). In turn, each of the targets were held on the top of a clamp stand's post using Bostik Blu Tack (as is shown in Figure 62), and 50 images of their top surfaces, containing the dimple-like ONH structures, were acquired from different and random poses using the camera of a Samsung Galaxy S8 smartphone, set to capture at a resolution of 4032 x 1960 pixels. The targets were tilted towards a window for illumination purposes. An example of one of the acquired images, cropped to show only the model, is shown in Figure 63.

The images acquired of each target were input into the software, which was set to high quality settings, to yield proportionally accurate ground truth reconstructions. These were uniformly scaled in three dimensions to the correct size using the mesh processing software

Figure 62: Setup for acquiring the images of the ONH targets for ground truth scanning using photogrammetry software.

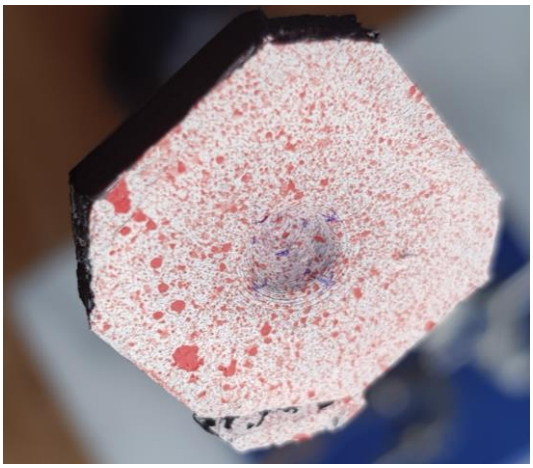


Figure 63: One of the images obtained of a target during ground truth acquisition.

CloudCompare (*CloudCompare*, 2020). The scaling factor for each was determined by taking measurements of the distances between certain speckle features on the top surfaces of the physical targets and comparing these against the same measurements taken on the acquired scans. The measurements of the physical targets were taken from photographs, where the legs of a pair of ABSOLUTE Digimatic Calipers, set to be 10 mm apart, were used to act as a photographic reference scale. The legs were made to touch the top surfaces and distances were measured close to the legs to limit parallax and perspective errors, respectively. The software package ImageJ was used to perform the measurements on the images. The images were acquired using a Samsung Galaxy S8 smartphone, set to capture at a resolution of 1952 x 4032 pixels and with its distortion correction setting activated. The measurements of the same distances on the scans were made using CloudCompare. In total, 4 images were taken, with the legs of the calipers at a different edge of the target in each. An example of one of the images acquired of the glaucomatous ONH target; the distance determined on the physical targets and the scan's surface; and the corresponding final ground truth reconstruction is shown in Figure 64.

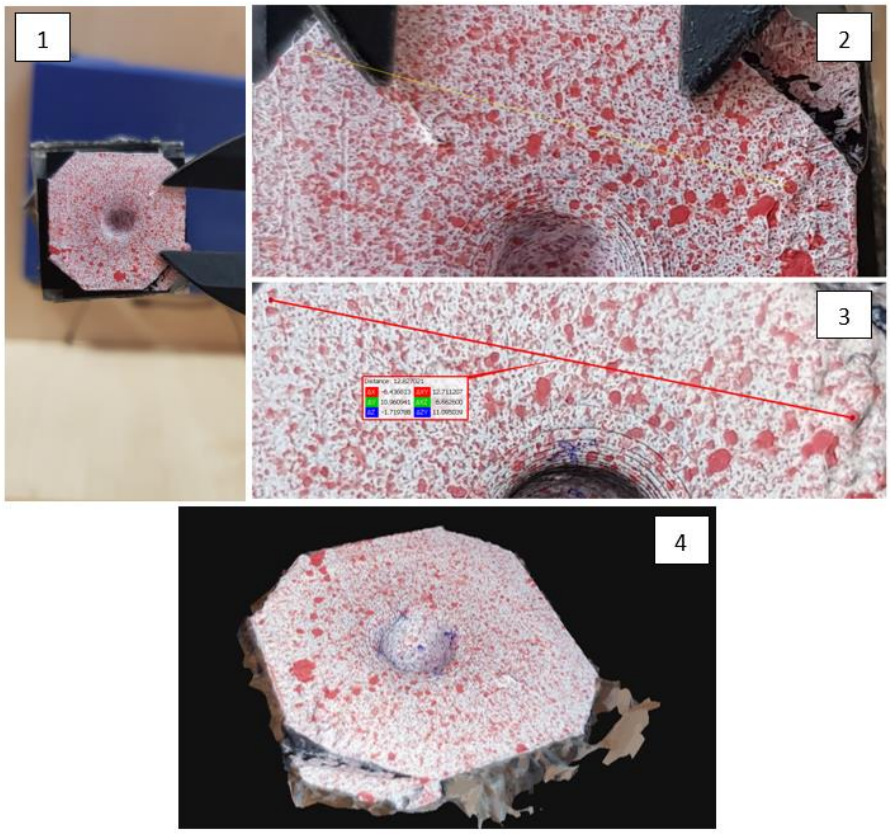


Figure 64: Ground truth and its scaling. ***IMAGE 1****: One of the images of the glaucomatous model used to determine the ground truth scaling factor.* ***IMAGE 2****: Measurement of the physical model using ImageJ.* ***IMAGE 3****: Measurement of the unscaled ground truth using CloudCompare.* ***IMAGE 4****: Ground truth reconstruction obtained.*

6.4.4.2 Quantitative Comparison

The CloudCompare software was also used to perform the comparison between the obtained 3D reconstructions and the ground truths. Two forms of comparison were made. In the first, two measurements were made of the dimple-like structures diameter in each reconstruction and the ground truth. The measurements were made using the anchor points present on the targets' top surfaces. Measurement 1 was made between points A and B, and measurement 2 was made between points C and D, according to the labels shown in Figure 65.

The second form of comparison was performed on a cloud-to-cloud basis. This involved first aligning the reconstructions to their corresponding ground truths using the anchor points on each target. This was done using the align tool within CloudCompare which performs registration based on selected point pairs. The transformation applied during registration was a rigid body transformation only. Following this, the perpendicular signed distance errors between the ground truth and each reconstruction, for each point of the ground truth, were determined. The mean and SD of such errors for each point on the ground truth were determined, giving an idea of the accuracy and precision of the reconstructions, respectively.

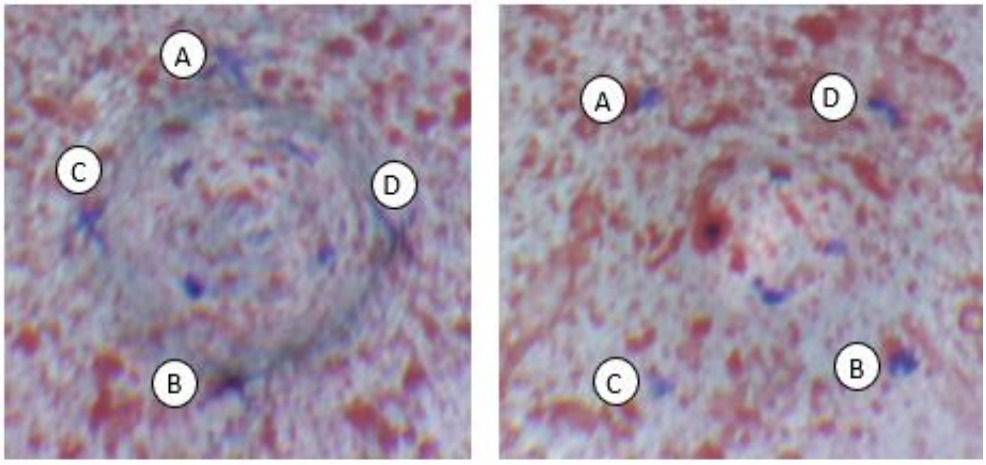


Figure 65: Marks on the targets' surfaces where measurements were made between. ***LEFT****: Glaucomatous ONH target.* ***RIGHT****: Normal ONH target.*

6.5 Results and Discussion

6.5.1 Calibration Performance

The coefficients and parameters obtained in the calibration with 100 image pairs are shown in Table 5. Also shown in this table are the associated uncertainties, expressed as absolute

Table 5: Calibration coefficients and parameters estimated for the system. SD: Standard Deviation.

	Coefficient/Parameter	Value	Uncertainty (3SD)	% Uncertainty
Left	f_x (pixels)	8655.2	47.0	0.543026
	f_y (pixels)	8624.3	47.0	0.544972
	C_x (pixels)	1965.4	10.4	0.529154
	C_y (pixels)	530.3	12.8	2.413728
	k_1	-1.819	0.031	-1.70423
	k_2	23.685	1.013	4.276969
	k_3	-132.572	8.412	-6.34523
	P_1	0.0123	0.0014	11.38211
	P_2	-0.00033	0.00100	-303.03
Right	f_x (pixels)	8229.2	45.6	0.554124
	f_y (pixels)	8352.9	46.6	0.55789
	C_x (pixels)	65.5	N/A	N/A
	C_y (pixels)	-41.7	N/A	N/A
	k_1	-0.991	0.018	-1.81635
	k_2	2.746	0.124	4.515659
	k_3	-4.286	0.340	-7.9328
	P_1	0.0123	0.0005	4.065041
	P_2	0.04850	0.00110	2.268041
	R_1	-0.08067	0.00150	-1.85943
	R_2	0.19501	0.00150	0.769191
	R_3	0.02999	0.00021	0.700233
	T_x (mm)	2.40660	0.02080	0.86429
	T_y (mm)	0.29370	0.01970	6.707525
	T_z (mm)	0.35330	0.10530	29.8047

values and relative percentages. It can be seen that the greatest relative uncertainties for each of the cameras are mainly for the distortion coefficients. This would tie in with the fact that the distortion maps obtained, as shown in Figure 67 and Figure 68 for the left and right cameras respectively, appear abnormal compared to that which would be expected from a typical stereo camera system but those for the right camera appear particularly abnormal. Relatively high uncertainties can also be seen for the camera translation parameters, although not for the x component of translation.

The calibration coefficients and parameters appeared to all stabilise by 70-80 calibration images. While not shown here due to the large number of them, the graphs of their values against the number of images used, are provided in Appendix 6C. A graph showing the overall mean reprojection error for 10 to 100 images, in increments of 10, is shown in Figure 66. It appears that the value of this metric tended to remain between around 0.85 and 0.82 pixels from the calibration using 20 images onwards. The value determined using 100 calibration images was 0.876 pixels (SD = 0.589). This value represents 4.5% of the total range of disparity values determined for one of the sparse disparity maps of the normal ONH target. It still needs to be determined what level of error would be sufficient for this work.

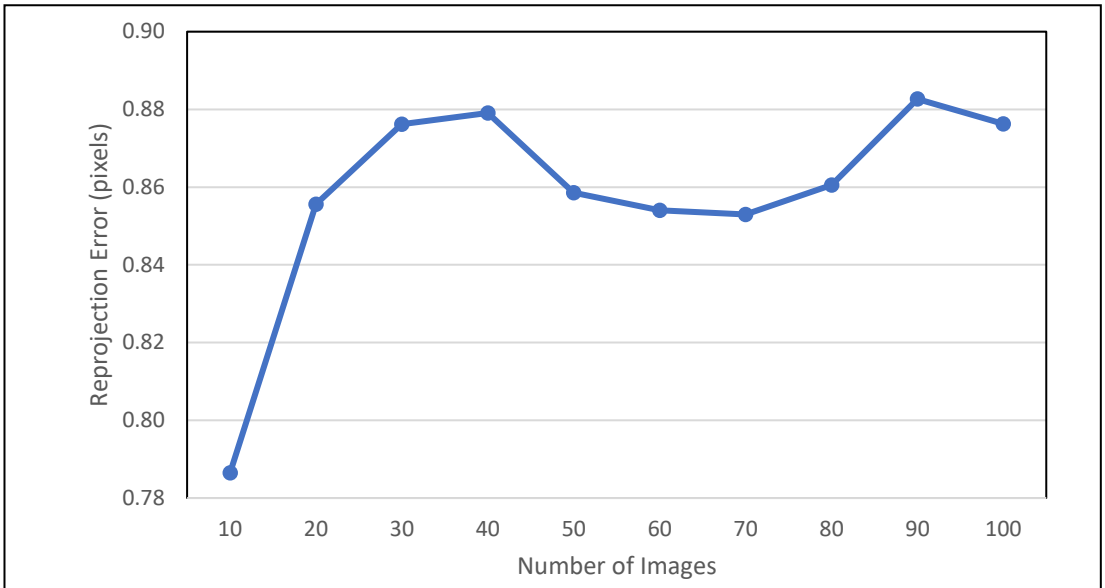


Figure 66: Overall mean reprojection error determined using 10 to 100 calibration images, in increments of 10.

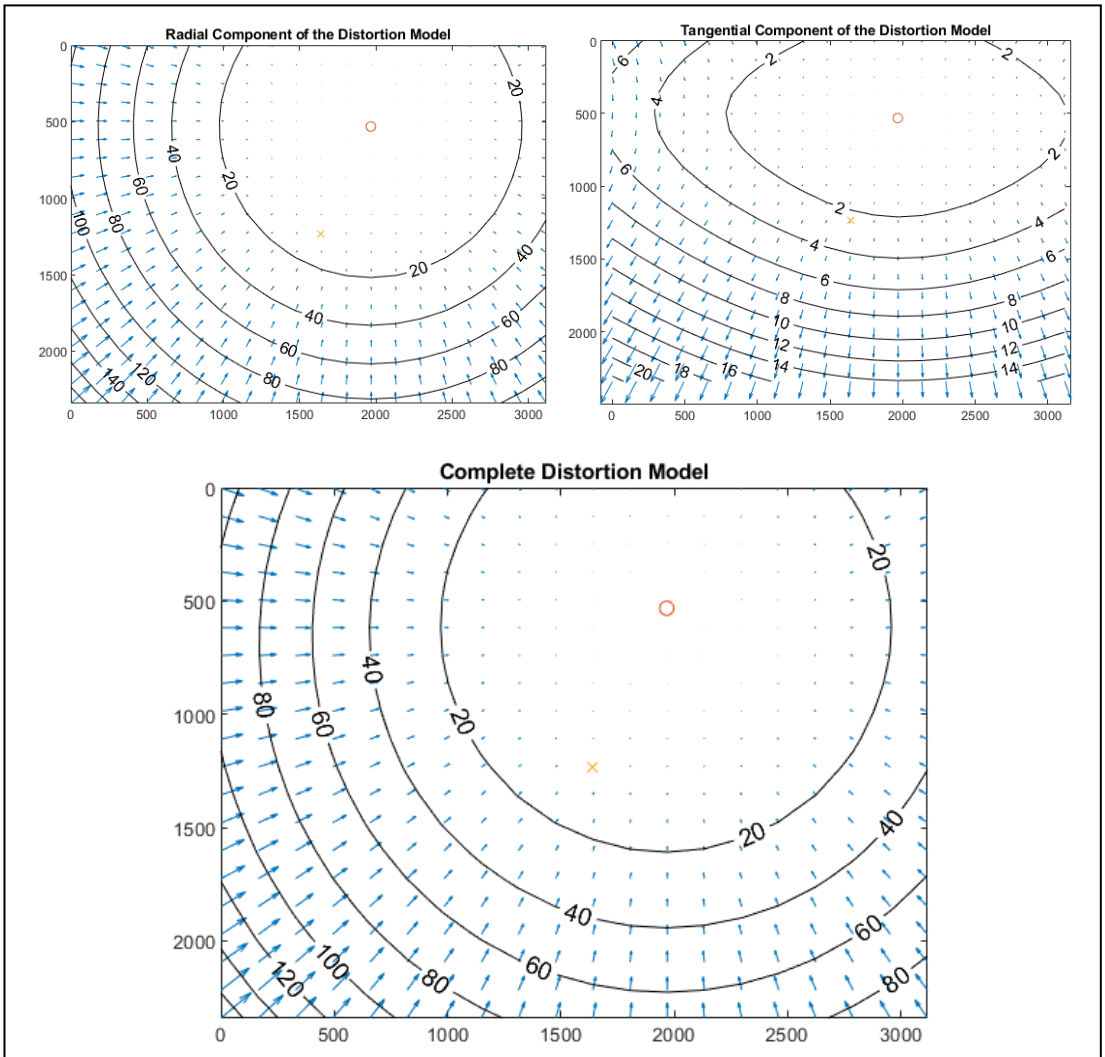


Figure 67: Distortion maps to visualise the distortion coefficients estimated for the left camera.

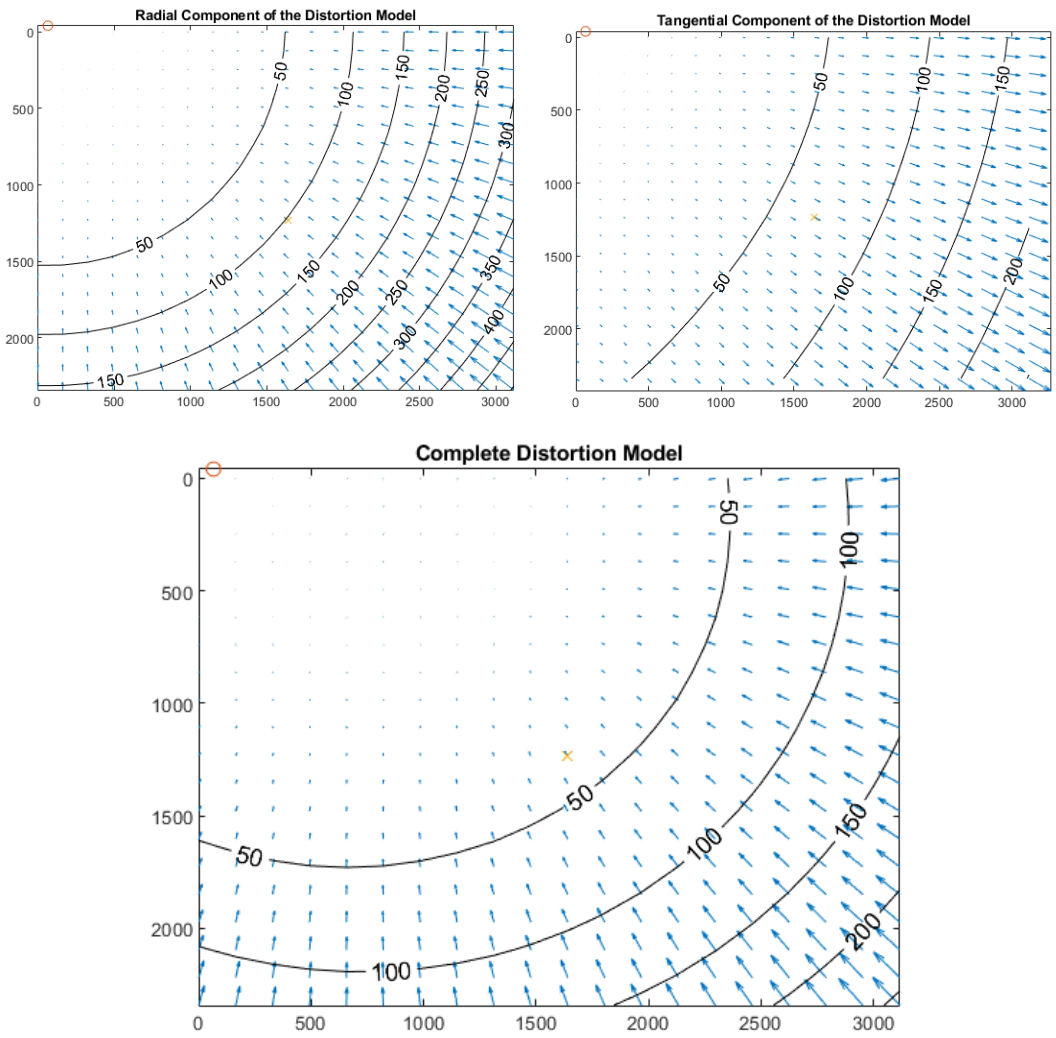


Figure 68: Distortion maps to visualise the distortion coefficients estimated for the right camera.

It was seen that most of the coefficients and parameters stabilised by 100 images, meaning that it is unlikely that more images of the same type and quality would have led to very different calibration results. It is also unlikely that it would have significantly changed the overall mean reprojection error. It is therefore likely that exploring aspects other than the number of images may prove more worthwhile in helping to reduce calibration uncertainty and improve accuracy. Regarding accuracy, it is difficult to obtain the measurements which correspond to some of the parameters (e.g., baseline distance and focal length) in order to validate them due to the complex nature of the system compared to standard stereo camera rigs. The potential sources of error which could have contributed to calibration error include:

Corner localisation errors: inaccurate subpixel localisation of the corners of the checkerboard likely contributed to calibration error. Corners where focus was poor,

particularly the case for some corners when the checkerboard is held at an angle, due to the shallow depth of focus of the system, or near the edges of the field of view, may have been difficult to accurately locate. Future work may benefit from removing images in which corners are in poor focus. Moreover, a look into different types of calibration pattern may prove useful. It may be the case that patterns of circles offer more accurate subpixel localisation in the presence of image blur. The resolution of the cameras is particularly high and is therefore likely not a significant contributor to corner localisation errors.

Inaccurately fabricated checkerboard: It could be the case that the checkerboard is either not completely flat or contains some inaccuracies within its pattern. Regarding its flatness, it was glued to a standard glass microscope slide which was thought to be perfectly flat, however it is not clear whether the film containing the pattern is also perfectly flat (i.e., if the glue was of a uniform thickness). Regarding the accuracy of the pattern, the process used to create it may have introduced some error. In using a camera to capture images of the pattern displayed on a screen, distortion could have been introduced. It is likely that a significant amount was not introduced since the pattern was small and at the centre of the image. Moreover, other distortions could have been introduced if the camera was not perfectly in alignment with the screen. Thus, it may be beneficial in future to explore having the calibration pattern precisely made, already on a flat material.

Inaccurate checkerboard square size estimation: Errors could have been introduced by measurement errors of the checkerboard size. Exploring a measurement technique, known to be more accurate and precise, would likely be beneficial.

Sub-optimal camera alignment: The locations of the principal points were very far from where they would be expected to be in a traditional camera system (i.e., very close to the centre of the image). Sub-optimal camera alignment may have been the reason for this. It may be beneficial in future to therefore concentrate on better alignment of the cameras. Moreover, given that in order to use M12 lenses with the cameras, some modification had to be done, which may have led to poor alignment of the sensor with the lens, the use of different cameras, where such modification is not required, may lead to better alignment.

Inadequacy of distortion models: Given that the system is quite different to traditional camera systems, in terms of optical configuration, for which radial and tangential distortion models are able to well describe the lens distortion present, those models may not be able to adequately describe the distortion of this system.

6.5.2 Reconstruction Performance

6.5.2.1 Results

The reconstructions obtained of the normal ONH target, with the corresponding stereo image pairs, and the ground truth data, are shown in Figure 69. The same for the glaucomatous ONH target are shown in Figure 71. The measurements made for each of the targets, on their reconstructions and ground truths, are tabulated in Table 6. Colour error maps, showing the accuracy and precision of the reconstructions obtained for the normal and glaucomatous targets, are shown in Figure 70 and Figure 72, respectively.

6.5.2.2 Discussion

Measurements: The values obtained for each of the measurements indicate that the diameters measured on the reconstructions were close but slightly overestimated in both cases. There are a number of potential reasons for the discrepancies, including reconstruction error (as discussed below) and measurement error.

Reconstructions and Cloud-to-Cloud Comparison: It is evident from the depth maps for the reconstructions of each target that the reconstructions do not have a perfectly circular appearance, like the ground truth does. Moreover, it can be seen in these and the reconstructions, that areas furthest from the centre (i.e., the edges) appear to look abnormal, and not practically flat which they are in the ground truth. This trend can also be seen in the accuracy and precision maps. The least accurate and precise areas are those near the edges. A clear correlation can be seen between the location of the anchor points and the accuracy maps. This is likely an impact of the visual texture on the stereo matching algorithm, of a nature yet to be determined, since the only difference that these locations have compared to adjacent regions of the same depth are texture related. In general, it appears that global to local errors are the main errors which are present. These likely come from the sources detailed below. The reconstructions obtained are smooth and free of noise, therefore noise has not been considered.

Table 6: Reconstruction and ground truth measurements. SD: Standard Deviation.

Target	Type	Measurement 1	Measurement 2
Normal	Reconstruction	Mean = 7.2 mm (SD = 0.1)	Mean = 7.1 mm (SD = 0.1)
	Ground Truth	7.0 mm	6.8 mm
Glaucomatous	Reconstruction	Mean = 6.0 mm (SD = 0.1)	Mean = 5.8 mm (SD = 0.1)
	Ground Truth	5.9 mm	5.7 mm

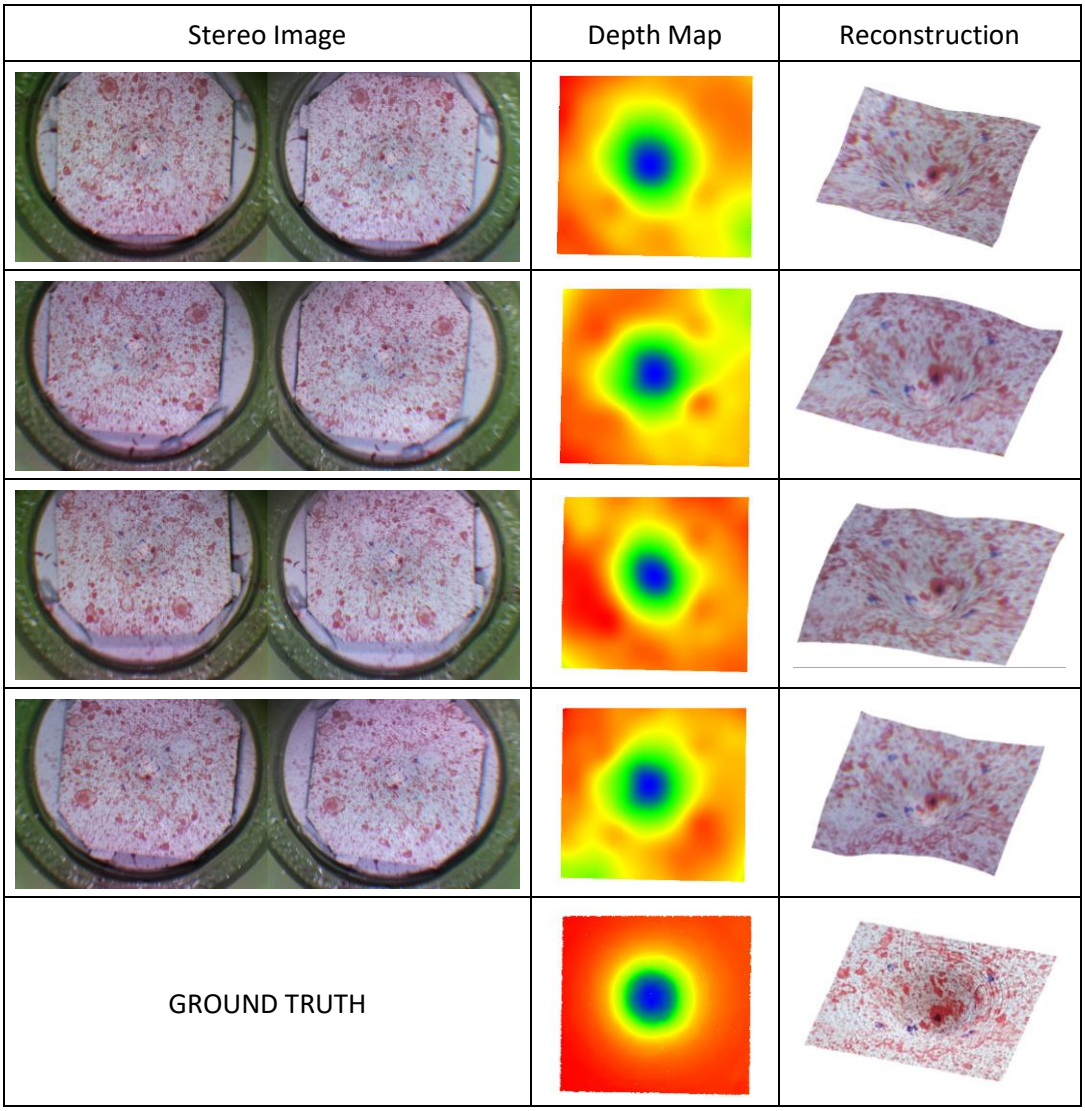


Figure 69: Reconstructions obtained of the normal ONH target, alongside the ground truth data.

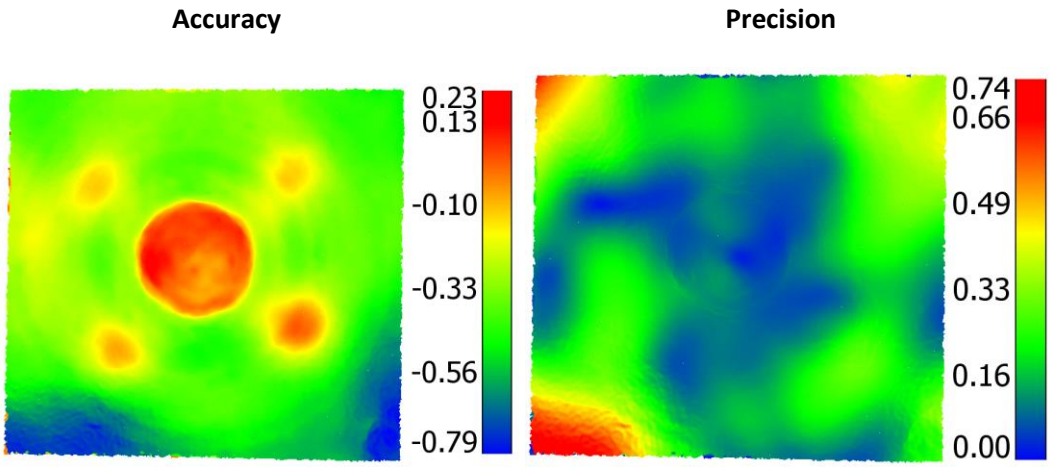


Figure 70: Accuracy (mean of distance errors) and precision (standard deviation of distance errors) maps of the reconstruction of the normal ONH target. Values are in units of millimetres.

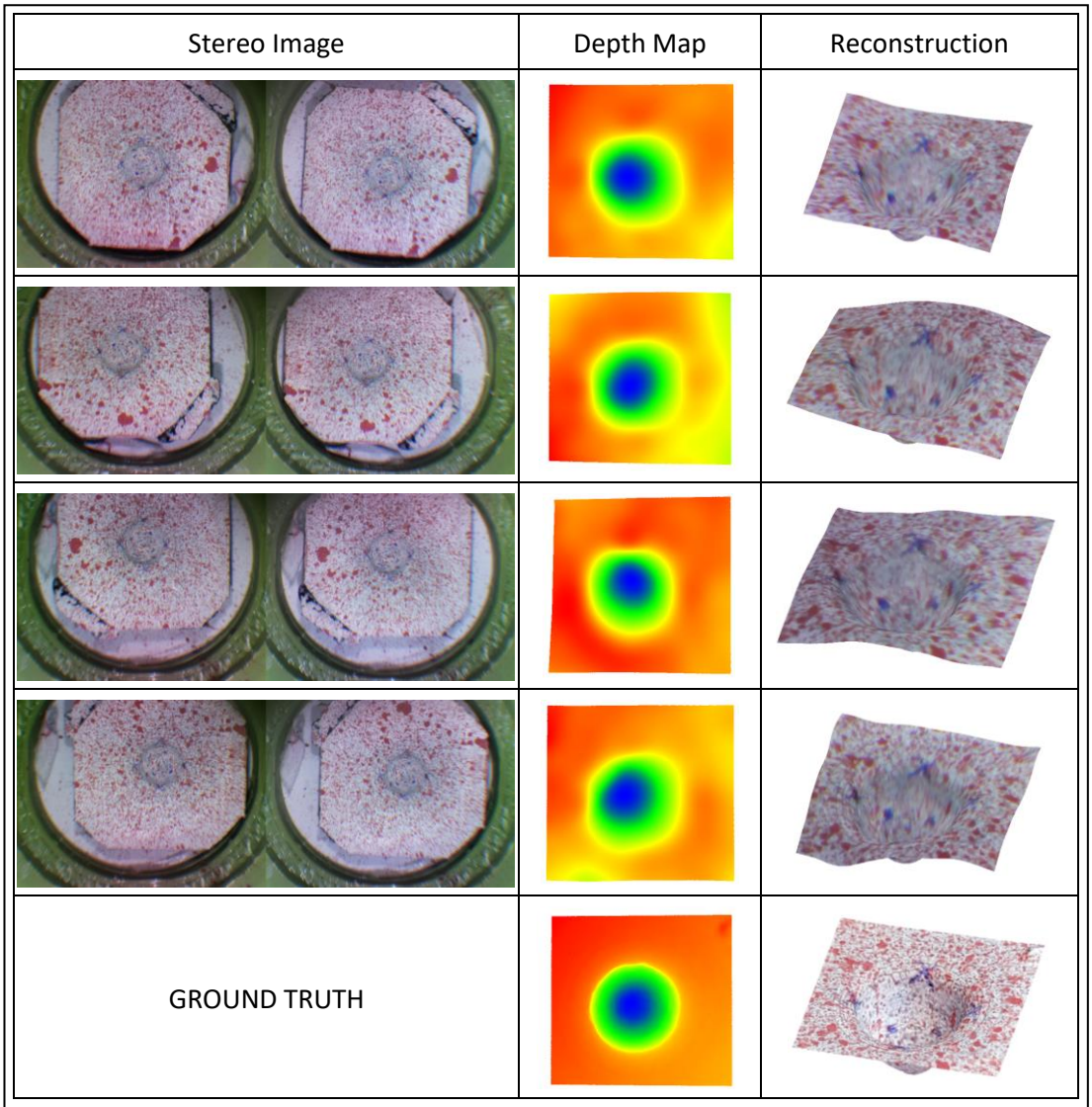


Figure 71: Reconstructions obtained of the glaucomatous ONH target, alongside the ground truth data.

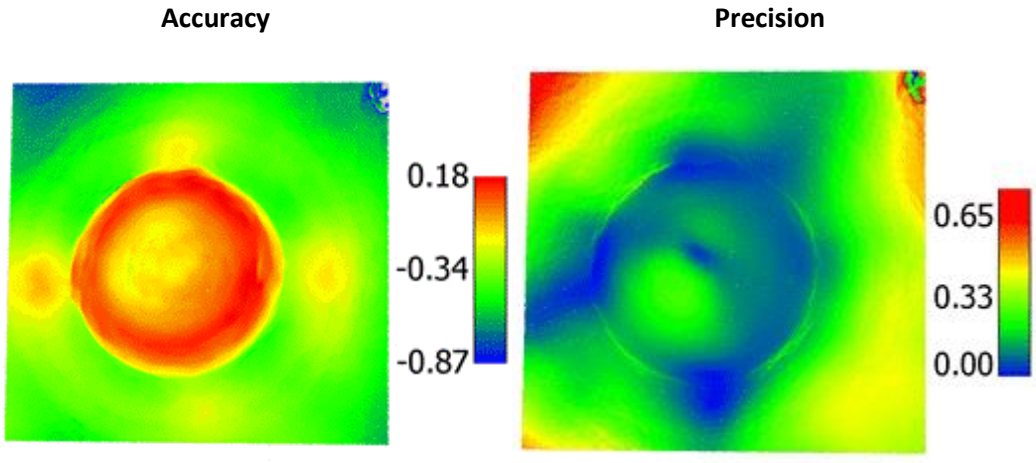


Figure 72: Accuracy (mean of distance errors) and precision (standard deviation of distance errors) maps of the reconstruction of the glaucomatous ONH target. Values are in units of millimetres.

6.5.2.3 Main Sources of Reconstruction Error

Stereo matching: The stereo matching algorithm developed for this trial, together with the fact that the resolution of the cameras was higher, seems to not be affected by the quantisation-like errors which were present in the work detailed in the previous chapter – the reconstructions obtained in this work are smooth. The fact that the reconstructions did not have a practically perfect circular appearance, and that the anchor points were seen in the accuracy maps, indicates that stereo matching performance is still sub-optimal.

Calibration: Calibration is likely another major source of error in the reconstructions. Errors in calibration propagate all the way through the reconstruction process. They first can lead to poor rectification results which then introduces errors when stereo matching is performed, then also in the reprojection of the disparity map to real-world coordinates. Work needs to be done to investigate this type of error more thoroughly.

Image focus: As was the case in the work of the previous chapter, the stereo images obtained had local areas of poor focus. This almost certainly degrades stereo matching performance.

6.5.3 Limitations of This Work

The limitations which were described in the previous chapter, which also apply to this trial, are as follows:

- Surface texture of ONH targets [subsubsection 5.6.4.2]
- Optical properties of ONH targets [subsubsection 5.6.4.3]
- Illumination [subsubsection 5.6.4.5]
- Eye phantom optics [subsubsection 5.6.4.6]

The additional limitations apply to this trial:

- **Ground truth accuracy**: it is not entirely clear how accurate the method used for ground truth scanning, through feeding images into a professional photogrammetry software package, was/is. Even if the software reconstructed the ONH targets correctly in terms of proportion, its scale still needed to be manually corrected. In this work, scale factors were determined based on measurements made on the physical targets and the reconstructions. Some error could have been introduced as a result of incorrect scale factors. Future work would benefit from using a scanning system known to be accurate and precise enough for this work.

- **Camera synchronicity**: while this was mentioned for the work in the previous chapter, it is mentioned here since there are some differences. In this case, the cameras are supposed to be able to capture simultaneously due to being orchestrated by the single StereoPi board. That being said, it is unclear to what degree it can synchronise the cameras. On top of this, the cameras used have a rolling shutter, which may not be ideal for imaging of a moving target for quantitative work. Future work, especially when the time comes to perform in-vivo work, would benefit from using cameras which have a global shutter and that can be sufficiently synchronised.
- **Precision assessment**: Since the reconstructions were performed at different angles of rotation, systematic errors such as calibration errors, could have affected the precision maps obtained. It may be beneficial in future to remove and replace targets, maintaining its rotational orientation, to assess the precision, since reconstruction of human ONHs would not involve different rotations at each scan. It may actually be useful to perform reimaging of the targets by moving the slit lamp out of the way and then repositioning it between captures, which would be more in line with what would happen when imaging human ONHs.
- **Shape of ONH targets**: Another aspect related to the ONH models that is not a huge limitation but could be useful to consider in future work is the shape of the structures of the ONH targets. In this work, the two different ONH structures created (normal and glaucomatous) were rotationally symmetric, while biological ONHs are typically not. It is not clear if the same performance, as seen in the eye phantom experiments, would be obtained if the ONH was not rotationally symmetric. It would therefore be interesting to create ONH targets based on full OCT scans.

6.6 Conclusion

The target of this work was to address some of the issues uncovered in the trial detailed in the previous chapter. The main aspect of this was to allow for quantitative reconstructions to be obtained, and to assess their accuracy and precision against ground truth data. A further aim was to improve performance. This work achieved such aspects. Moreover, it showed that while the reconstructions obtained resembled the ground truth data, there are still likely significant improvements to the system and algorithms that can be made. The

issues which were identified in this chapter and the previous, as well as their status, are provided in Table 7.

Table 7: Issues identified until this stage and their current status (chapter 6). ***Changes are shown in bold.***

Issue Number	Issue	Status
1	Stereo matching technique resulted in quantisation effect	**The newly implemented coarse-to-fine stereo matching algorithm did not suffer from this issue.**
2	Low image resolution	**The cameras used here (Raspberry Pi Cameras) were of a much higher resolution than those used in the work of chapter 5. Resolution is likely no longer a significant limiting factor.**
3	Calibration performance sub-optimal	**The reprojection errors appear large and the distortion maps abnormal.**
4	Unstable rectification results	**The rectification process used here appears to give repeatable rectification outputs.**
5	Poor focus in some areas of the images	**This is still an issue at this stage.**
6	Validation was only qualitative since reconstructions are uncalibrated	**Reconstructions in this work were calibrated and quantitative validation was performed.**
7	ONH target surface texture different to human ONH	**The surface texture density was still greater than that of the human ONH in this work. The ONH targets had speckle painted onto their top surfaces.**
8	ONH target optical properties different to human ONH	**The targets used were still opaque – different to the human ONH.**
9	ONH target shape different to human ONH	**The shapes of the targets in this work were more similar to the human ONH, than those used in the work of Chapter 5, but were rotationally symmetrical - different to the human ONH.**
10	Illumination not provided from outside the eye	**Illumination was also provided from inside the eye phantom in this work.**
11	Eye phantom optics different to human eye	**The optics of the eye phantom used here were simple compared to the optics of the human eye.**
12	Image capture is not synchronous	**The synchronicity of the cameras used was not tested in this work.**
13	**Stereo matching performance sub-optimal**	**There are many settings for the stereo matching algorithm developed which were not optimised.**
14	**Repeated measures not representative of imaging a human eye**	**Precision is yet to be determined where the ONH targets are imaged in the same location in the field of view of the cameras.**
15	**Unclear ground truth accuracy and precision**	**Future work would benefit from using a scanning method whereby its precision and accuracy are known.**

CHAPTER 7

ADDITION OF A PATTERN PROJECTOR TO THE SLIT LAMP SYSTEM AND RECONSTRUCTION OF TEXTURELESS ONH TARGETS

7.1 Introduction

The work of this chapter built on the work of the previous chapter through further development and testing, to address some of the issues presented in Table 7. The main aim of this work was to perform reconstruction of ONH targets which had a texture density which was more in line with that of the human ONH, addressing issue 7 (ONH target surface texture different to human ONH). As a result, ONH targets were created which have a white and practically textureless surface apart from the presence of some ink anchor points. The lack of texture presented difficulties to the stereo matching algorithm and, as such, a pattern projector was incorporated into the slit lamp system to project a speckle dot pattern onto the surface of the targets, artificially creating texture. The stereo matching algorithm used here was different to the one developed in the work of the previous chapter and was tailored to the pattern being projected, by performing matching only with patches containing speckle dots. This work was the first in this project where illumination was provided from outside the eye phantom, addressing issue 10 (illumination not provided from outside the eye).

7.2 Eye Phantom Development

7.2.1 Brief Overall Description

The eye phantom used in this work was a slightly updated version of the one used in the previous two chapters. In terms of optical configuration, it was essentially the same, but it was constructed using an optical cage system which gave it an open design, allowing for the projection on the ONH targets to be directly visualised. Moreover, it contained a lens of higher optical quality, but of around the same power, instead of the trial lens used previously. 3D CAD rendered views, including a colour-coded half-sectional view, and a photograph of the fabricated eye phantom are shown in Figure 73.

7.2.2 Optical Design

The optical design considerations for this phantom remain unchanged from those for the previous eye phantom (detailed in subsection 5.2.2).

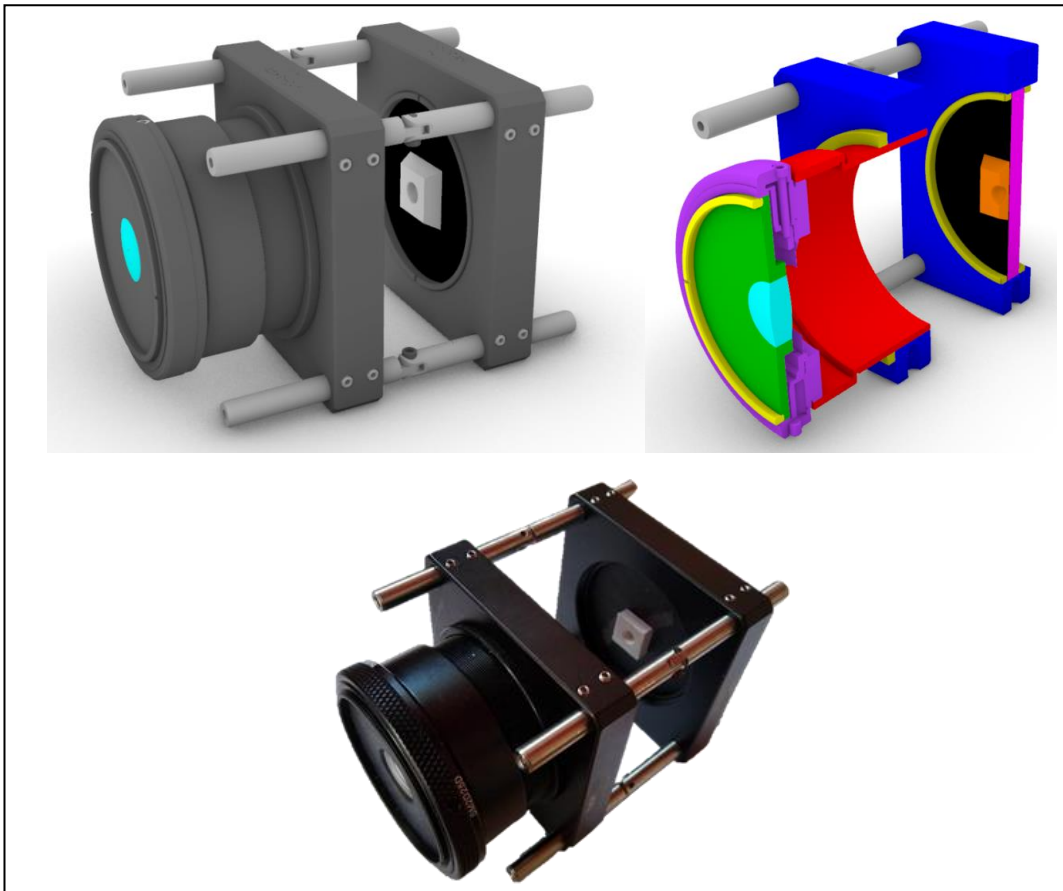


Figure 73: 3D CAD rendered views and photographs of the eye phantom. ***TOP LEFT****: 3D CAD rendered view of eye phantom.* ***TOP RIGHT****: 3D CAD rendered half-sectional view of eye phantom with colour coding to allow identification of the different components:* ***cyan*** *– achromatic doublet lens,* ***blue*** *– lens mount,* ***purple*** *– ring-actuated iris diaphragm,* ***red*** *– lens tube,* ***blue*** *– cage plates,* ***yellow*** *– retaining/locking ring,* ***grey*** *– swivel rod,* ***magenta*** *– steel disc,* ***orange*** *– ONH target.* ***BOTTOM****: photograph of eye phantom.*

7.2.3 Mechanical Design and Fabrication

7.2.3.1 Main Structure

In the following description, the colour-coded half-sectional view is referred to in order to point out the various components of the eye phantom. The lens of this phantom (cyan) was mounted to a ring-actuated iris diaphragm (purple) using a 3D printed mount (green) that was printed using black PLA+ filament on the Wanhao Duplicator i3 3D Printer. The mount held the lens such that the distance between the posterior surface of the lens was fixed at approximately 1 mm from the blades of the iris diaphragm. A retaining ring (Thorlabs, Newton, New Jersey, US), shown in yellow, locked the 3D printed mount in position on the diaphragm. The stop diameter of the iris diaphragm could be adjusted between 1 mm and 25 mm, allowing for a size range well beyond the typical dilated and undilated human pupil sizes. The diaphragm was screwed onto the end of a compatible lens tube (SM2V10, Thorlabs,

Newton, New Jersey, US), shown in red, which was itself screwed into a compatible cage plate (LCP01/M, Thorlabs, Newton, New Jersey, US), shown in blue, and its position locked using a locking ring (Thorlabs, Newton, New Jersey, US), shown in yellow. The lens tube allowed fine adjustments to be made to the distance between the lens and the retina of the eye phantom to obtain focus. A second cage plate (blue), which was used to hold the ONH targets (orange) in place, was connected to the first cage plate using swivel connector rods (C3A, Thorlabs, Newton, New Jersey, US), shown in grey. This formed an open back design so that the ONH targets could be directly seen from the outside. A mild steel disc (B.E Jenkinson Ltd, Lincolnshire, UK), 50 mm in diameter and 5 mm thick, shown in magenta, was mounted inside the second cage plate and secured using retaining rings. The flat face of the disc that faces the inside of the eye phantom was covered with black card (Hobbycraft Trading Limited, Dorset, UK), shown in black, secured using Original Pritt Glue Stick (Pritt, Düsseldorf, Germany), to mitigate reflections from this component. ONH targets were secured to this plate magnetically by means of four neodymium magnets (Magnet Expert Ltd, Tuxford, UK), each 2 mm in diameter and 1 mm thick, on their back surfaces.

7.2.3.2 ONH Targets

The ONH targets created had the exact same design as those described in the previous chapter, in subsection 6.2.2, but were instead painted white (titanium white heavy body acrylic paint) without the red speckle texture, and had the magnets embedded into their back surfaces. The two targets created are shown in Figure 74. As was the case with the previous targets, 4 ink marks were made near the bottom of the dimple and 4 on the top surface, to act as anchor points for referencing the reconstructions to the ground truth data.

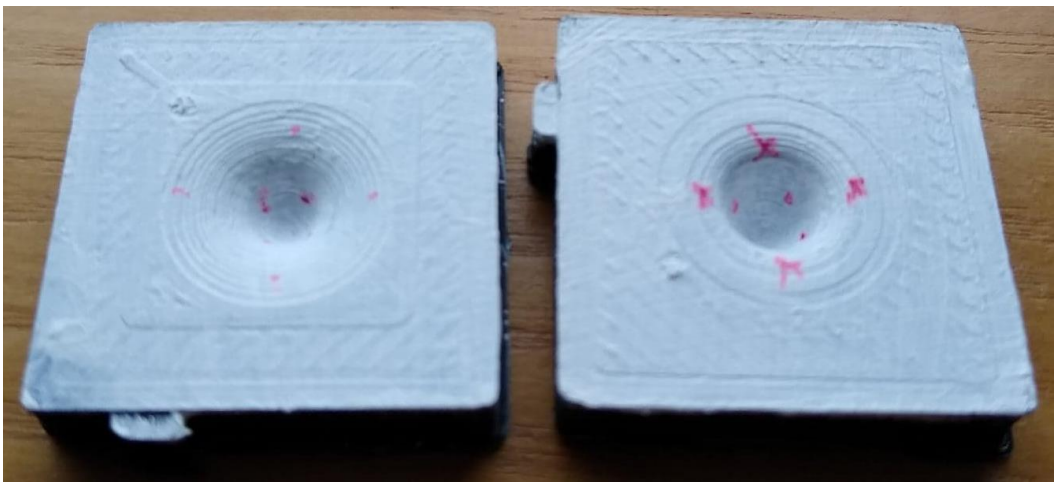


*Figure 74: Photograph of the normal (**LEFT**) and glaucomatous (**RIGHT**) ONH targets.*

7.2.3.3 Calibration Target

The calibration target used in this work was the same as that used in the work detailed in the previous chapter (in subsection 6.2.3), containing 10 x 7 squares in its pattern.

7.3 Upgraded Slit Lamp System

7.3.1 Brief Overall Description

A photograph, basic schematic and colour coded 3D CAD rendered view, with some parts sectioned to reveal internal components, of the add-on component fitted to the slit lamp, are shown in Figure 75. The assembled system is composed of the same optical/imaging components (slit lamp lens and cameras) as the system detailed in the previous chapter but with the addition of a projection component to allow texture to be projected onto the ONH targets.

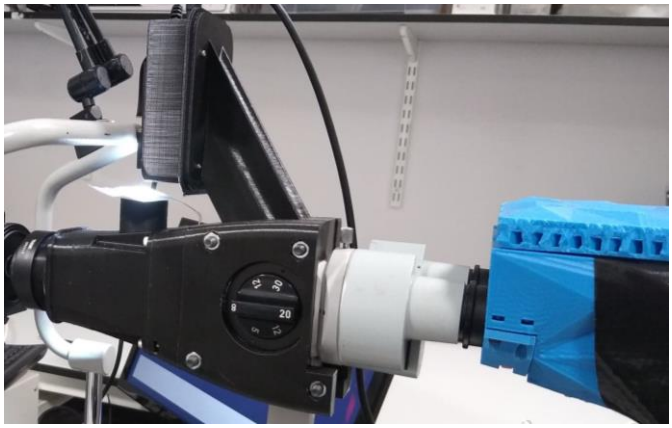
7.3.2 Design and Fabrication

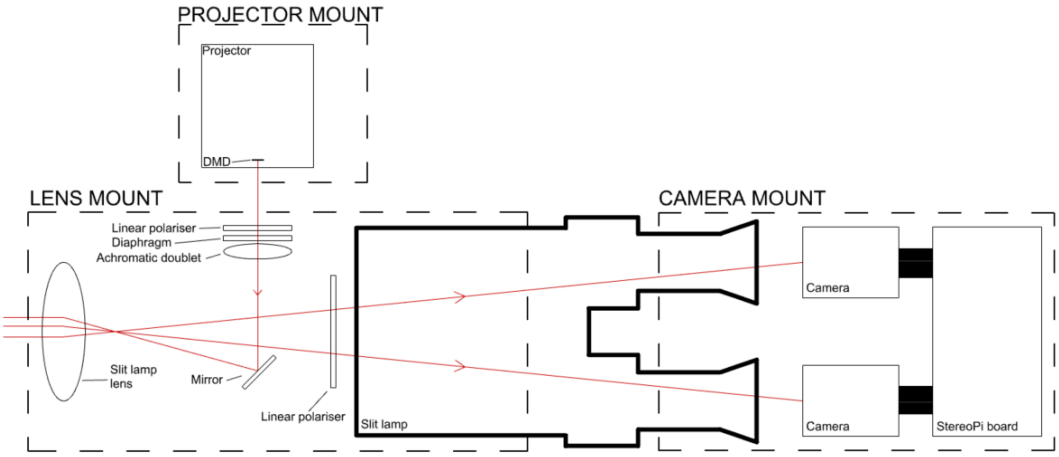
7.3.2.1 Slit Lamp Lens Mount

The slit lamp lens (90 D SuperField) was held in the same position as in the previous versions of the system but mounted using lens tubes and a different 3D printed main body. The lens was mounted inside an adjustable lens tube (SM2V10, Thorlabs, Newton, New Jersey, US) using a 3D printed mount, shown in black in the colour-coded CAD view in Figure 75. This lens tube was screwed into a stackable lens tube (SM2L10, Thorlabs, Newton, New Jersey, US) which was press fit into the 3D printed main body (shown in orange). Screwing and unscrewing the adjustable lens tube, containing the slit lamp lens, allowed for the focus to be finely adjusted. The 3D printed main body was clamped to the main block of the slit lamp and tightened using nuts and bolts. All 3D printed parts of the add-on components used in the work of this chapter, apart from the camera module, were printed in ecoPLA (3DJake, Paldau, Austria) on the Ultimaker 2+ 3D printer.

7.3.2.1 Cameras

The cameras, controlling board, and 3D printed mount, are the same as those used in the work detailed in the previous chapter (presented in subsubsection 6.3.1.2).





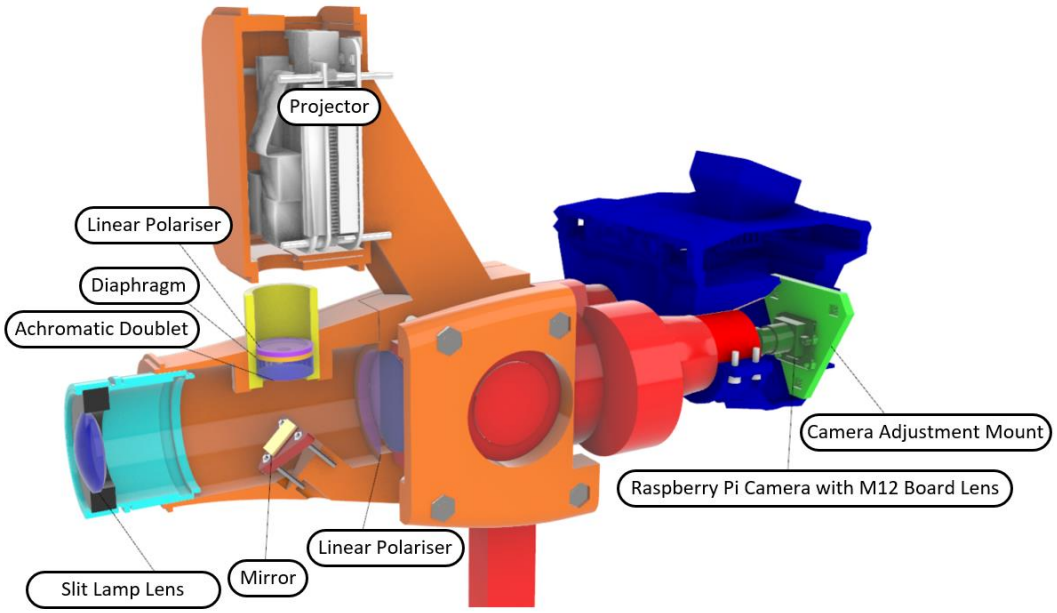


*Figure 75: Photograph (**TOP**) with corresponding schematic (**MIDDLE**) and colour-coded and labelled CAD rendered 3D model (**BOTTOM**), with some parts cut to reveal internal components, of the slit lamp fitted with add-on components (Chapter 7). **Blue** – 3D printed camera mount; **Red** – Zeiss 30 SL-M slit lamp; **Orange** – Slit lamp lens and projection components mount; **Cyan** – Lens tubes; **Black** – Slit lamp lens mount.*

7.3.2.1 Projection Component

The vertically mounted projection system features a small projector, relay lens, diaphragm, mirror to direct the projection towards the slit lamp lens (and therefore eye), and linear polarisers to mitigate reflections. All components were mounted onto the slit lamp lens' mount, apart from the projector, which had its own 3D printed mount, both shown in orange in Figure 75. A schematic of the geometrical optical design of the projection system, where lens thickness was ignored, can be found in Figure 76, showing the spacing between the individual components. The projector used was a DLP LightCrafter Display 2000 Evaluation Module (Texas Instruments Inc, Dallas, Texas, US). The image formed at its digital micromirror device (DMD) was relayed to the plane of the aerial image of the retina, located behind the slit lamp lens, as can be seen in Figure 76. An achromatic doublet lens (53DC22, Comar Optics Ltd, Linton, UK) with a focal length of 53 mm, was used to do so (the projector's stock optics were removed). It was noticed that a portion of the DMD was blocked by a mirror located within the projector's optical block when projected onto the retina and only a roughly square area, with a side length equal to the height of the DMD, of the DMD could be seen. The geometrical optical design was created such that the height of the DMD (2.72 mm) would conjugate to a 7.55 mm height on a human retina (~24°), assuming the effective focal length of the eye is 16.7 mm (Atchison & Smith, 2000).

The achromatic doublet lens, diaphragm and one of the linear polarisers (Comar Optics Ltd, Linton, UK), were held on a 3D printed mount, shown in yellow in Figure 75. This mount could be raised or lowered, to enable focusing of the projection system, and rotated, to allow for the direction of the linear polariser to the adjusted. The diaphragm, also 3D printed, contained a 3 mm aperture, and was included to reduce the brightness of the projection on the retina. The mirror (16MX10, Comar Optics Ltd, Linton, UK) was secured onto a 3D printed triangular adjustment piece using Loctite 4305 UV Curable Adhesive. This piece, which allowed for adjustment of the tilt of the mirror, was secured to the slit lamp lens' main mount using bolts, with springs on them, placed between the adjustment piece and the main mount, to maintain extension. A second linear polariser was secured, using a 3D printed mount, between the mirror and the slit lamp's objective lens, to complement the other linear polariser. In order to control the projector and set the patterns to project, a BeagleBone Black Rev C board (Texas Instruments Inc, Texas, US) with the Debian Linux operating system installed was connected to it.

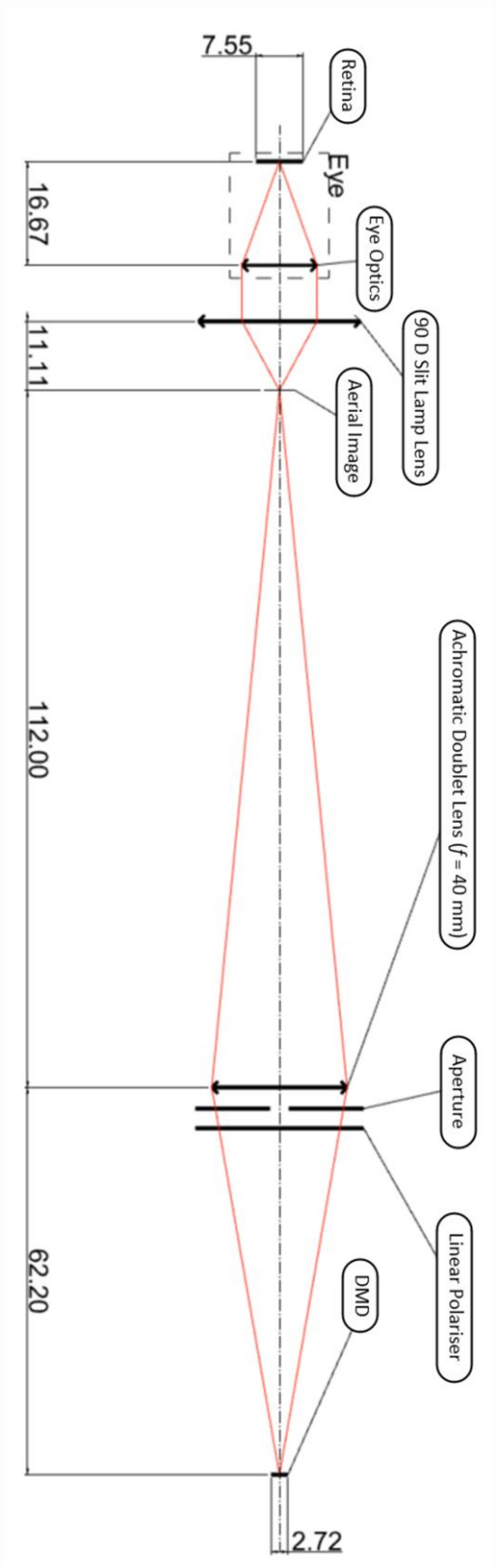


Figure 76: Schematic of the optical design of the illumination system upgraded sit lamp system (chapter 7).

7.4 Performance Evaluation

7.4.1 Brief Overall Description

The slit lamp with fitted add-on components was used to capture the required images of the ONH targets, while a speckle pattern was projected onto their surfaces, and calibration target in the eye phantom. These images were then processed using a stereo vision pipeline. Following this, quantitative validation against ground truth data, in the same way as was done in the work of the previous chapter, was performed.

7.4.2 Trial Setup and Image Capture

7.4.2.1 Setup

The experimental setup is not shown since it is very similar to that shown in the previous chapter: the eye phantom was held on an aluminium frame in front to the slit lamp lens of the imaging system. The cameras were fixed at the same focus as before (to infinity), the slit lamp set to x30 magnification, the eye pieces set to their zero-dioptre setting and the slit lamp lens at a position such that the imaging system was infinity focused. Once the imaging system was focused, the eye phantom was focused such that a target on its retina was in focus to the imaging system (i.e., it was emmetropic). The diameter of the diaphragm's aperture was previously set to be 8 mm, using a set of ABSOLUTE Digimatic Calipers. The room was darkened in preparation for focusing of the projection system and imaging. The projector was set to project a green speckle pattern, which is depicted in Figure 77. This pattern was chosen after promising results with it were achieved during unreported

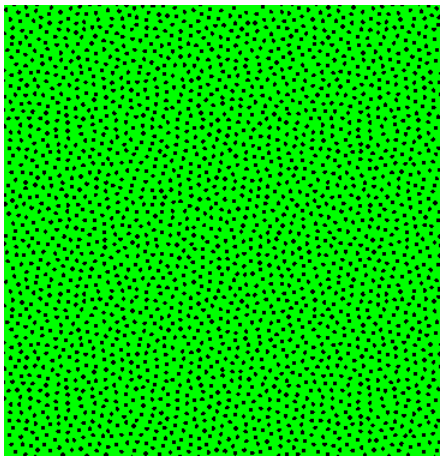


Figure 77: Image of the pattern projected on to the surface of the ONH targets in this performance trial (Chapter 7).

development trials where several different patterns were tested (striped and speckle noise). The mirror's tilt was adjusted in order for the projection to cover the ONH structures. The vertical position of the achromatic doublet lens, of the projection system, was adjusted to optimise the projectors focus on the ONH target's surface. The linear polariser above it was rotated until reflections were maximally suppressed.

7.4.2.2 Image Capture of the ONH Targets

Stereo images were acquired of the ONH target already inside the phantom, with the projection set to the speckle pattern, and then again with the projection set to a blank white image. The target was then rotated 90° around the phantom's optical axis and this was repeated. This was repeated two more times, such that the target had been imaged at four different rotations. Following this, the other target was imaged in the same way.

7.4.2.3 Image Capture of the Calibration Target

Following this, the images necessary for calibration of the system were acquired. In order to do so, the back cage plate of the phantom was removed, providing enough space for the calibration component to be held in the appropriate position. The calibration pattern was then imaged in the same way as in the previous chapter (subsubsection 6.4.2.3), until 100 stereo image pairs were obtained.

7.4.3 Stereo Vision Pipeline

7.4.3.1 Calibration and Rectification

The procedures followed to perform calibration and rectification were the same as those carried out in the previous chapter (detailed in subsubsection 6.4.3.2) but with one difference. In this case, the calibration was run using the full 100 images but then those images which contained a point for which the x or y component of its reprojection error exceeded 1.5 pixels were removed. This led to 29 images being removed. Calibration was therefore performed using 71 images. In order to see how the values of the coefficients and parameters vary with the number of images, the calibration was also run using 7 to 63 images, in increments of 7 images.

7.4.3.2 Stereo Matching

The stereo matching algorithm was applied to the images of the ONH targets which were taken with the speckle pattern projected. The corresponding images obtained while the

projection was white, were not used in stereo matching but used later as the texture overlay. The stereo matching algorithm used here was a purely local method. The algorithm found points to match by finding the locations of individual speckle dots of the projection, and then performed matching of patches over these. The images were converted to grayscale prior to matching. The stereo matching algorithm followed the following procedure:

A. *Initial course disparity estimation:*

In precisely the same manner as is detailed in the previous chapter (step A, subsubsection 6.4.3.3), a region of interest was selected and an initial disparity value for that whole region was obtained, using the template matching function of the OpenCV library.

B. *Search for patches to be matched:*

The speckle dots within the projection pattern were searched for, over the whole region of interest, using the template matching function of the OpenCV library. The template used was a copy of a small cropped out area from the left image, containing a single speckle dot. A threshold value was set, which could be tuned, to return patches which were most similar to the template. An example, showing the region of interest with the speckle dots within it marked with bounding boxes, is shown in Figure 78. The patch sizes are all at a fixed size equal to the size of the initial speckle dot template.

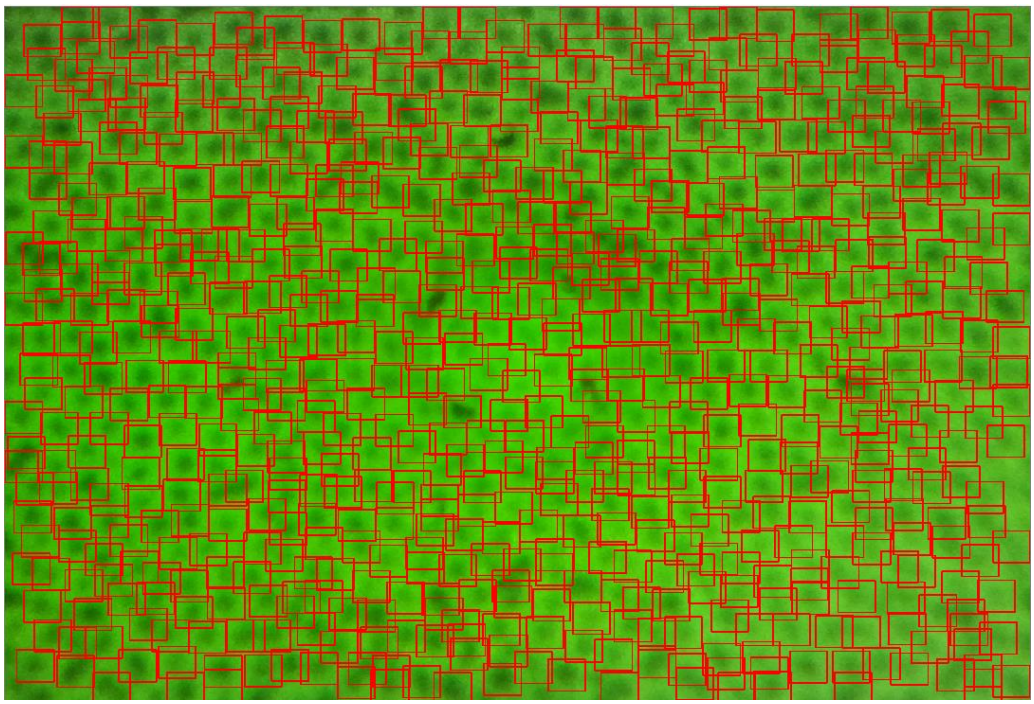


Figure 78: Speckle dots of the projected pattern marked with bounding boxes.

C. *Apply texture threshold:*

As was the case in the previous chapter, the SD of a normalised version of the patch to be matched was determined. The threshold set in this case was slightly higher, at 0.35. If the patch gives a value lower than this, then the algorithm skips to stage G.

D. Forward and backward matching:

The same method for matching in the forward (left to right) and backward (right to left) directions, as was detailed in the previous chapter, was used to yield disparity values in both directions.

E. Apply a consistency threshold:

If the disparity values determined in the forward and backward matching stages differ by more than 0.9 pixels, then the algorithm skips to stage G.

F. Store disparity result:

It the algorithm succeeded through stages C and E, then the disparity value was set to be the average of the values determined in the forward and backward matching directions and stored for the final sparse disparity map.

G. Perform matching for all patches:

Stages C to F are repeated for the next patch, until all patches have been processed. The result is a sparse disparity map.

7.4.3.3 Reprojection, Interpolation and Texture Addition

In the work of this chapter, the sparse disparity map estimated from stereo matching was reprojected straight away, before interpolation, using the same equations as were detailed in the previous chapter (in subsubsection 6.4.3.5).

Interpolation was carried out in a different way in this case since the previous algorithm, which used smoothing splines, only works for disparity maps where the points fall on the same rows and columns, which is not the case here since the initial points were not on a regular grid. In this case therefore, a mesh was fit to the sparse point cloud, which was later used to generate a dense textured point cloud. In order to estimate the mesh, the surface normals had to be estimated first. This was carried out using the normal estimation feature of the Point Cloud Library (Rusu & Cousins, 2011). The accuracy of the resulting mesh

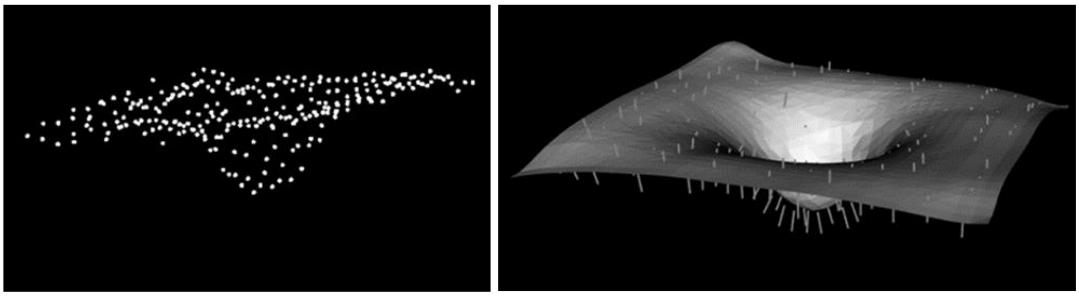


*Figure 79: Mesh, with normals shown, (**RIGHT**) and corresponding sparse point cloud (**LEFT**).*

appeared to be highly dependent on the search radius term of the normal estimation feature. This had to be tuned to suit the scale and density of the sparse point cloud. The normals returned were often not, in this case, oriented in the same direction (i.e., some were directed inside the ONH structure while some were oriented away). These were flipped, if needed, towards a distant point so that they all were oriented inside the ONH structure to avoid problems during the mesh fitting procedure. Taking the sparse point cloud and surface normals as inputs, a mesh was fit using the Poisson based surface reconstruction feature of the Point Cloud Library. An example of a mesh, fit in this way, is shown in the right image in Figure 79. In this image the estimated normals are also shown. The sparse point cloud to which the mesh was fit is shown in the left image of Figure 79.

Following this, a dense textured point cloud was estimated using the mesh. This was achieved by defining a ray for the reprojection of every pixel in the left image, within the selected region of interest, and determining where each ray would intersect the mesh. For every ray, this required computing, for each triangle in the mesh, whether or not that ray intersected it. Once a triangle was found to be intersected by a particular ray, the intersection coordinates were recorded and the RBG data of the originating pixel were assigned to it. This process, carried out on all pixels, yielded a dense textured point cloud. It must be noted that the image reprojected was in fact the left image obtained while a white projection was projected on it. An example of one of the dense textured point clouds obtained in this work is provided in Figure 80. This method was very computationally expensive since every triangle had to be tested for every ray. It must be noted however that once an intersection was found for a particular ray, further testing for that ray was skipped and the next ray processed, but the process still often took a couple of minutes to complete. In light of this, in this work the process was carried out for every second pixel in the x and y directions. The C++ code written to perform rectification, stereo matching, interpolation and texture addition is provided in Appendix 7A.

Figure 80: Example of one of the dense textured point clouds obtained in this work.

7.4.4 Quantitative Validation

7.4.4.1 Ground Truth Data Acquisition

The ground truth scans of the ONH targets were acquired in the same manner as detailed in the previous chapter (in subsubsection 6.4.4.1), using a professional photogrammetry software package, but speckle patterns were created, using pencil lead, on their top surfaces, since the targets were practically textureless. The use of pencil lead meant that the red ink marks were still clearly visible, allowing them to still be used as anchor points for alignment. Images of the targets after applying such texture are provided in Figure 81.

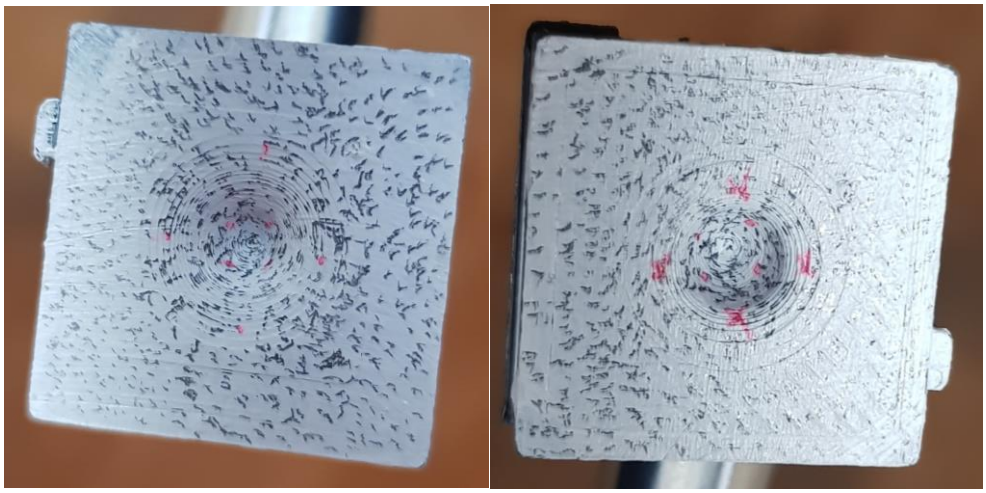


*Figure 81: Images taken of the normal (**LEFT**) and glaucomatous (**RIGHT**) ONH targets, showing the speckle pattern applied, for ground truth scanning.*

7.4.4.2 Quantitative Comparison

The forms of quantitative comparison performed were the same as those described in the previous chapter (in subsubsection 6.4.4.2): through measurements and a cloud-to-cloud comparison. Measurement 1 was made between points A and B, and measurement 2 was made between points C and D, according to the labels shown in Figure 82.

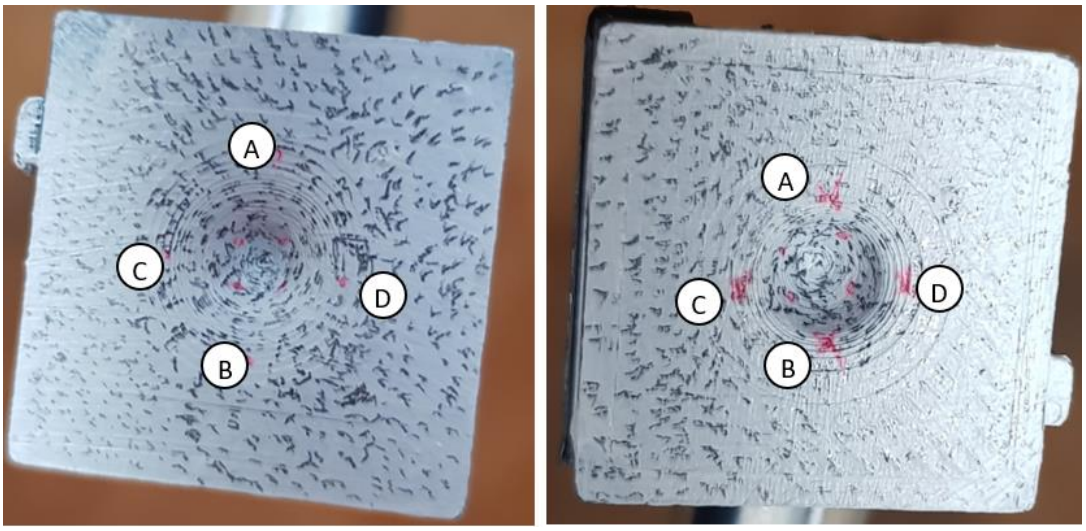


Figure 82: Marks where measurements were made between on the targets' surfaces. ***LEFT****: Glaucomatous ONH target.* ***RIGHT****: Normal ONH target.*

7.5 Results and Discussion

7.5.1 Calibration Performance

7.5.1.1 Reprojection Error

Reprojection error plots from the initial calibration with all 100 images, and the calibration using 71 images after bad images were removed, are shown in Figure 83. It can be seen that the outlier points are much reduced after bad images were removed. This is also highlighted in the overall mean reprojection error. The value before was 0.74 pixels, whereas after removal was 0.64 pixels. It is still not clear whether or not this level of error is sufficiently low for glaucoma diagnostics however it represents an improvement on the value obtained for the calibration in the work of the previous chapter, which was 0.88 pixels.

7.5.1.2 Estimated Coefficients and Parameters

The graphs obtained, showing each of the calibration values and associated uncertainties (3 times the SD) obtained using an incremental number of calibration images, are shown in

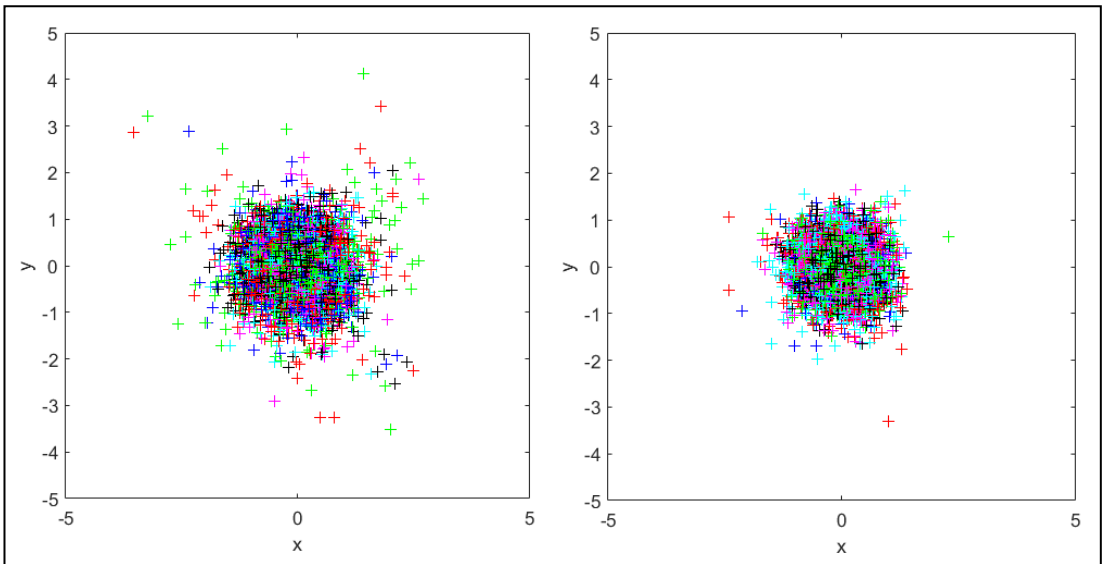


*Figure 83: Reprojection error plots from the calibration before (**LEFT**) and after (**RIGHT**) removing bad images.*

Table 8: Calibration coefficients and parameters estimated for the system. SD: Standard Deviation.

	Coefficient/Parameter	Value	Uncertainty (3SD)	% Uncertainty
Left	f_x (pixels)	7892	34	0.43
	f_y (pixels)	7868	32	0.41
	C_x (pixels)	1980	10	0.51
	C_y (pixels)	560	11	1.96
	k_1	-1.47	0.02	-1.36
	k_2	18.4	0.737608	4.01
	k_3	-94.7	5.6	-5.91
	P_1	0.0019	0.0010	52.63
	P_2	0.0142	0.0008	5.63
Right	f_x (pixels)	7563	32	0.42
	f_y (pixels)	7648	32	0.42
	C_x (pixels)	512	15	2.93
	C_y (pixels)	275	11	4.00
	k_1	-0.97	0.02	-2.06
	k_2	5.4	0.2	3.70
	k_3	-13.9	0.9	-6.47
	P_1	-0.0093	0.0006	-6.45
	P_2	0.0182	0.0010	5.49
	R_1	-0.053	0.002	-3.77
	R_2	0.157	0.003	1.91
	R_3	0.031	0.0002	0.65
	T_x (mm)	2.220	0.014	0.63
	T_y (mm)	0.200	0.013	6.50
	T_z (mm)	0.194	0.078	40.21

Appendix 7B. It was seen in these graphs that all values practically stabilised by 71 images, indicating that more calibration images of the same quality would not have led to very different results. The values and associated uncertainties, expressed in absolute and relative percentage terms, of each calibration coefficient/parameter, determined using 71 images, are tabulated in Table 8. It can be seen that it is again the case that the distortion coefficients tended to have the greatest uncertainties associated with them, especially when compared to the intrinsic parameters. It does seem to be the case that the distortion maps, shown in

Figure 84 and Figure 85 for the left and right cameras respectively, appear more similar to that which would be expected from a typical stereo camera system, compared to the work of the previous chapter. In particular, the principal points are both within the image. However, they are still quite some distance from the image centres.

As was also the case in the work of the previous chapter, the y and z translation parameters had relatively large uncertainties associated with them. It is not entirely clear why this is the case and how they could be reduced.

7.5.1.3 Potential Sources of Calibration Error

The potential sources of calibration error in this case are unchanged from those detailed in the previous chapter (subsection 6.5.1): corner localisation errors, inaccurately fabricated checkerboard, inaccurate checkerboard square size estimation, sub-optimal camera alignment and inadequacy of distortion models.

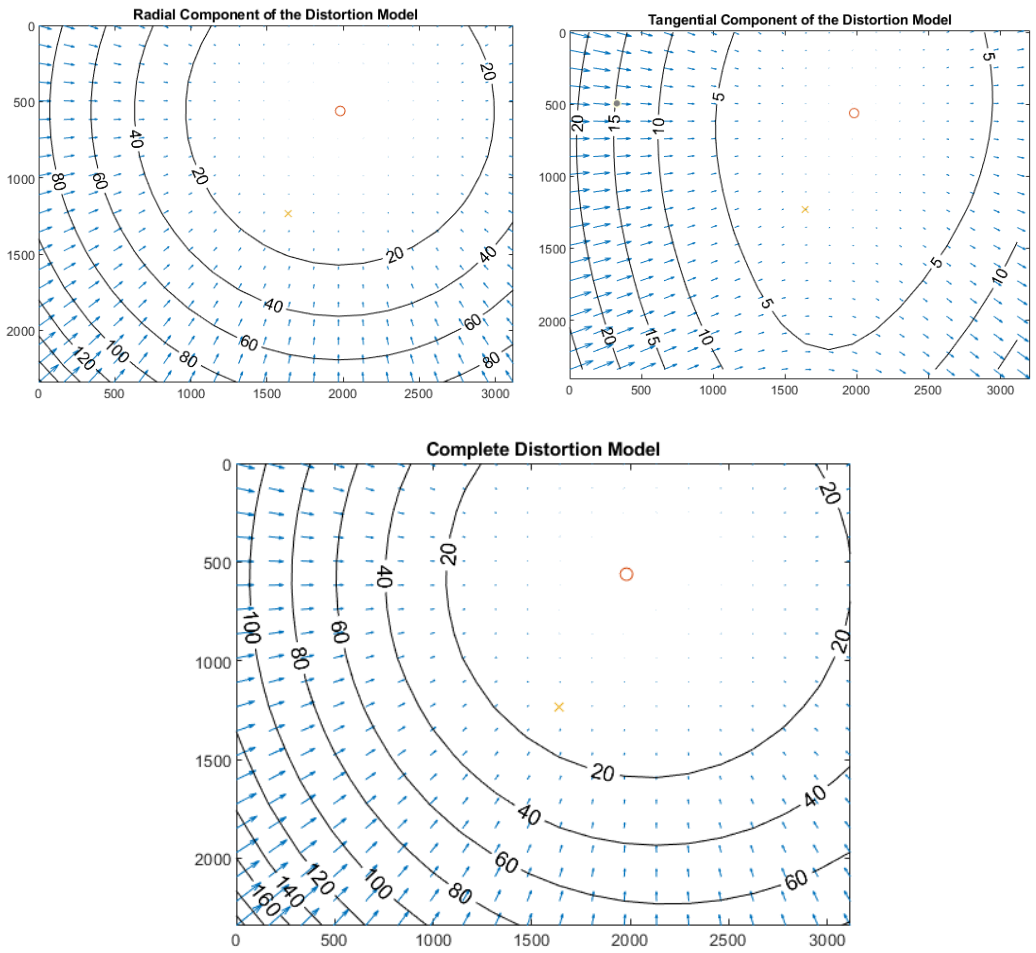


Figure 84: Distortion maps to visualise the distortion parameters estimated for the left camera.

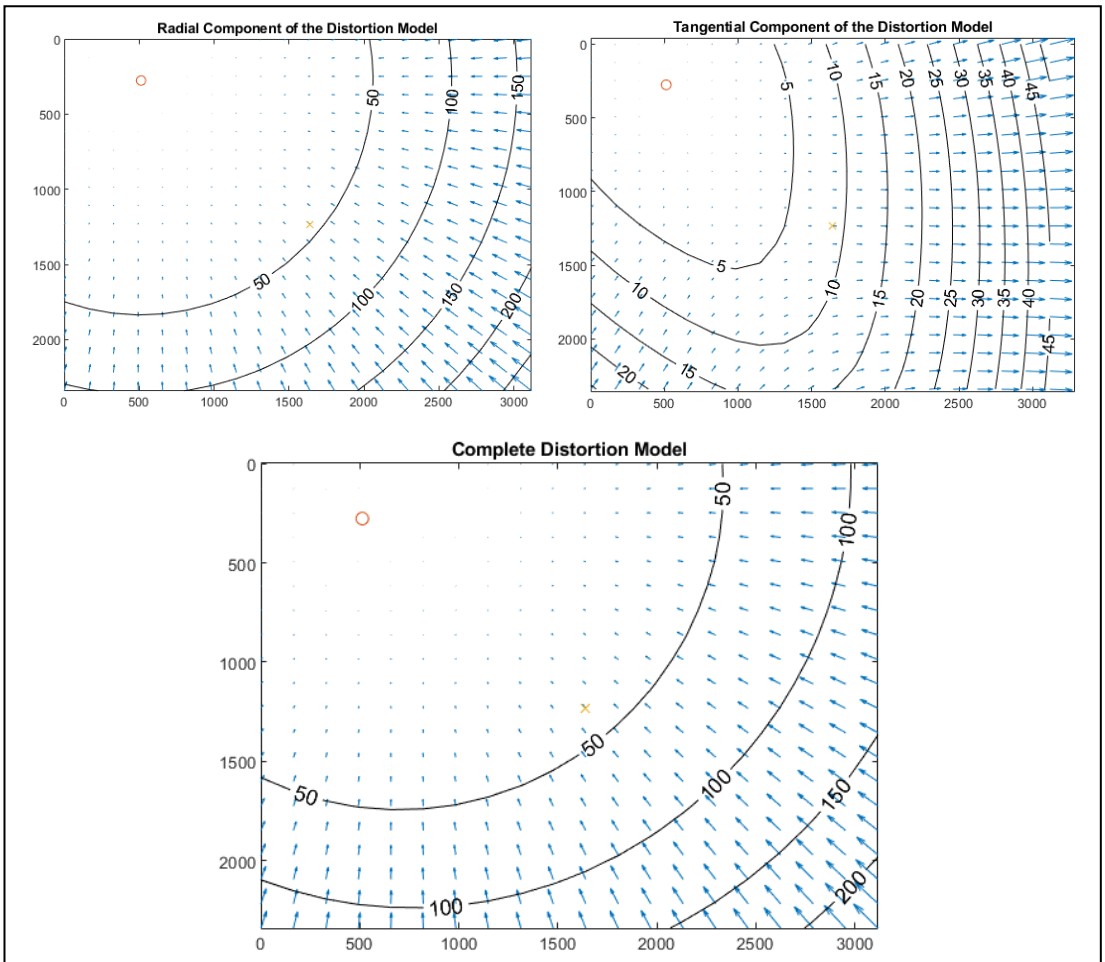


Figure 85: Distortion maps to visualise the distortion parameters estimated for the right camera.

7.5.2 Reconstruction Performance

7.5.2.1 Results

The reconstructions obtained of the normal ONH target, with corresponding stereo image pairs, and the ground truth data, are shown in Figure 86. The same for the glaucomatous ONH target are shown in Figure 88. The measurements made for each of the targets, on their reconstructions and ground truths, are tabulated in Table 9. Colour error maps, showing the accuracy and precision of the reconstructions obtained for the normal and glaucomatous targets are shown in Figure 87 and Figure 89, respectively.

Table 9: Reconstruction and ground truth measurements. SD: Standard Deviation.

Target	Type	Measurement 1	Measurement 2
Normal	Reconstruction	Mean = 9.58 mm (SD = 0.05)	Mean = 8.27 mm (SD = 0.04)
	Ground Truth	9.55 mm	8.22 mm
Glaucomatous	Reconstruction	Mean = 7.02 mm (SD = 0.06)	Mean = 7.87 mm (SD = 0.08)
	Ground Truth	6.97 mm	7.78 mm

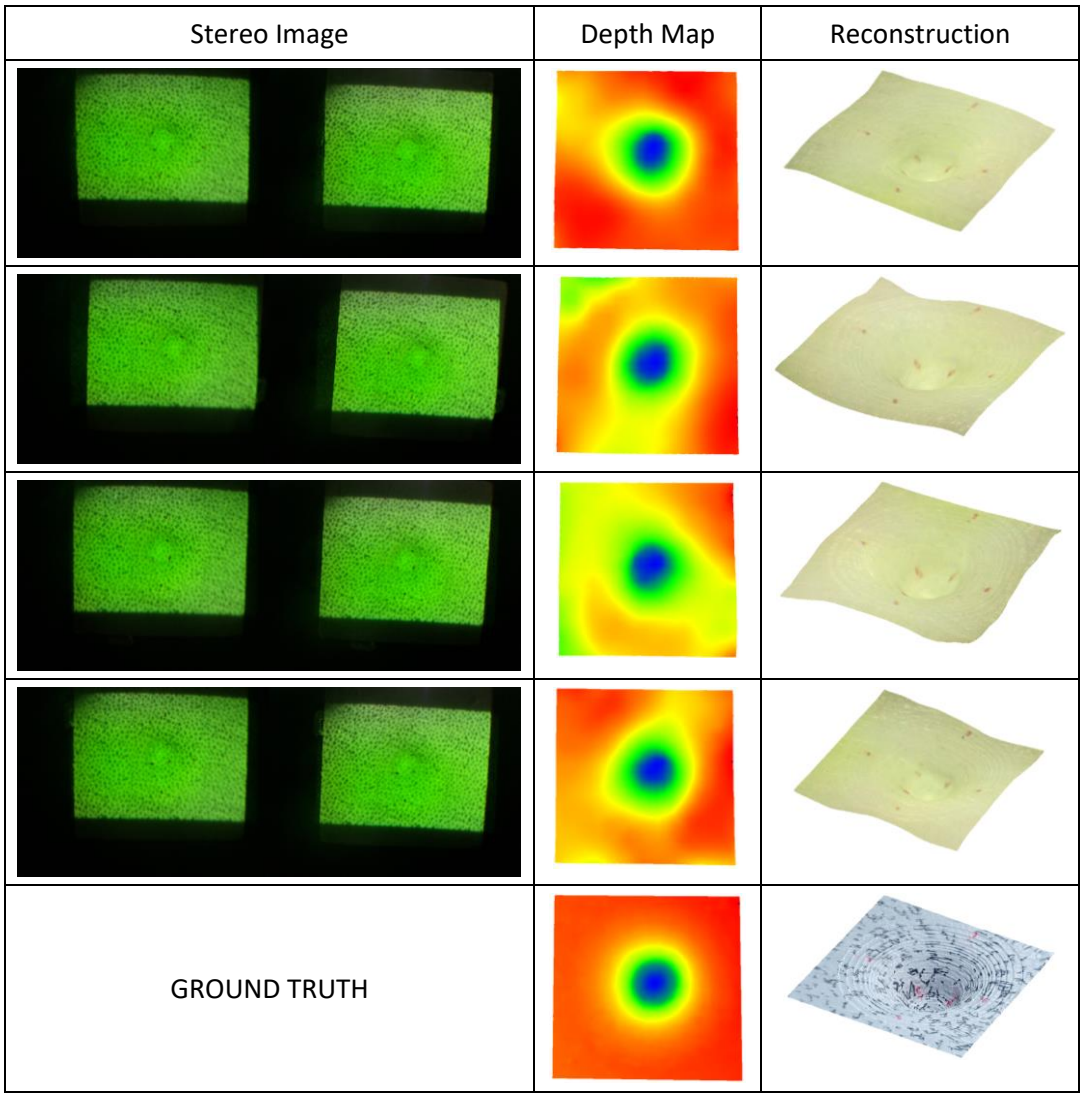


Figure 86: Reconstructions obtained of the normal ONH target, alongside the ground truth data.

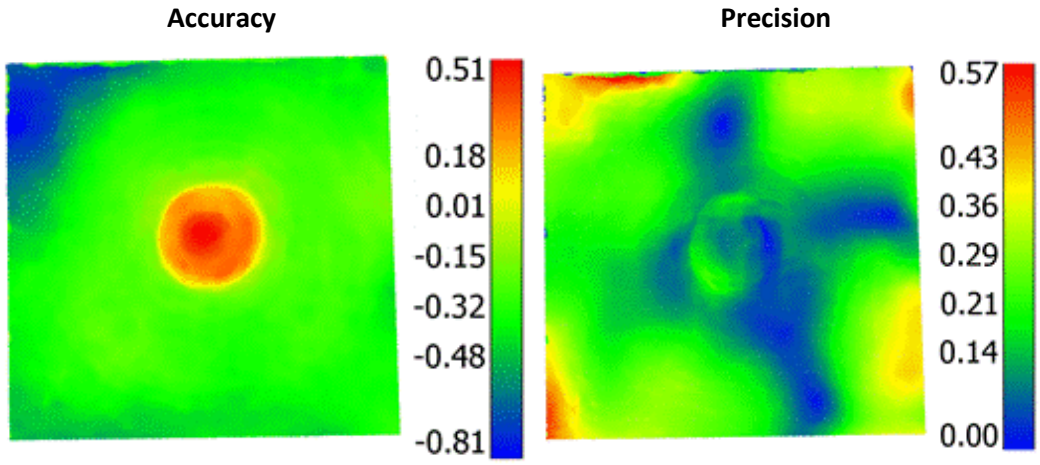


Figure 87: Accuracy (mean of distance errors) and precision (standard deviation of distance errors) maps of the reconstruction of the normal ONH target (Chapter 7). Values are in units of millimetres.

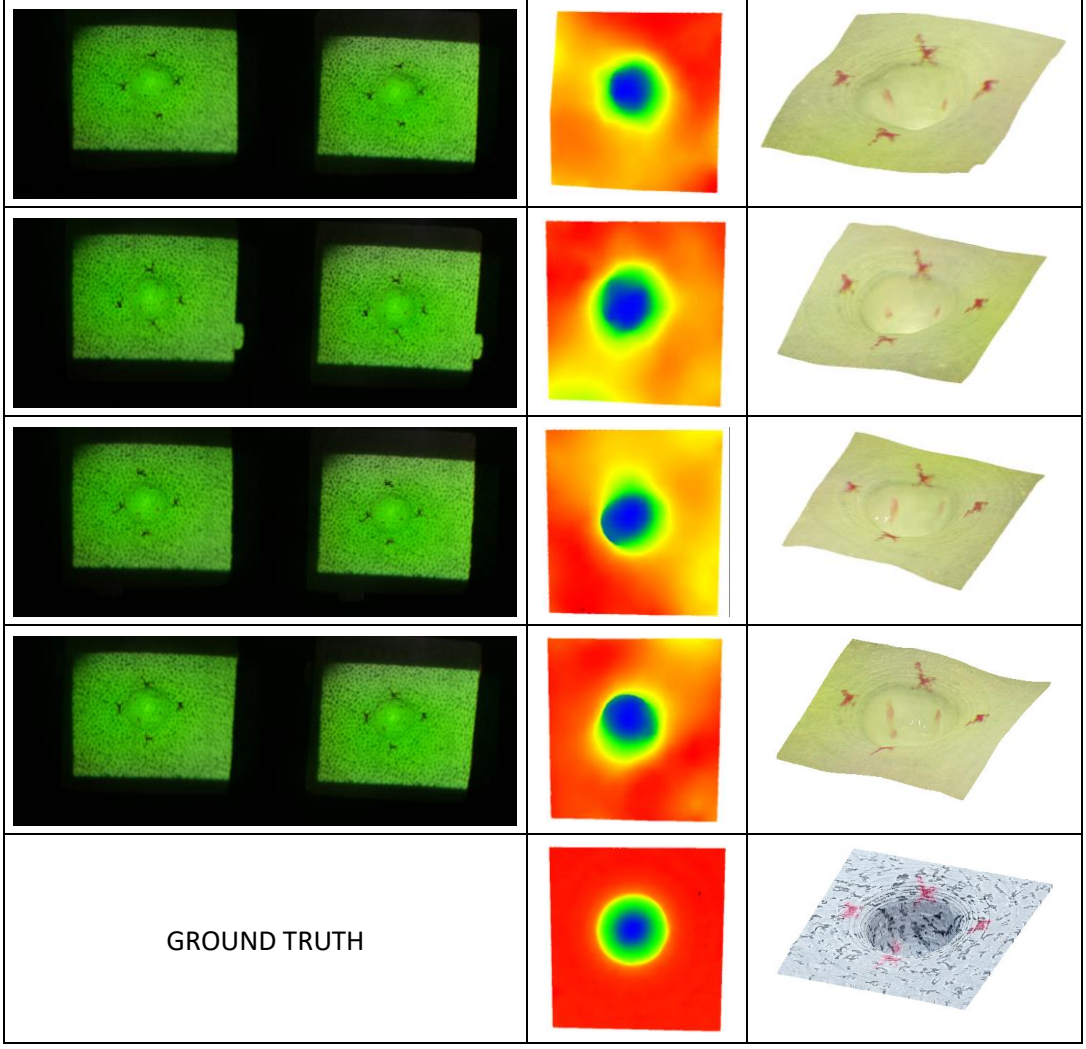


Figure 88: Reconstructions obtained of the glaucomatous ONH target, alongside the ground truth data.

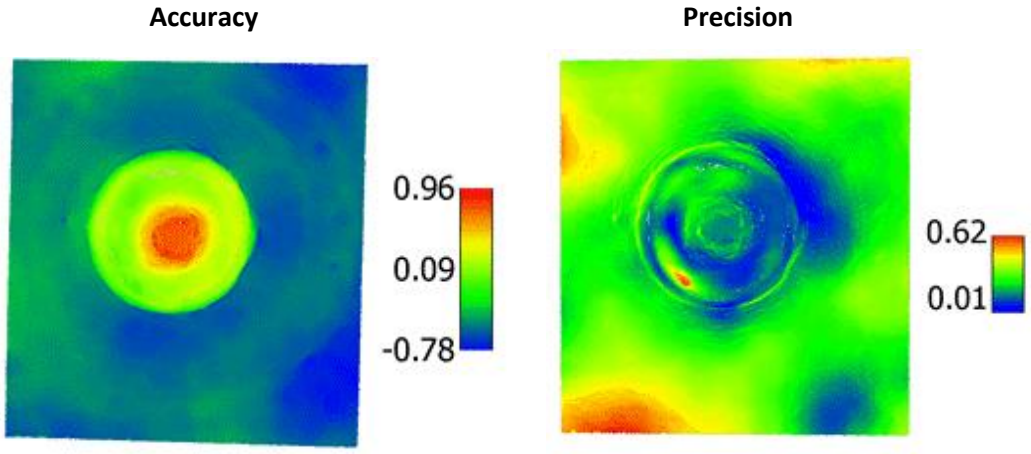


Figure 89: Accuracy (mean of distance errors) and precision (standard deviation of distance errors) maps of the reconstruction of the glaucomatous ONH target (Chapter 7). Values are in units of millimetres.

7.5.2.2 Discussion

Measurements: The values obtained for each of the measurements indicate that the diameters measured on the reconstructions were close but, as was the case in the work detailed in the previous chapter, slightly overestimated in both cases. The SD values seen in the table appear low relative to the measurement values, indicating that the reconstructions were in good agreement with each other. There are a number of potential reasons for the discrepancies, including reconstruction error (as discussed below) and measurement error. Measurement error could be reduced by creating better anchor points. It may be that smaller circle marks would have allowed better localisation of the points. Moreover, each measurement on each point cloud could be repeated and an average taken.

Reconstructions and Cloud-to-Cloud Comparison: Looking at the depth maps obtained of the normal ONH target, it appears that the reconstruction is relatively similar to the ground truth but has some errors. The ground truth shows perfectly flat peripheral regions, which is not the case for the reconstructions – they appear to not be flat and have abnormal edges, which was also the case in the work of the previous chapter. A similar trend is seen in the depth maps of the glaucomatous ONH targets. In these it also appears that the bottom of the dimple structure was flatter in the reconstructions compared to the ground truth – the blue area is larger. The maps do show a more circular profile than those obtained in the work of the previous chapter. This could be down to the calibration potentially being better in this case since the reprojection error was lower, and/or stereo matching performance. In the work of the previous chapter, it could be seen in the accuracy maps that the anchor points were causing abnormalities in the reconstruction, which does not appear to be the case here, which could also explain why the reconstructions here appear to have a more circular profile. This could also explain why the accuracy maps obtained appear cleaner than in the previous chapter.

On examining the accuracy maps, it appears that the depths of the dimple structures were not well captured, particularly in the case of the glaucomatous ONH target. The precision maps show, as was the case in the previous chapter, that the reconstructions taken at different angles have differences. This work still did not permit the precision of a target not rotated but simply removed and replaced, with the same rotation and position, to be determined. It would be interesting to do so, since it would give some indication on whether calibration or stereo matching was responsible for the largest proportion of the errors seen.

In general, it appears that global to local errors are the main errors which are present. These likely come from the sources detailed below. The reconstructions obtained are smooth and free of stochastic noise, therefore noise has not been considered.

7.5.2.3 Main Sources of Reconstruction Error

Calibration: The calibration performance is likely a significant source of the large-scale errors seen but this needs to be more rigorously tested. Accurate calibration is vital for obtaining accurate reconstructions. Future work needs to better determine the impact that calibration is having on performance, separate from the other error sources, to see if it is something that is worth investing time on improving.

Projection and stereo matching: In this case where the texture was provided by a projector, since the targets were practically textureless, good projection quality was vital for good stereo matching performance. The fineness of the speckle pattern was set to be the best that the projection quality would allow. The quality was less than perfect, and the fineness of the pattern was likely less than sufficient for this work. Moreover, it appeared that the focus of the projector was better on the top surfaces of the targets and poorer at the bottom of the dimple structures. This is evident in the images in Figure 86 and Figure 88, where it can be seen that the dots are darker on the top surfaces of the targets and less bright at the bottom and sides of the dimples. Future work to improve projection quality, potentially through a redesign of the projection system, would likely benefit this work. The fact that the superficial layers of the human ONH are semi-transparent should be considered in such work. Given better projection quality, and a finer speckle pattern on the ONH, there will likely be an increase in accuracy seen in stereo matching. As is usually the case, the stereo matching algorithm could always be improved. A move to high performing global algorithms, which were used by Bansal et al (Bansal et al., 2013) and Zhang et al (Zhang et al., 2005), may enhance matching accuracy in this case. They may also allow for matching of a human ONH without the need for a projected pattern.

Image quality: On looking at the images taken where the projection pattern was white, it is evident that there were differences in image quality/focus in some regions. This is most apparent when looking at the anchor points. As examples, two anchor points, taken from the left and right images, are shown in Figure 90.

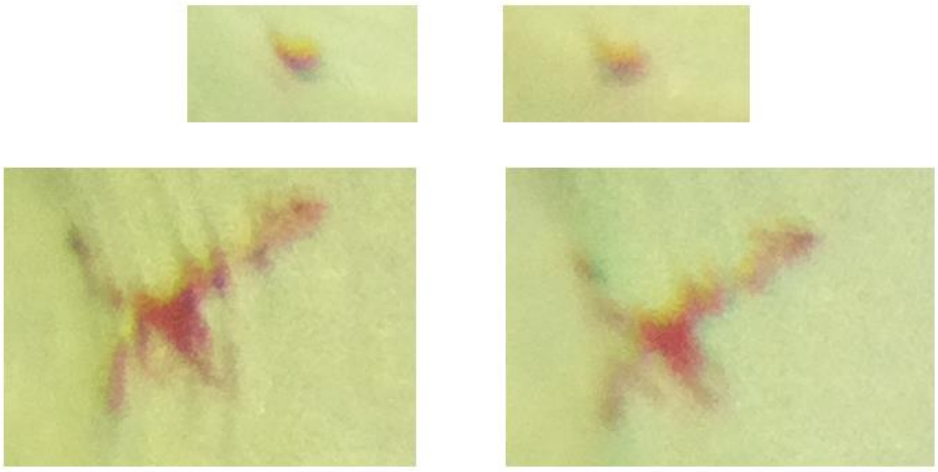


*Figure 90: Two different anchor points (**ON EACH ROW**) extracted from the left (**LEFT**) and right (**RIGHT**) images of one stereo image pair captured.*

7.5.3 Limitations of This Work

The limitations which were described in previous chapters, which also apply to this trial, are as follows:

- Optical properties of ONH targets [subsubsection 5.6.4.3]
- Eye phantom optics [subsubsection 5.6.4.6]
- Precision assessment [subsection 6.5.3]
- Shape of the ONH targets [subsection 6.5.3]
- Ground truth accuracy [subsection 6.5.3]
- Camera synchronicity [subsection 6.5.3]

7.6 Conclusion

The main target of this work was to adapt the system to enable it to perform reconstruction of practically textureless ONH targets, given that the human ONH contains large textureless regions. The system was successfully adapted, through the addition of a projection system, which allowed for texture to be projected onto the surface of practically textureless ONH targets, providing enough texture for a stereo matching algorithm to operate. Reconstructions were obtained, which captured the general overall structure of the targets but, as was the case in the work in the previous chapter, indicate that significant improvements can still be made to the system and algorithms. The issues which were

identified in this chapter and the previous two chapters, as well as their status, are provided in Table 10.

Table 10: Issues identified until this stage and their current status (chapter 7). ***Changes are shown in bold****.*

Issue Number	Issue	Status
1	Stereo matching technique resulted in quantisation effect	The coarse-to-fine stereo matching algorithm did not suffer from this issue. **Moreover, the algorithm used in this work also did not have this issue.**
2	Low image resolution	The cameras used here (Raspberry Pi Cameras) were of a much higher resolution than those used in the work of chapter 5. Resolution is likely no longer a significant limiting factor.
3	Calibration performance sub-optimal	The reprojection errors appear large and the distortion maps abnormal.
4	Unstable rectification results	The rectification process used here appears to give repeatable rectification outputs.
5	Poor focus in some areas of the images	This is still an issue at this stage.
6	Validation was only qualitative since reconstructions are uncalibrated	Reconstructions in this work were calibrated and quantitative validation was performed.
7	ONH target surface texture different to human ONH	**The texture density of the targets in this work was very low, in line with that which is seen in many regions of the human ONH.**
8	ONH target optical properties different to human ONH	The targets used were still opaque – different to the human ONH.
9	ONH target shape different to human ONH	The shapes of the targets in this work were more similar to the human ONH, than those used in the work of Chapter 5, but were rotationally symmetrical - different to the human ONH.
10	Illumination not provided from outside the eye	**Illumination was provided from outside the eye using the projector-based illumination system.**
11	Eye phantom optics different to human eye	The optics of the eye phantom used here were simple compared to the optics of the human eye.
12	Image capture is not synchronous	The synchronicity of the cameras used was not tested in this work.
13	Stereo matching performance sub-optimal	There are many settings for the stereo matching algorithm developed which were not optimised.
14	Repeated measures not representative of imaging a human eye	Precision is yet to be determined where the ONH targets are imaged in the same location in the field of view of the cameras.
15	Unclear ground truth accuracy and precision	Future work would benefit from using a scanning method whereby its precision and accuracy are known.

CHAPTER 8

FURTHER SLIT LAMP SYSTEM UPGRADES AND RECONSTRUCTION IN A LIFE-SIZED FLUID-FILLED EYE PHANTOM

8.1 Introduction

The work detailed in this chapter built on the work of the previous chapter, which involved a minor change to the slit lamp system and the creation of a new eye phantom, to address some of the issues listed in Table 10. The main aspect of this work was the creation of a life-sized fluid-filled eye phantom, which more closely matches the optical characteristics of the human eye than the phantoms used in Chapters 5, 6 and 7, in order to address issue 11 (eye phantom optics different to human eye). As well as being life-sized and fluid-filled, it contained a cornea and lens. The second core aspect of this work relates to the performance evaluation conducted (issue 14: repeated measures not representative of imaging a human eye). It involved repeating imaging of targets held in the same position, and also in two different positions, to enable for a better sense of the precision compared to in the work of previous chapters, and also get a better idea on the error caused separately by stereo matching and calibration. Also related to the performance trial, ground truth data acquisition was carried out using a highly accurate and precise optical measurement system, to address issue 15 (unclear ground truth accuracy and precision). Regarding upgrades to the slit lamp system, the cameras were switched with ones which were easier to work with.

8.2 Eye Phantom Development

8.2.1 Brief Overall Description

A basic sectional 3D CAD rendered view and photograph of the eye phantom created is shown in Figure 91. The phantom featured a cornea (shown in green), lens (shown in blue), fluid-filled interior, ONH target (shown in orange), and 3D printed main body (shown in red); and its optical design was based on the gold-standard reference eye model commonly known as Navarro's eye model (Escudero-Sanz & Navarro, 1999). The phantom can also be used for calibration, as is detailed in subsubsection 8.2.3.4.

8.2.2 Optical Design

8.2.2.1 Overall Design Considerations

The first step in designing the optical configuration of the eye phantom was to select an appropriate optical model (or schematic eye model) on which to base the design. A number of schematic eye models with varying degrees of complexity exist. They range from very

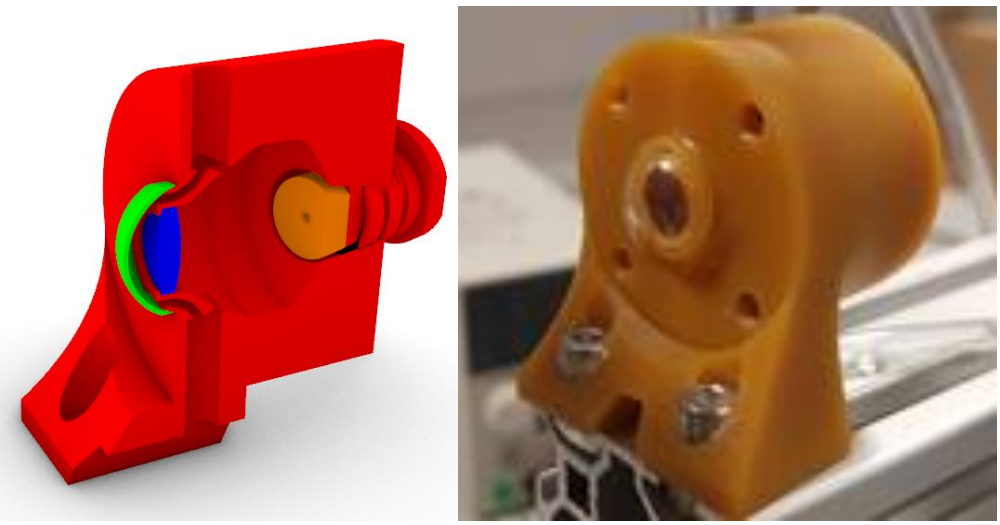


*Figure 91: 3D CAD rendered Sectional 3D model (**LEFT**) and photograph (**RIGHT**) of the life-sized fluid filled eye phantom. **Green**: Cornea; **Blue**: Stock Bi-Convex Lens; **Orange**: ONH target; **Red**: 3D printed parts and sealing rings.*

simple models with only a single refracting surface to those which have two refracting surfaces to model the cornea, and a gradient-index lens to model the crystalline lens (Atchison & Thibos, 2016). As was done by some researchers in designing their eye phantoms (Corcoran et al., 2015; Xie et al., 2014), it was decided to base the optical design of this phantom on Navarro's wide-angle schematic eye model (Escudero-Sanz & Navarro, 1999). This schematic eye model strikes a good balance between optical performance and complexity. Importantly, it accurately accounts for the optical performance of the average adult human eye off-axis, which is appropriate for ONH related work as the ONH is located at around 10^{o} off the axis of the lens (Atchison & Smith, 2000). The optical data for Navarro's eye model are shown in the first four columns of Table 11, while a diagram depicting its design is shown at the top of Figure 92.

Table 11: Optical Data for Navarro's Wide-Angle Schematic Eye Model and the Proposed Eye Phantom.

Surface	Navarro's Wide-Angle Schematic Eye Model				Eye Phantom			
	Radius (mm)	*Thickness (mm)*	*Q*	*Refractive Index (@ 589.3 nm)*	*Radius (mm)*	*Thickness (mm)*	*Q*	*Refractive Index (@ 589.3 nm)*
1	7.72	0.55	-0.26	1.376	7.5	0.5	0	1.5915
2	6.5	3.05	0	1.3374	7	3.05	0	1.3324
Iris	∞	0	-	1.3374	∞	0	-	1.3324
3	10.2	4	-3.1316	1.42	18.15	2.6	0	1.5167
4	-6	16.3203	-1	1.336	-18.15	17.465	0	1.3324
5	-12	-	0	-	-12	-	0	-

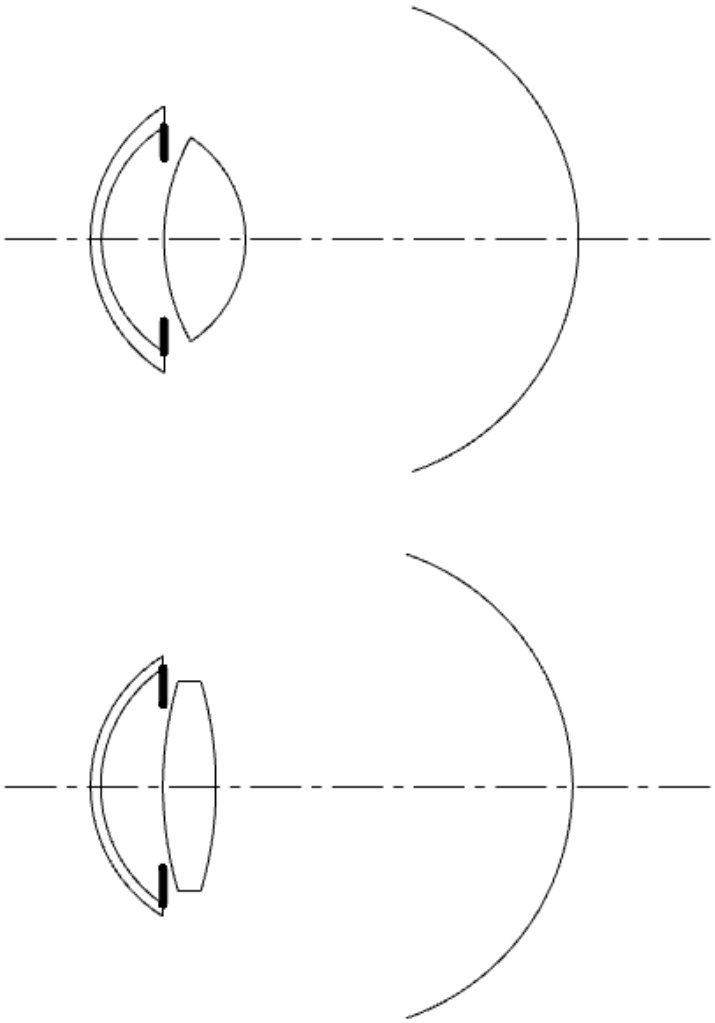


*Figure 92: Diagrams depicting the optical design of Navarro's eye model (**TOP**) and the life-size fluid-filled eye phantom (**BOTTOM**).*

Approximating Navarro's eye model physically is challenging and requires custom optics to be fabricated. A number of researchers have created eye phantoms, containing custom optics, that closely matched the optical design of Navarro's eye model, such as Corcoran and colleagues (Corcoran et al., 2015). In the current work however, custom optics were not used, in order to contain costs. It was decided to fabricate the cornea by vacuum forming a clear plastic sheet over a ball bearing as this could be done at low cost and yield a cornea relatively similar to the human cornea. In terms of the lens, it was decided to use a low-cost stock bi-convex lens to imitate the crystalline lens. The use of an IOL, as was the case for Xie and colleagues (Xie et al., 2014) with their phantom, was considered but not used as they are typically only around 6 mm in diameter which limits the pupil size by too much – it would have needed to be less than 6 mm to avoid lens edge effects. While the stock bi-convex lens

differed significantly in terms of dimensions and refractive index from the lens of Navarro's eye model, one that matches closely, in terms of optical characteristics, was selected.

8.2.2.2 Cornea Design

The cornea of the eye phantom was designed such that its refractive power matched that of the cornea in Navarro's eye model as closely as possible. The vacuum forming technique, as mentioned above, constrained its design. First, the thickness of the cornea could only be uniform, however, as it could be argued that the cornea of Navarro's eye model does not deviate significantly from uniformity, this constraint was not considered to constitute a significant limitation. Secondly, the radius of the posterior surface of the cornea would need to match that of an available forming mould which in this case is effectively a ball bearing. The radius of the anterior surface would be set based on these two constraints just mentioned – it is a function of the thickness, and radius of the posterior surface. It is also important to note at this stage that, as the ball bearings are perfectly spherical, the two surfaces could only be spherical. The last important constraint to mention relates to the refractive index. This was constrained to that of an appropriate vacuum forming material. Despite its high index of refraction, it was decided to use High Impact Polystyrene (HIPS; n = 1.5915 at 589.3 nm [Sultanova et al., 2009]) as the cornea needed to operate while immersed in a liquid. HIPS is non-hygroscopic, and the alteration of optical and mechanical properties by the fluid interior of the eye phantom is low compared to most transparent plastics. HIPS also has excellent optical quality and is readily available in a thickness of 0.5 mm, close to the 0.55 mm central thickness of the cornea in Navarro's eye model. Using the equations described by Xie and colleagues (Xie et al., 2014) for determining corneal power, it was calculated that the cornea in Navarro's eye model has a power of +42.88 dioptres. Using the same equations, it was determined that a HIPS cornea of uniform thickness with anterior and posterior surfaces radii of 7.5 mm and 7 mm, respectively, achievable by vacuum forming a 0.5 mm thick HIPS sheet over a 14 mm diameter ball bearing, would have a power of +42.77 dioptres, hence this was chosen for the design.

8.2.2.3 Lens Selection

The bi-convex lens was selected based on comparing the optical simulation results of models of eye phantom designs that contained various commercially available stock bi-convex lenses with those obtained from a model of Navarro's eye model to find which resulted in the closest match in terms of resolution, axial length and distortion. The models were created

with the optical design software Zemax OpticStudio 14.2 (Kirkland, Washington, US). The models of the eye phantoms were created in the following way:

- The HIPS cornea was implemented.
- The anterior chamber depth was set to be the same as that in Navarro's eye model (3.05 mm).
- The pupil size of the iris, placed in the same position as in Navarro's eye model, was set to 8 mm.
- A bi-convex lens was implemented.
- As water would be used to fill the interior of the eye phantom, the refractive index of the aqueous and vitreous were set to be that of water (n = 1.3324) at 589.3 nm and 25^{o}C (Hale & Querry, 1973).
- The radius of the retina was set to be the same as that in Navarro's eye model (12 mm).
- Fields were set up at angles between 0^{o} and 25^{o}, in increments of 5^{o}.
- The distance between the posterior surface of the bi-convex lens and the retina was adjusted automatically to minimise the root-mean-square (RMS) spot size.

This process was carried out for a number of candidate bi-convex lenses. The different lenses evaluated are tabulated in Table 12. A model of Navarro's eye model was also implemented – the pupil size of which was set to be the same as in the models of the eye phantom (8 mm). Simulations were conducted at a single 589.3 nm wavelength. In each case, the axial length was determined, and the RMS spot radius and distortion were recorded between field angles of 0^{o} and 25^{o}. The results obtained through the various simulations are depicted in Figure 93. In terms of distortion, the use of many of the lenses resulted in very similar percentages of distortion at all field angles tested, apart from with the 20DQ10 lens. This trend can also be seen when looking at the axial lengths – the use of most lenses resulted in a length that does

Table 12: Lenses Evaluated for Use in the Eye Phantom

Lens	Manufacturer
16VQ10	Comar Optics Ltd (Linton, UK)
20VQ10	Comar Optics Ltd (Linton, UK)
16DQ10	Comar Optics Ltd (Linton, UK)
20DQ10	Comar Optics Ltd (Linton, UK)
32-968	Edmund Optics Ltd (York, UK)
LB1212	Thorlabs (Newton, New Jersey, US)

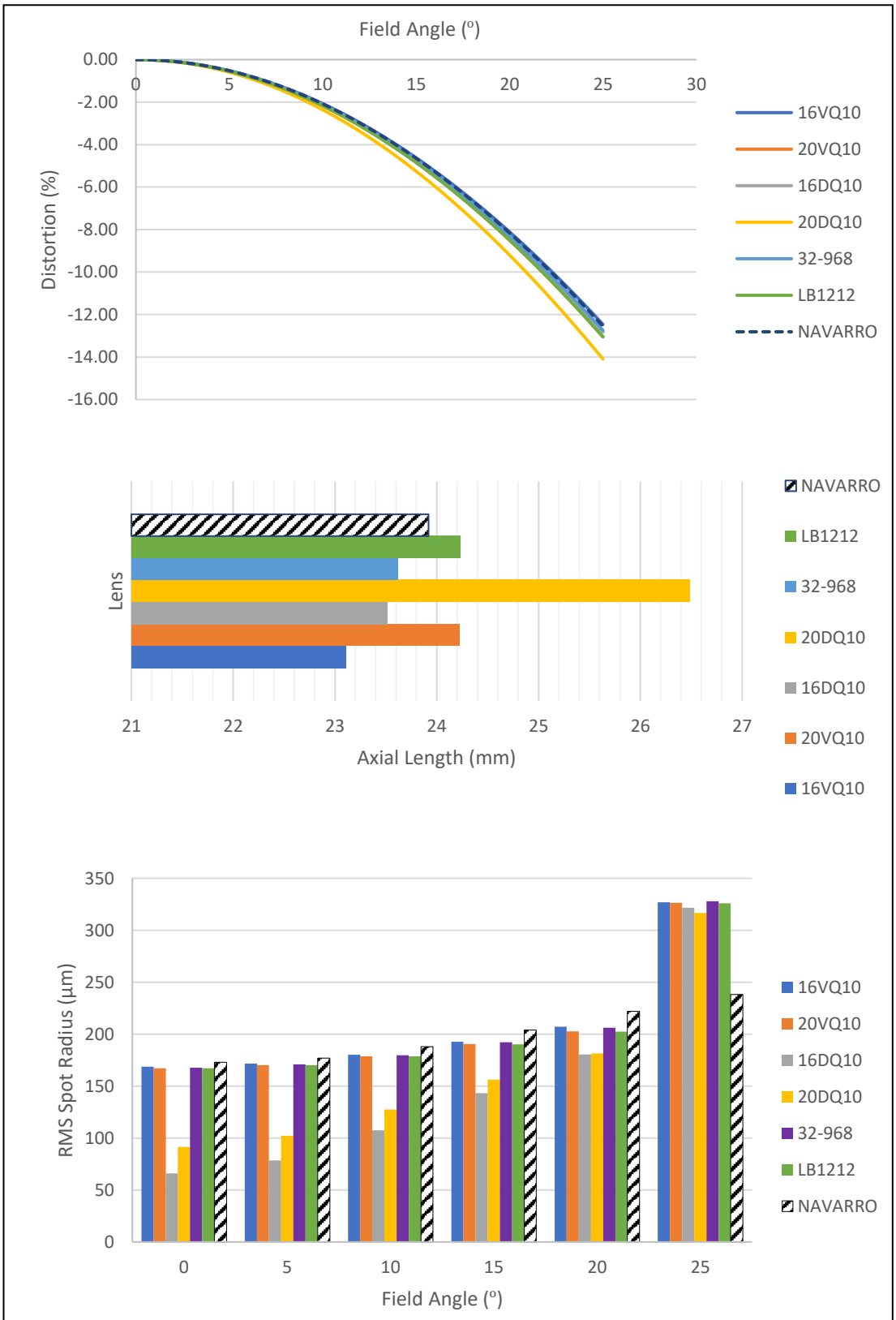


Figure 93: Optical simulation results of the Navarro eye model and the eye phantom designs with the various commercially available stock lenses used as the lens. ***TOP****: Distortion vs field angle.* ***MIDDLE****: Axial Length.* ***BOTTOM****: RMS spot radius vs field angle.*

not deviate much from that of Navarro's eye model. In terms of RMS spot radius, the value obtained with all of the lenses is lower than in Navarro's eye model for all of the field angles tested up to and including 20°. There is a spike in RMS spot radius for all of the lenses at 25°, where the values obtained are much greater than for Navarro's eye model. It is likely that this could have been corrected for by optimising the curvature of the retina, potentially making it aspherical, as perhaps reasonable focus on the retina is not maintained after somewhere between 20° and 25°. It was decided that the eye phantom best matching Navarro's eye model over the three characteristics analysed contained the 32-968 lens (Edmund Optics Ltd, York, UK). Hence, it was selected for use in the eye phantom to be fabricated. The final optical design of the complete eye phantom is described by the optical data provided in the rightmost four columns of Table 11 and depicted in Figure 92 above. It should be noted that the surfaces in the eye phantom are spherical only, which is why a value of zero has been set for the Q value of each of the surfaces.

8.2.3 Mechanical Design and Fabrication

8.2.3.1 Introduction

3D CAD rendered views and photographs of the eye phantom are provided in Figure 94 to complement the description that follows. In this figure, the eye phantom is shown in its imaging (left top and middle images) and calibration (right top and middle images) configurations. These configurations share the same anterior section (containing the cornea, lens and iris) but have different posterior sections which can be interchanged to switch between the configurations. The ONH models (bottom left) and calibration target (bottom right) used are also shown in the figure.

8.2.3.2 3D Printing

All of the 3D printed components of the phantom were printed in EnvisionTEC ABS Tough (EnvisionTEC GmbH, Gladbeck, Germany) on the Perfactory Aureus 3D printer (EnvisionTEC GmbH, Gladbeck, Germany), apart from the mount of calibration target, which was printed in black PLA+ filament on the Wanhao Duplicator i3 3D printer.

8.2.3.3 Imaging Configuration

The eye phantom in its imaging configuration features an anterior section and a posterior section, which is used to hold ONH targets in place. The anterior section holds the cornea and lens in place and features a fixed pupil size, 8 mm in diameter and 0.35 mm thick,

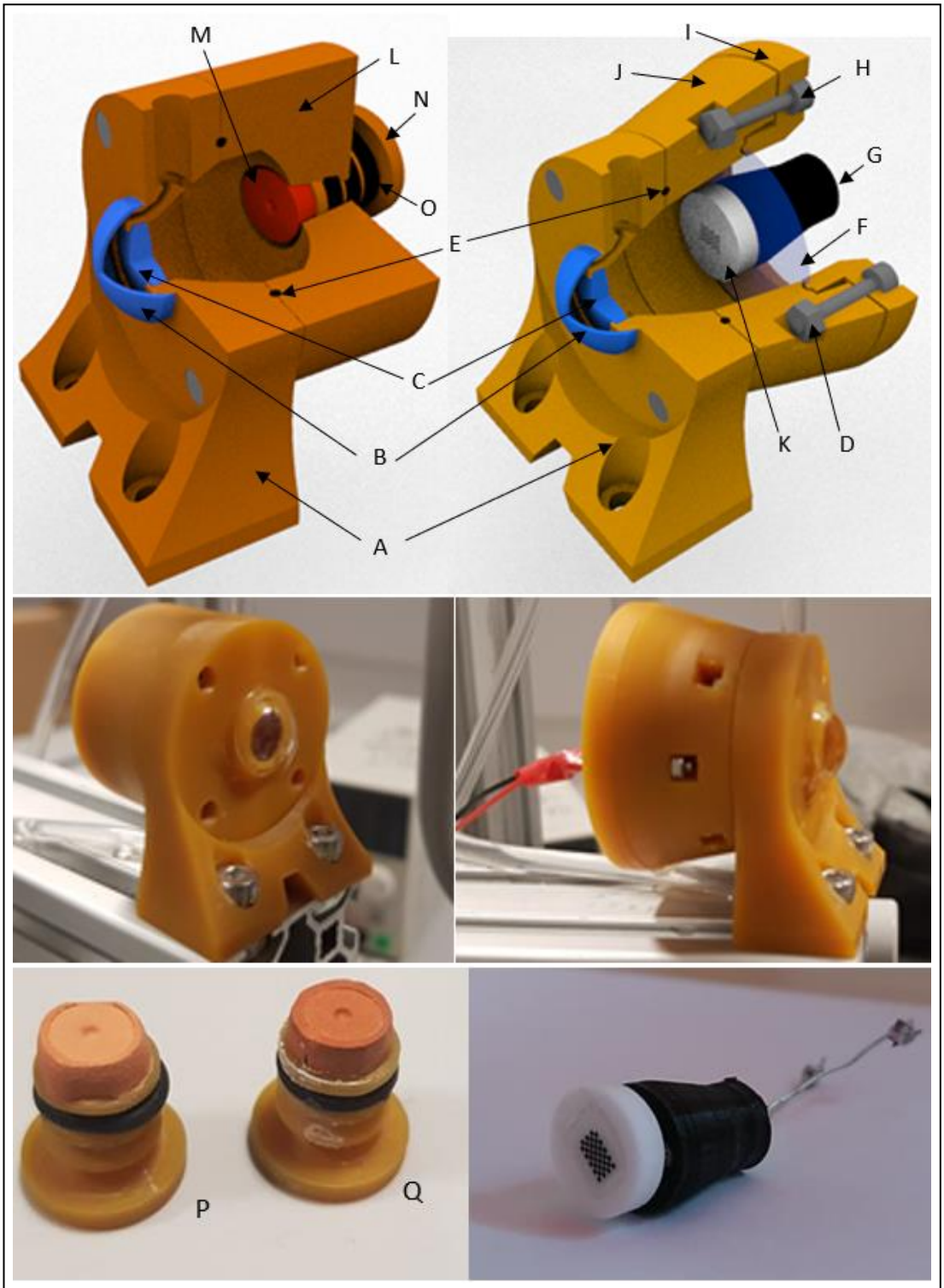


*Figure 94: Sectional 3D model (**TOP LEFT**) and photograph (**MIDDLE LEFT**) of the life-sized fluid filled eye phantom. Sectional model (**TOP RIGHT**) and photograph (**MIDDLE RIGHT**) of the phantom when being used for imaging system calibration. Photograph of optic nerve head targets (**BOTTOM LEFT**) and calibration pattern component (**BOTTOM RIGHT**). **LABELS**: **A** – Anterior Section, **B** – HIPS Cornea, **C** – Bi-Convex Lens, **D** – M2 Nut, **E** – Nitrile Rubber O-Ring, **F** – Nitrile Rubber Sheet, **G** – LED Mount, **H** – M2 Bolt, **I** – Posterior Section for Calibration's Main Body, **J** – Posterior Section for Calibration's Retaining Ring, **K** – Calibration Pattern, **L** – Posterior Section for Imaging, **M** – Terracotta ONH Model, **N** – ONH Model Mount, **O** – Nitrile Rubber O-Ring, **P** – Normal ONH target, **Q** – Glaucomatous ONH target.*

incorporated into its design. The main body of this section was 3D printed. As explained in the optical design section, the cornea was fabricated from a 0.5 mm thick HIPS sheet (4D Modelshop Ltd, London, UK). Fabrication of this involved vacuum forming over a 14 mm diameter grade 100 hardened 52100 chrome steel ball bearing (Simply Bearings Ltd, Leigh, Greater Manchester, UK) using a dental vacuum forming machine (Jintai, Yuyao, Zhejiang, China). The formed cornea was trimmed down to a diameter of approximately 13.5 mm, allowing for it to be fitted onto the anterior section. It was secured to it using Loctite 4305 UV Curable Adhesive. The bi-convex lens was pressed into position against the back of the iris and secured with a very small amount of the same adhesive. In order to allow filling of the eye phantom with water, the anterior section contained holes on its top and bottom that lead to channels which allowed water to pass into the space between the cornea and lens (anterior chamber) and behind the lens (vitreous chamber). A 50 ml disposable syringe (Dieckhoff & Ratschow Praxisdienst GmbH & Co.KG, Longuich, Germany) was connected to the hole at the bottom of the phantom using an IV extension tube (Dieckhoff & Ratschow Praxisdienst GmbH & Co.KG, Longuich, Germany) to enable injection of water into the phantom. A second IV extension tube was connected to the hole at the top of the phantom and secured at a height at its other end, with another syringe connected, allowing the displaced air to escape.

The posterior section was bolted, using M2 nuts and bolts, to the anterior section. A nitrile rubber O-ring (SourcingMap, Hong Kong, China) was placed between these two sections to create the seal. The posterior section allowed for an ONH target to be held in the appropriate position at the back of the eye phantom. The desired location for the ONH targets was 10^{o} left off-axis horizontally and 1.5^{o} upwards to match ONH position in the human left eye (Atchison & Smith, 2000). The main body of this section was 3D printed. In fact, two main bodies were fabricated: one where the ONH target is in the correct position according to the human left eye and the other where it is in the correct position according to the human right eye. The design of the second was simply created by horizontally mirroring the design of the first. The ONH targets were attached to 3D printed mounts, using Loctite 4305 UV Curable Adhesive, that were inserted into position in the main body, and sealed using nitrile rubber O-rings. The mounts position could be adjusted, allowing modification of the lens to ONH target distance, and ultimately the focus. Two ONH targets were created for this eye phantom: one resembling an ONH without pathology and the other resembling an ONH with significant glaucomatous damage. These had the same dimple morphologies as those used

in Chapters 6 and 7, whose design is detailed in subsubsection 6.2.2.1, but were of the correct scale for this eye phantom (4.3 times smaller than in the previous eye phantoms). It was decided to manufacture the targets from terracotta using a computer numerical control (CNC) engraving machine (Shenzhen Ni Xing Ni Shang Electronic Commerce Co Ltd, Shenzhen, China). One of the reasons for this was that 3D printing using the resin that was available would have created targets with a relatively reflective surface and applying paint to overcome this would have been difficult due to its size. It was decided to use terracotta as it is an easily obtainable type of ceramic that already has a colour not too dissimilar to the human retina while also exhibiting virtually no specular reflection.

8.2.3.4 Calibration Configuration

In order to switch the eye phantom from its imaging to calibration configuration, the posterior section which holds the ONH targets needed to be removed and the posterior section for calibration bolted, using M2 nuts and bolts, onto the anterior section. The seal between these two parts was again made using a nitrile rubber O-ring. This configuration allowed for a flat checkerboard pattern to be held in many different positions, with different tilts, inside the eye phantom at the position of the retina, while also still being sealed so that it could be filled with water. To enable this, the phantom was capped at the back with a thin nitrile rubber sheet, cut from nitrile gloves (Supermax Healthcare Ltd, Peterborough, Cambridgeshire, UK), clamped between the main body of the calibration posterior section and its retaining ring (both 3D printed). The retaining ring was tightened to the main body using M2 nuts and bolts. The calibration pattern was located on one side of the rubber sheet, facing the inside of the phantom. It was magnetically connected, through the thin sheet, to a mount that could be held and used to manipulate the patterns position and tilt. The calibration pattern was created by gluing, using Loctite 4305 UV Curable Adhesive, a film containing a checkerboard pattern, created photographically using the same film and technique as was detailed in subsection 6.2.3, to a cast white acrylic disc. The disc was 12 mm in diameter and 3 mm thick, and laser cut from a 3 mm thick cast white acrylic sheet (Simply Plastics Ltd, Essex, UK). As was the case before, the film was transparent with only the black squares of the pattern, the white squares and background were provided by the white backing (i.e., the acrylic disc). In the same manner as was detailed in subsubsection 6.4.3.2, the square size of the pattern was determined to be 0.3967 mm (SD = 0.0022). On the other face of the acrylic disc, four neodymium magnets (Magnet Expert Ltd, Tuxford, UK), 2 mm in diameter and 1 mm thick, were glued on using Loctite 4305 UV Curable Adhesive.

These magnets were magnetically connected, through the rubber sheet sealing the phantom, to four of the same type of magnet glued onto, using the same glue, a 3D printed mount. The mount held, at its other end, a 5 mm through hole white LED (RS Components Ltd, Corby, Northamptonshire, UK), allowing for retroillumination of the checkerboard pattern to avoid difficulties with specular reflections from the shiny surface of the film.

8.3 Slit Lamp System Changes

The slit lamp add-on components used in the work of this chapter remained mostly unchanged from that used in the previous chapter. In this work however, the camera module was swapped for a different camera. Such changes are reflected in Figure 95. The new camera was more straightforward to work with since it could be connected to a PC via USB and operated like a webcam. The new camera made stereo image capture much faster, which was particularly advantageous for calibration where many images needed to be taken. The camera used was a synchronised stereo camera module (Kayeton Technology Co, Ltd, Shenzhen, China) which featured two 2-megapixel cameras. It must be noted that while the resolution is much less than that of the raspberry pi cameras, they do have a larger sensor size (5.44 x 3.072 mm vs 3.68 x 2.76 mm), and pixel size (2 µm vs 1.12 µm), which may mean that they perform better in low light tasks, such as this work. The same M12 lenses with a focal length of 8 mm were used with this module. The module was mounted to the slit lamp using a 3D printed mount, printed in ecoPLA on the Ultimaker 2+, that clamped onto its eye pieces. Unlike the previous camera module where the cameras were placed on mounts that allowed for the distance to the eye pieces, and their tilt, to be adjusted, the positions of the cameras were fixed in this new module.

8.4 Performance Evaluation

8.4.1 Brief Overall Description

The slit lamp with fitted add-on components was used to capture the required images of the ONH targets while different speckle patterns were projected onto their surfaces, and the calibration target, in the eye phantom. The images were processed using the same stereo vision pipeline, featuring a coarse-to-fine stereo matching algorithm and smoothing splines-based interpolation, as was detailed subsection 6.4.3. The pipeline used in the previous

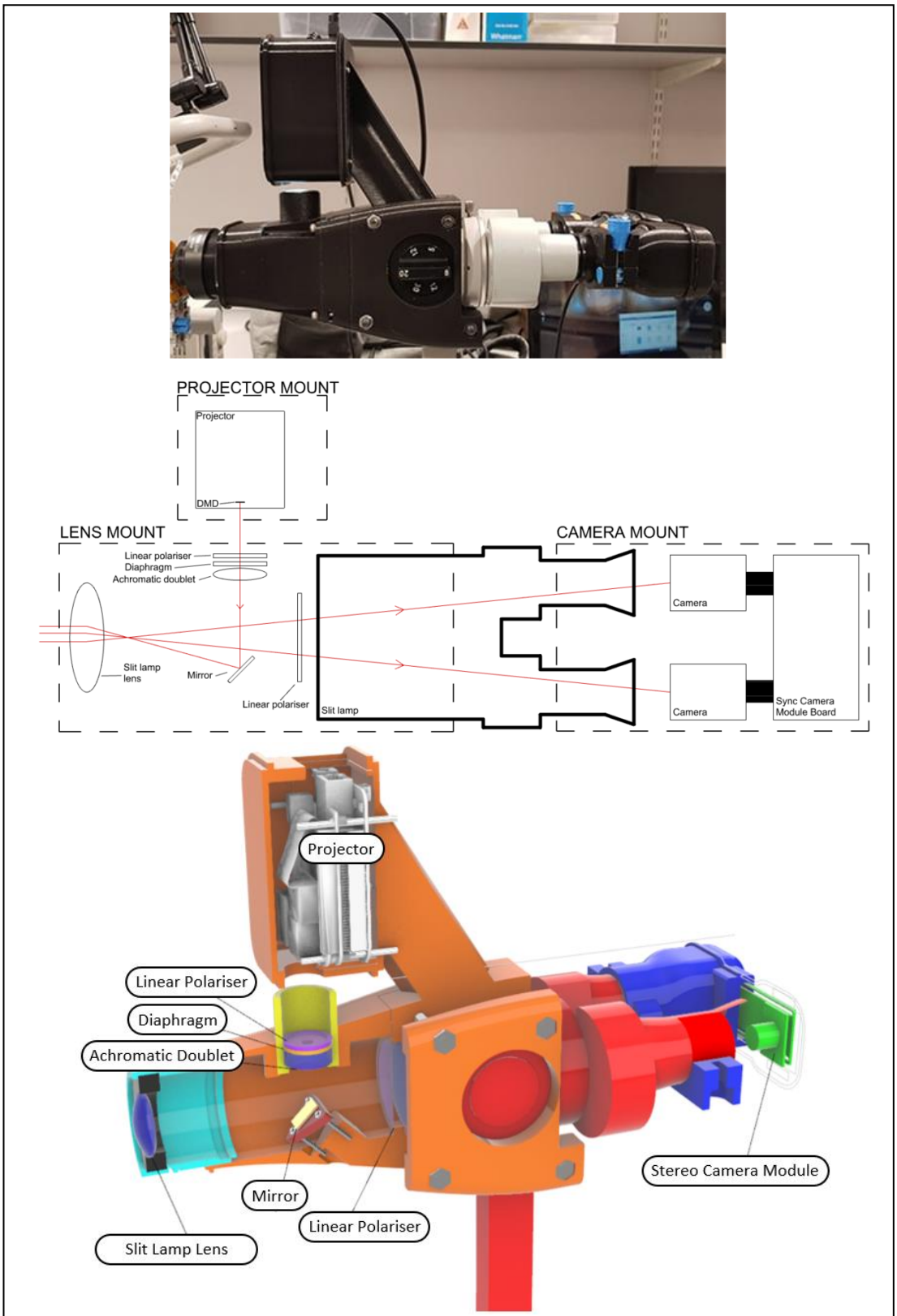


*Figure 95: Photograph (**TOP**) with corresponding schematic (**MIDDLE**) and colour-coded and labelled CAD rendered 3D model (**BOTTOM**), with some parts cut to reveal internal components, of the slit lamp fitted with add-on components. **Blue** – 3D printed camera mount; **Red** – Zeiss 30 SL-M slit lamp; **Orange** – Slit lamp lens and projection components mount; **Cyan** – Lens tubes; **Black** – Slit lamp lens mount.*

chapter was not used since some of the speckle patterns projected in this work were not compatible with it – they were not all dot patterns, where the dots could be located and used as locations to match. Following reconstruction, the reconstructions obtained were quantitatively compared against ground truth data.

8.4.2 Trial Setup and Image Capture

8.4.2.1 Setup

The setup of the experiment is shown in Figure 96. The setup is not detailed in full since it was set up in the same way as in the previous chapter, with the only differences being that in this case the eye phantom and cameras were different. It must be noted that the patterns projected were also different.

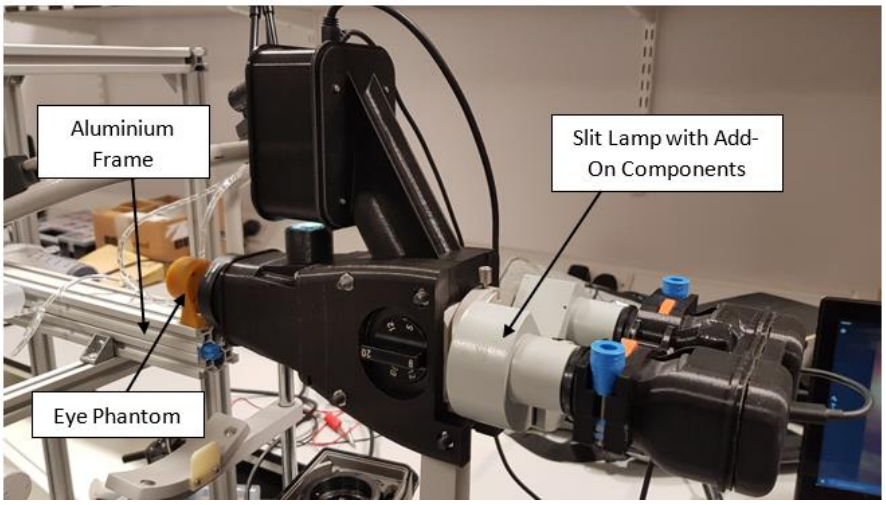


Figure 96: Experimental setup.

8.4.2.2 Image Capture of the ONH Targets

Stereo images were acquired of the targets, with different patterns projected onto their surfaces each time, according to the following:

a) 4 stereo images of the glaucomatous target in one position.
b) 2 stereo images of the glaucomatous target in a position shifted approximately 2 mm downwards, and 2 mm to the right, of the position in a.
c) 3 stereo images of the normal target in one position.

8.4.2.3 Image Capture of the Calibration Target

The movements required in moving the calibration target to different positions and tilts were very subtle and could not be well achieved using the flexible helping-hand used previously,

and as such had to be held by hand. In total, 66 images were acquired of the calibration target. The reason for fewer images to be taken in this case was due to the significantly longer time required to take each image in this case, due to the difficulty of handling.

8.4.3 Stereo Vision Pipeline

The only aspect of the pipeline which varied from that detailed in subsection 6.4.3 was the calibration step. Calibration was performed on 66 images to begin with but then those images where one or more data points corresponded to a reprojection error of greater than 2.5 pixels in any direction, were removed. 40 images remained after this. Calibration was then performed with the remaining images. Coefficients/parameters and associated uncertainties were recorded for 4 to 40 images, in increments of 4. The final calibration values used in this work were those that were estimated using the remaining 40 images.

8.4.4 Quantitative Validation

8.4.4.1 Ground Truth Data Acquisition

Ground truths of each of the targets were obtained using the Alicona Infinite Focus IFM G4 (Alicona Imaging GmbH, Graz, Austria), a commercial optical 3D measurement system which, at the setting used, offers a vertical resolution of 410 nm – more than sufficient for this task.

8.4.4.2 Quantitative Comparison

Given that the targets did not have any anchor points on them in this case, no measurements could reliably be taken from the reconstructions and compared against ground truth data. As such, only cloud-to-cloud comparisons, using the CloudCompare software, were made. Cloud-to-cloud comparison was carried out between:

a. The 4 reconstructions of the glaucomatous target in one position (*to estimate the precision).*
b. The 2 reconstructions of the glaucomatous target in a position shifted approximately 2 mm downwards, and 2 mm to the right, of the position in a (*to estimate the precision*).
c. The 3 reconstructions of the normal target in one position (*to estimate the precision*).
d. One reconstruction each from (a) and (b) (*to estimate the precision when the target is imaged in two different locations*).
e. The reconstructions in (a) against the ground truth data (*to estimate the accuracy*).
f. The reconstructions in (c) against the ground truth data (*to estimate the accuracy*).

In a and b, the reconstructions did not need to be aligned since they were already, due to being in the same position during capture. In c there appeared to be a slight shift in position, potentially due to a small movement of the setup between captures, and thus they needed to be aligned. Alignment was also required in d, e and f. In this case, alignment was carried out first by rough manual alignment, then using the fine registration tool of CloudCompare. It must be noted that only rigid motion movements were used for alignment. In each case, the clouds were cropped to be the same size.

8.5 Results and Discussion

8.5.1 Calibration Performance

8.5.1.1 Reprojection Error

Reprojection error plots from the initial calibration with all 66 images, and the calibration using 40 images after bad images were removed, are shown in Figure 97. It can be seen that the outlier points are reduced but not by a significant amount. Such reduction is reflected in the overall mean reprojection error values obtained. A value of 1.08 pixels (3SD = 2.04) was obtained in the calibration using all 66 images, while a value of 0.89 pixels (3SD = 1.52) was obtained with the images remaining after removal. A lower reprojection error threshold could have led to less erroneous points, and a lower mean reprojection error, but may have resulted in few images remaining for calibration. The mean reprojection error seen here is only 0.25 pixels larger than that seen in the work of the previous chapter. However, given

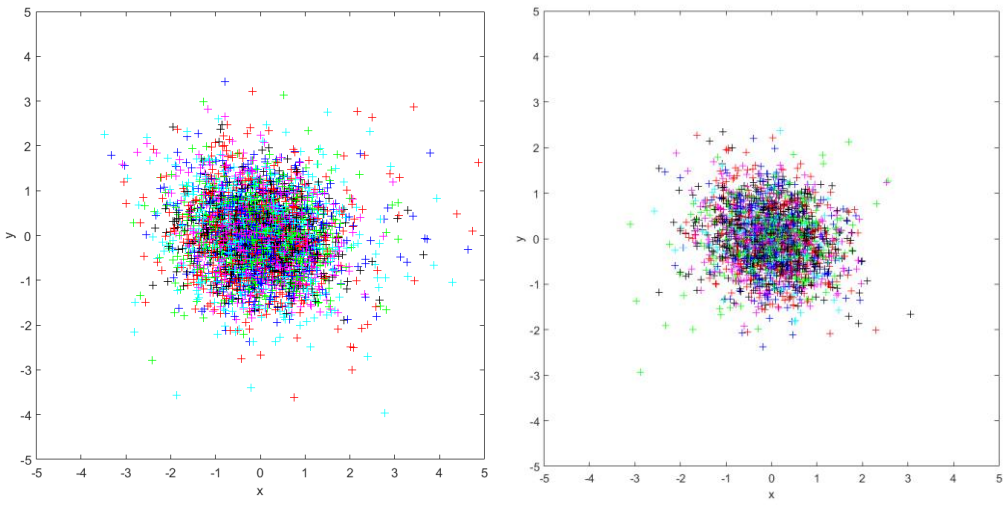


*Figure 97: Reprojection error plots from the calibration before (**LEFT**) and after (**RIGHT**) removing bad images.*

that the pixel size of the camera used in the previous work was smaller, the errors seen here are in fact much larger. The error from the work in the previous chapter, expressed in µm on the sensor, was 0.72 µm (0.64 pixels), whereas in this work the same is 1.78 µm (0.89 pixels).

8.5.1.2 Estimated Coefficients and Parameters

The graphs obtained, showing each of the calibration values and associated uncertainties (3 times the SD) obtained using an incremental number of calibration images, are shown in Appendix 8A. The calibration estimation failed to converge with some specific numbers of calibration images and is the reason for some missing data points in the graphs. It must be noted that graphs of the values obtained for the principal points at various increments were not included since the calibration algorithm did not recompute these in the stereo calibration step. This is due to the toolbox disabling recomputation of principal points as they were determined to be outside of the image in the single camera calibration step.

The graphs indicate that the parameters mostly stabilise by 40 images, indicating that more calibration images of the same quality would not have led to very different results. As was the case in the works of Chapters 6 and 7, the distortion maps, shown in Figure 98 and Figure 99 for the left and right cameras respectively, look to be abnormal compared to that which would be expected from a typical camera.

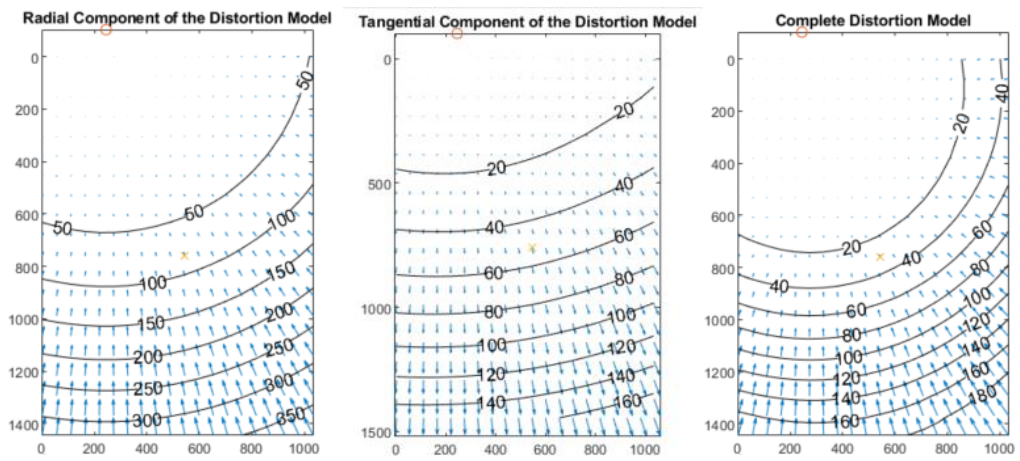


Figure 98: Distortion maps to visualise the distortion parameters estimated for the left camera.

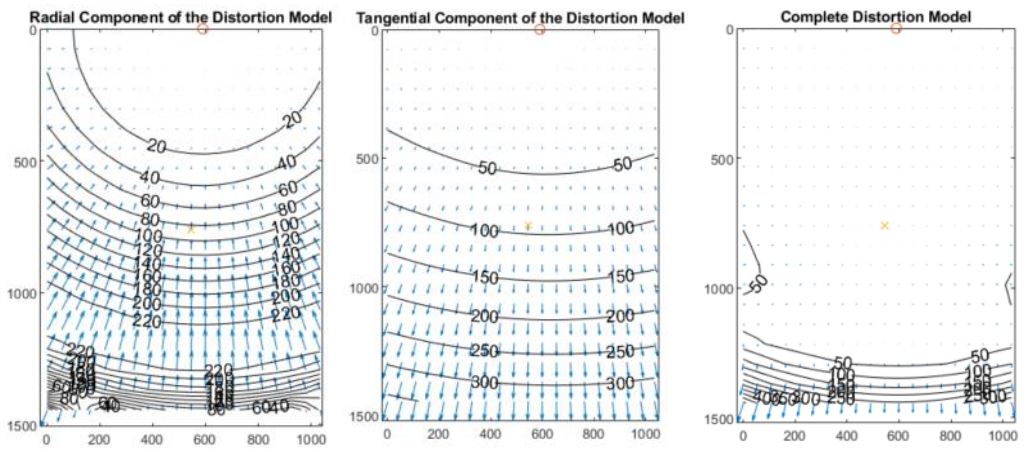


Figure 99: Distortion maps to visualise the distortion parameters estimated for the right camera.

Table 13: Calibration coefficients and parameters estimated for the system. SD: Standard Deviation.

	Coefficient/Parameter	Value	Uncertainty (3SD)	% Uncertainty
Left	f_x (pixels)	1776	57	3.21
	f_y (pixels)	1752	55	3.14
	C_x (pixels)	244	N/A	N/A
	C_y (pixels)	-104	N/A	N/A
	k_1	-0.326	0.042	-12.88
	k_2	-0.0433	0.113	-260.97
	k_3	0.154	0.124	80.52
	P_1	0.0366	0.0065	17.76
	P_2	0.00388	0.00205	52.84
Right	f_x (pixels)	1815	58	3.20
	f_y (pixels)	1741	55	3.16
	C_x (pixels)	589	N/A	N/A
	C_y (pixels)	-1	N/A	N/A
	k_1	-0.511	0.07	-13.70
	k_2	-0.745	0.33	-44.30
	k_3	2.01	0.72	35.82
	P_1	0.091	0.0129	14.18
	P_2	-0.00444	0.00236	-53.15
	R_1	0.0804	0.006137	7.63
	R_2	-0.0566	0.003501	-6.19
	R_3	-0.00034	9.59E-04	-282.06
	T_x (mm)	-1.85	0.043484	-2.35
	T_y (mm)	-0.254	0.086182	-33.93
	T_z (mm)	-0.263	0.251323	-95.56

The calibration values, and associated uncertainties, determined using 40 calibration images, are shown in Table 13. It can be seen again, as was the case in the calibrations conducted in Chapters 6 and 7, that the distortion coefficients, and x and y translation parameters, have large uncertainties associated with them. In this work, the rotation parameter R_3 also had a high uncertainty tied to it. The focal lengths again were seen to have low uncertainties.

7.5.1.3 Potential Sources of Calibration Error

The potential sources of calibration error in this case are unchanged from those detailed in the previous chapter (subsection 6.5.1): corner localisation errors, inaccurately fabricated

checkerboard, inaccurate checkerboard square size estimation, sub-optimal camera alignment and inadequacy of distortion models.

The fact that the principal points were estimated to be outside of the image sensors further indicates that sub-optimal alignment of the sensors and lenses is likely to be a large cause of error in this work, and likely a large contributor to the abnormal distortion maps obtained. Likely to a lesser extent, the distortion models may not be fully able to model the distortion of this atypical stereo camera system. After this, given that the overall mean reprojection error (and its SD) is high, it is likely that image quality is also a major issue here – affecting corner localisation. Calibration target fabrication error or size measurement are likely not huge contributors of error here, but this needs to be assessed. A further potential source of error relates to the fact that the calibration target was held by hand in this work, and not fixed onto a flexible arm. Given that the synchronicity of the cameras was unknown and that they had rolling shutters, it is possible that small movements of the target during image capture affected the calibration.

8.5.2 Reconstruction Performance

8.5.2.1 Cloud-to-Cloud Comparisons

Figures showing the original stereo images, reconstructions, ground truth reconstructions (where applicable), corresponding depth maps, and a cloud-to-cloud distance error map, are referenced below according to the list of comparisons originally detailed above in subsubsection 8.4.4.2:

a. The 4 reconstructions of the glaucomatous target in one position (position 1) [Figure 100].
b. The 2 reconstructions of the glaucomatous target in a position different from a (position 2) [Figure 101].
c. The 3 reconstructions of the normal target in one position [Figure 102].
d. One reconstruction each from (a) and (b) [Figure 103].
e. The reconstructions in (a) against the ground truth data [Figure 104].
f. The reconstructions in (c) against the ground truth data [Figure 105].

Boxplots of the range values obtained in each of the comparisons made, (a) to (d), with the reconstructions, are shown in Figure 106.

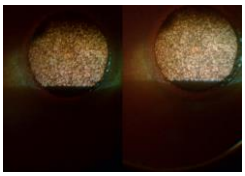
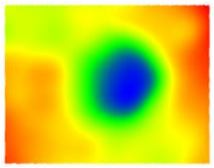
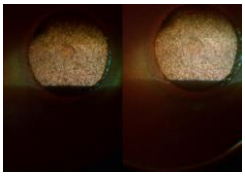
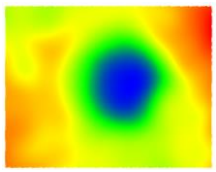
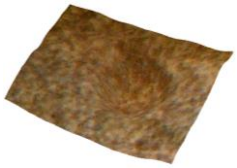
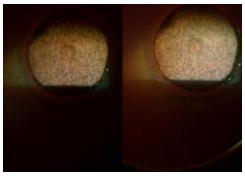
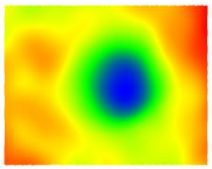
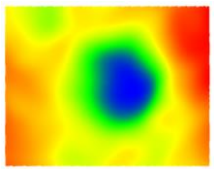
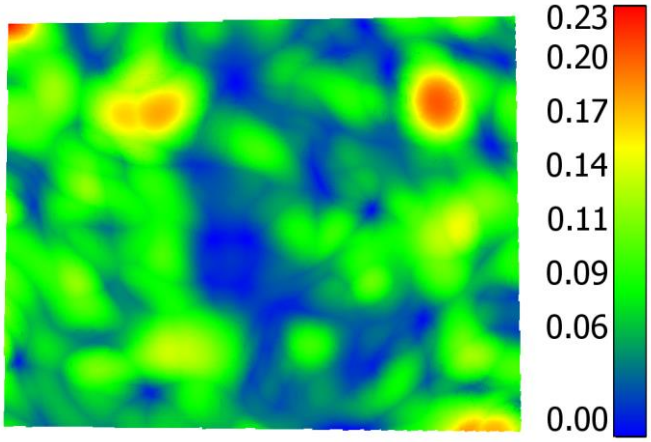
Stereo Image	Depth Map	Reconstruction
		
		
		
		
COMPARISON ***Range*** (scale in mm)		

Figure 100: Cloud-to-cloud comparison of the 4 reconstructions of the glaucomatous target in position one.

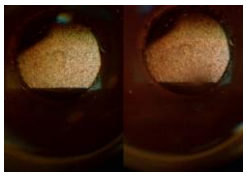
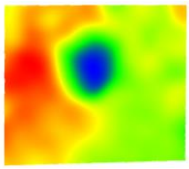
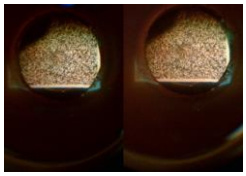
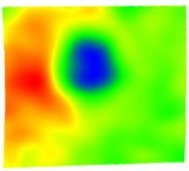
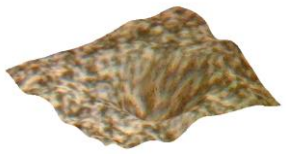
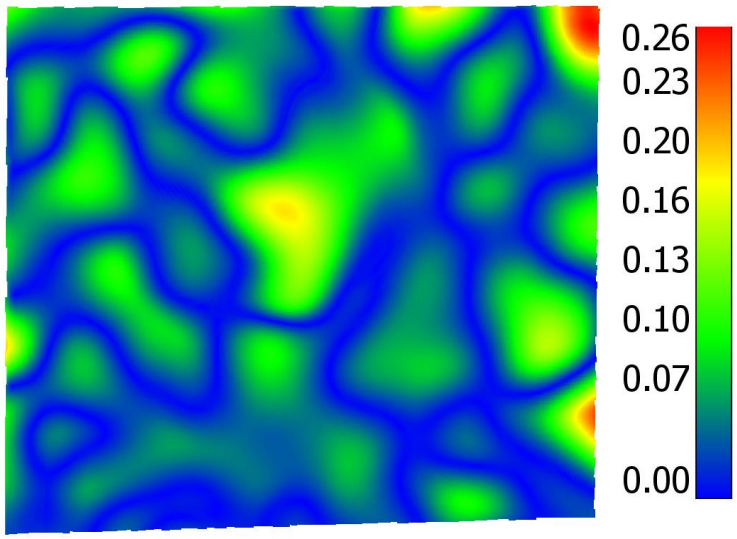
Stereo Image	Depth Map	Reconstruction
		
		
COMPARISON ***Range*** (scale in mm)		

Figure 101: Cloud-to-cloud comparison of the 2 reconstructions of the glaucomatous target in position 2.

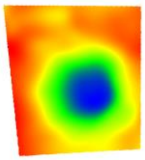
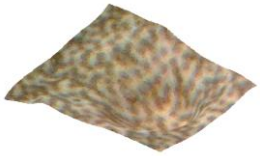
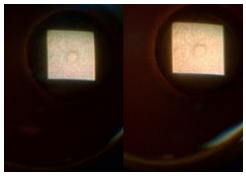
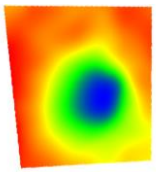
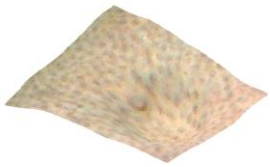
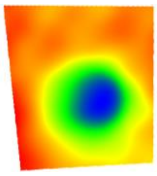
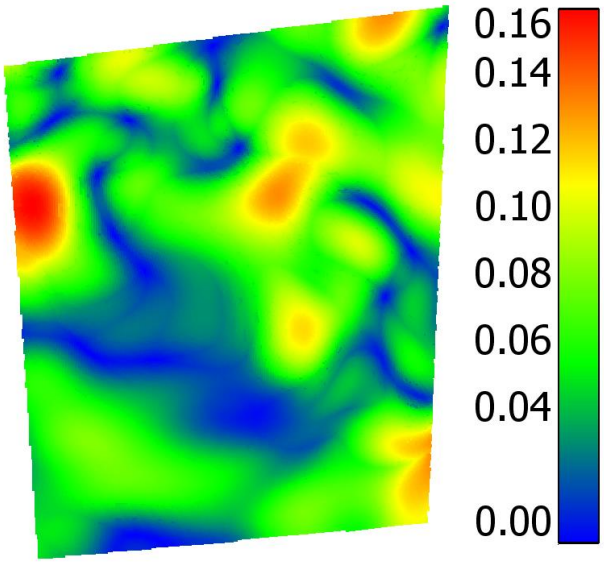
Stereo Image	Depth Map	Reconstruction
		
		
		
COMPARISON ***Range*** (scale in mm)		

Figure 102: Cloud-to-cloud comparison of the 3 reconstructions of the normal target.

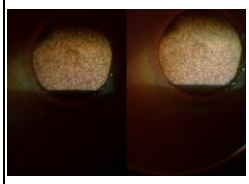
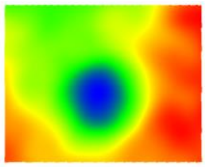
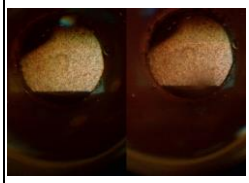
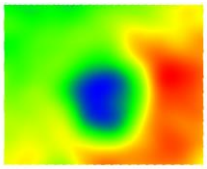
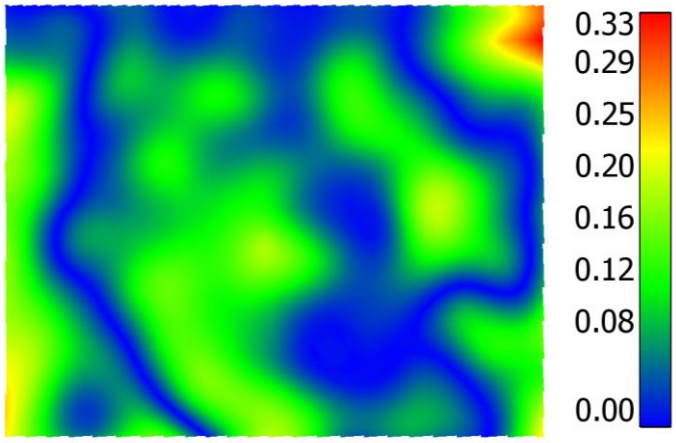
Stereo Image	Depth Map	Reconstruction
		
		
COMPARISON ***Range*** (scale in mm)		

Figure 103: Cloud-to-cloud comparison of two reconstructions of the glaucomatous target held in two different positions.

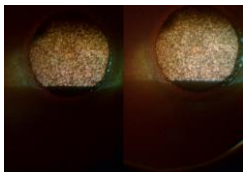
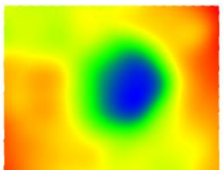
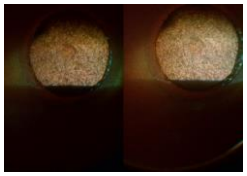
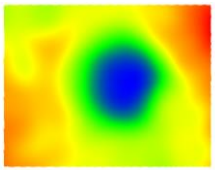
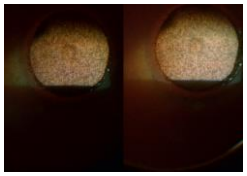
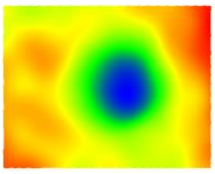
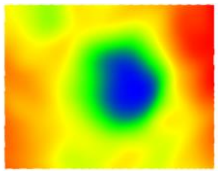
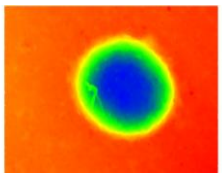
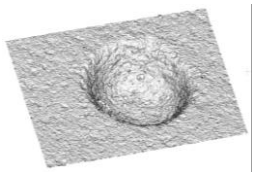
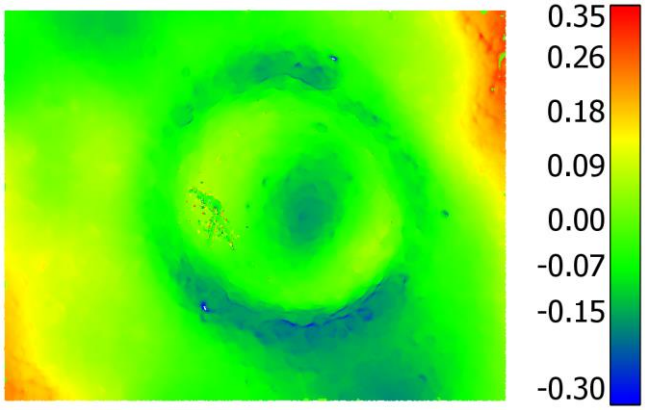
Stereo Image	Depth Map	Reconstruction
		
		
		
		
Ground Truth		
COMPARISON ***Mean Error*** (scale in mm)		

Figure 104: Reconstructions obtained of the normal ONH target, alongside the ground truth data.

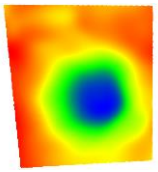
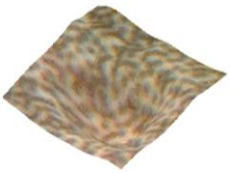
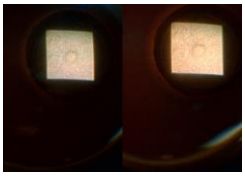
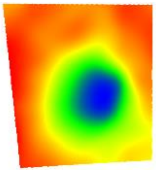
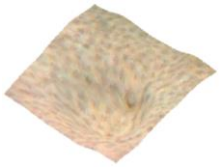
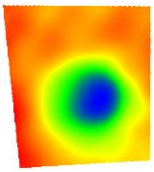
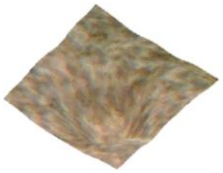
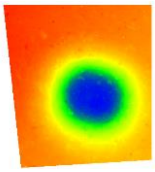
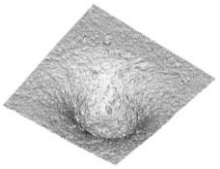
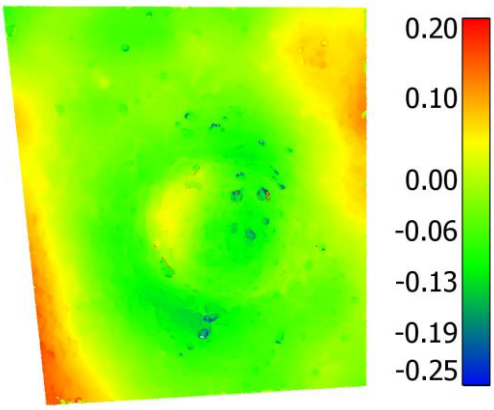
Stereo Image	Depth Map	Reconstruction
		
		
		
Ground Truth		
COMPARISON ***Mean Error*** (scale in mm)		

Figure 105: Reconstructions obtained of the normal ONH target, against the ground truth data.

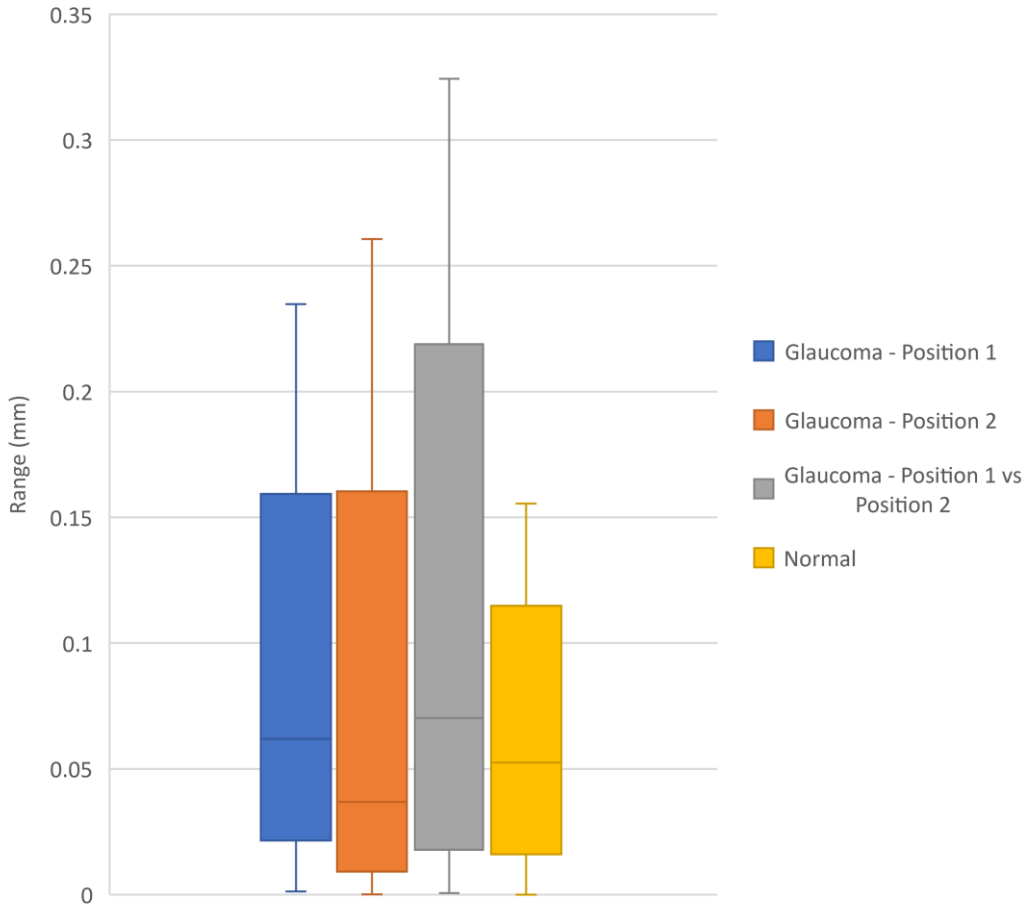


Figure 106: Boxplots of the range values determined in the various comparisons made between the reconstructions obtained.

8.5.2.2 Discussion

Precision: Reconstruction precision where targets were not moved appeared to be good. This is evident in the case where 4 reconstructions of the glaucomatous target held in position 1 were compared, as shown in Figure 100. It can be seen that the depth maps look very similar, but for very minor local variations. Such a degree of similarity is reflected in the range map generated, where the values are minimal in most locations. A similar pattern is seen in the comparison made between the 2 reconstructions of the glaucomatous target in the other position, as is shown in Figure 101, and the 3 reconstructions of the normal target held in one position, as is shown in Figure 102. Moreover, the patterns seen in the range maps generated are of a random nature. It is likely, since the main changing variables were texture related, that the local variations were as a result of different patterns being projected in each case, and therefore also stereo matching.

Where a reconstruction of the glaucomatous target in one position was compared with a reconstruction of it held in another position, as is shown in Figure 103, the precision appears to be less good than in the cases where the target was unmoved. This is evident when looking at the depth maps – they are similar but have more differences than in the other cases. This also appears to be reflected by greater values seen in the range map than in the range maps of the glaucomatous target in fixed locations (as in Figure 100 and Figure 101). The fact that greater values were present is better seen in the boxplot shown in Figure 106, where the situation for the change in position (shown in grey) shows a greater median and larger interquartile range, than the two situations where the position was fixed (shown in blue and orange). Such differences could be as a result of the two variables which changed between these reconstructions – the projection pattern and the location change. Given that the patterns were changed in the other comparisons and better precision was obtained, and the fact that the patterns projected in this case were similar, it indicates that the location change was the greatest contributor to the poorer precision seen. This provides a further clue, alongside the fact that the distortion maps obtained appear abnormal, that calibration error is a significant contributor to the reconstruction error of this system. Since the relevant comparison only involved one reconstruction from each, with differing patterns, and was the only comparison of this type made, further work to try to isolate this contributor and evaluate its effect on performance would likely be beneficial.

The boxplots indicate that reconstruction of the normal target was more precise than reconstruction of the glaucomatous target. Given that the main change between these relates to shape, it indicates that stereo matching performance is different for different shapes, and in this case different surface slopes.

Accuracy: The depth maps obtained of the reconstructions, in both the case of the normal and the glaucomatous target, appear to visually match that for their corresponding ground truth well. Those for the normal target appear to match better than those for the glaucomatous target. In the case of the glaucomatous target, the flat peripheral regions, away from the dimple like structure, appear to not correspond well with the ground truth. In fact, there appears to be an upward curving nature up to the left and right edges of the reconstructions. This pattern can be seen in the mean error maps. It is unclear whether or not the errors seen here are sufficient for this work but do indicate that there is room for improvement.

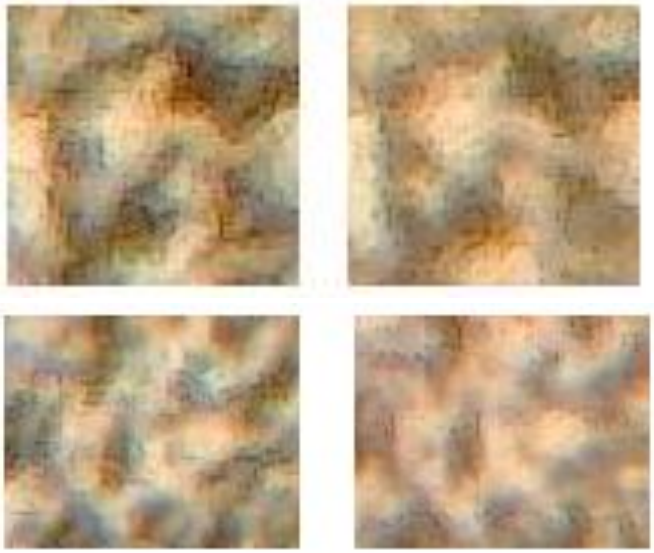


*Figure 107: Two different areas (**ON EACH ROW**) extracted from the left (**LEFT**) and right (**RIGHT**) images of one stereo image pair captured.*

Additional sources of error: Aside from the sources described above related to calibration and stereo matching, image quality, as was the case in the slit lamp work detailed in previous chapters, could have contributed to reconstruction error in this work. Figure 107 shows two examples of corresponding areas between the left and right images of one stereo pair captured in this work. It can be seen in these that there are differences in how the texture appears going from the left image to right image. Such differences likely pose a problem for stereo matching, and lead to errors in the correspondence search. It must be noted that during the setup of the experiment, it was noticed that achieving reasonably balanced focus, and thus resolution, between the left and right images was difficult. Very small lateral movements of the system, in aligning it with the eye phantom, resulted in one of the images being in good focus and the other being in very poor focus. It is not clear whether the same would be experienced with a human eye. However, attempts to improve image quality in future works would likely lead to improvements in the system's performance.

Clinical potential: A comment on clinical potential is made here since this work is the first to test the slit lamp system on a correctly sized eye phantom, allowing for some comparison to results from other 3D imaging technologies. Given the complex nature of early POAG detection it is not possible to state exact precision and accuracy levels which are needed for a system to be able to detect the condition early. However, a very rough indication of the clinical potential of this system can be gleaned from a crude comparison with a system known to be clinically useful. The first version of the HRT system, which has been shown to be able

to detect ONH changes before confirmed defects to vision (Kamal et al., 2000), and is likely the least accurate and precise 3D imaging technology which has been used clinically, represents a known lower limit of clinically useful performance. This system, the HRT I, has an axial and lateral resolution of 125 µm and 12 µm, respectively. In studies on the reproducibility of depth measurements in scans of the optic nerve head region of normal and POAG eyes, an error range spanning from 28.1 µm to 72.8 µm was determined (Dreher et al., 1991). A direct comparison cannot be made to these values since the test protocols were different and the errors in this work were of the closest point rather than vertical depth error. However, the values seen within many regions of the precision range maps for the glaucomatous and normal cases, in Figure 100 and Figure 102, respectively, are on the same order of magnitude as those values stated for the HRT I system. In the accuracy maps, in Figure 104 and Figure 105, the values seen in many regions are also on the same order of magnitude as the repeatability and depth resolution values of the HRT I system but are exceeded on the sloped regions of the cup in the glaucomatous case. This rough comparison suggests that the slit lamp system may not yet be accurate and precise enough to be useful clinically but given that errors are on the same order of magnitude, and the system is at an early stage of development, it indicates that it has potential to become so.

Another aspect which can help to gauge the clinical potential of this system is to look at the theoretically obtainable depth resolution of it and compare this with the depth resolution specified for the other key technologies in this area. Since the resolution of the cameras impacts upon the depth resolution that is obtainable, the calculation (provided in Appendix 8B) made is based on the system in the previous chapter where the higher resolution raspberry pi cameras were used. Using the calibration parameters obtained there, and after accounting for the scale factor of the eye phantom, the depth resolution (per matching accuracy in pixels) is in the range of 73.8 µm pixel^{-1} to 63.6 µm pixel^{-1} over the range of the ONH structure. A range is provided here since, for stereo imaging systems, the depth resolution is not constant over the imaging depth due to the geometry of the triangulation. Moreover, the resolution has been stated in terms of matching accuracy since it is not clear what level of matching accuracy is obtainable, and likely varies depending on the level of texture available at a particular point. If matching accuracy was on the level of 1 pixel then the system would be comparable to the HRT system, which has an axial resolution of 62.5 µm (Balasubramanian et al., 2011), but far from OCT systems, which have axial resolutions in

the range of 20 µm to 5 µm (Aumann et al., 2019). If matching was determined to be accurate to a subpixel level, then an axial resolution closer to that of OCT could be obtainable.

8.5.3 Limitations of This Work

The limitations which were described in previous chapters, which also apply to this trial, are as follows:

- Optical properties of ONH targets [subsubsection 5.6.4.3]
- Shape of the ONH targets [subsection 6.5.3]
- Camera synchronicity [subsection 6.5.3]

As well as these, there are some further limitations or comments to make. Regarding the method of the performance trial, it did allow for some indication to be gleaned as to the contribution of various sources of error, however the indication was subtle. Future work would benefit from a more rigorous trial, one that could better enable the relative contributions of different error sources to be deduced, and better allow for it to be understood whether or not the system can detect the kind of structural differences which occur due to POAG. Moreover, through repeated measures were taken where the target changed position, it would be interesting in future work to also evaluate the effects of roto-translation of the eye phantom, a movement which is likely between repeated measures on a human eye in-vivo.

Regarding the characteristics of the eye phantom: in imaging this phantom, it was clear that image quality was worse than with the previous phantoms. It is not entirely clear what is mostly to blame for this however the optics of the eye phantom may be a significant factor. In particular, given the nature of how the cornea was created, it is likely that its fabrication was imperfect. The shape could have not fully conformed to the ball bearing, and therefore design specifications, the surface quality may have been poor and bubbles, perhaps very small, may have formed within the material if any moisture was present prior to forming. Moreover, the phantoms optical design did not consider aberrations. Future work to further investigate the closeness of the phantom to the human eye in terms of aberrations, and potentially redesign and manufacture it in a different way, may allow for a more valid trial to be conducted.

The synchronicity of the cameras used in this work was not tested and they also had rolling shutters. While not an issue for the eye phantom work, for the reasons detailed in subsection 7.5.3, it may present difficulties for in vivo work and should be considered in future.

8.6 Conclusion

The key targets of this work were to create an eye phantom which was more similar, in terms of optical characteristics, to the human eye, compared with those used in the previous chapters, and to perform a performance trial of the developed imaging system (and algorithms) using the phantom. An eye phantom, which was shown through optical simulation to have a similar axial length, and resolution and distortion, up to a field angle of 20°, to the human eye, was created. A performance trial was performed which indicated good precision where the targets were imaged in the same position, and reasonable accuracy compared to ground truth data. Lower precision was seen between reconstructions of the same target imaged in two different locations, which indicates that calibration errors are a significant contributor to the total reconstruction errors. Further work is required to better investigate and reduce errors, and to better evaluate whether or not the technique has an accuracy and precision sufficient to be useful clinically. The issues which were identified in this chapter, and the previous two chapters, as well as their status, are provided in Table 14.

Table 14: Issues identified until this stage and their current status (chapter 8). ***Changes are shown in bold****.*

Issue Number	Issue	Status
1	Stereo matching technique resulted in quantisation effect	The coarse-to-fine stereo matching algorithm did not suffer from this issue. Moreover, the algorithm used in chapter 7 also did not have this issue.
2	Low image resolution	The cameras used in chapters 6 and 7 (Raspberry Pi Cameras) were of a much higher resolution than those used in the work of chapter 5. **The resolution of the cameras used in this work was lower than that of the raspberry pi cameras but did not appear to be a major source of reconstruction error.**
3	Calibration performance sub-optimal	The reprojection errors appear large and the distortion maps abnormal.
4	Unstable rectification results	The rectification process used here appears to give repeatable rectification outputs.
5	Poor focus in some areas of the images	This is still an issue at this stage.
6	Validation was only qualitative since reconstructions are uncalibrated	Reconstructions in this work were calibrated and quantitative validation was performed.
7	ONH target surface texture different to human ONH	The texture density of the targets in this work was very low, in line with that which is seen in many regions of the human ONH.
8	ONH target optical properties different to human ONH	The targets used were still opaque – different to the human ONH.
9	ONH target shape different to human ONH	The shapes of the targets in this work were more similar to the human ONH, than those used in the work of Chapter 5, but were rotationally symmetrical - different to the human ONH.
10	Illumination not provided from outside the eye	Illumination was provided from outside the eye using the projector-based illumination system.
11	Eye phantom optics different to human eye	**The eye phantom created and used in this work was closer, in terms of optical characteristics, to the human eye compared to the phantoms detailed in chapters 5, 6, and 7. The phantom could likely be improved through different corneal fabrication techniques and different optical designs.**
12	Image capture is not synchronous	The synchronicity of the cameras used was not tested in this work.
13	Stereo matching performance sub-optimal	There are many settings for the stereo matching algorithm developed which were not optimised.
14	Repeated measures not representative of imaging a human eye	**The precision of the method, where the target remained in the same position in the field of view of the cameras, was determined in this work.**
15	Unclear ground truth accuracy and precision	**This work used a highly accurate optical 3D measurement system for the acquisition of ground truth data. The work of the previous chapters would benefit from also using this system for ground truth acquisition.**

CHAPTER 9

INITIAL TRIAL OF SLIT LAMP-BASED RECONSTRUCTION OF THE PORCINE OPTIC NERVE HEAD

9.1 Introduction

This work set out to test whether or not the system and algorithms developed within this project could reconstruct the surface of a biological ONH, to address issue 8 (ONH target optical properties different to human ONH), in this case the ONH of a porcine eye, before a more extensive trial is conducted. As a result of this, this work was very exploratory in nature, testing a number of variables (magnifications, projection patterns, calibration phantom, etc) and does not follow a rigorous and structured method. Only the part of the trial which led to a reasonable reconstruction result is detailed in this chapter.

9.2 Methods

9.2.1 Porcine Eye Suitability

Porcine eyes have many commonalities with human eyes. As a consequence, they have been used in a number of ophthalmological research studies (Middleton, 2010). In relation to this work, the porcine eye contains an ONH which appears not too dissimilar to the human ONH. ONHs of both types of eye are shown in Figure 108. This, combined with the fact that the porcine eye is similar in size to the human eye (Sanchez et al., 2011), led to it being used in this work.

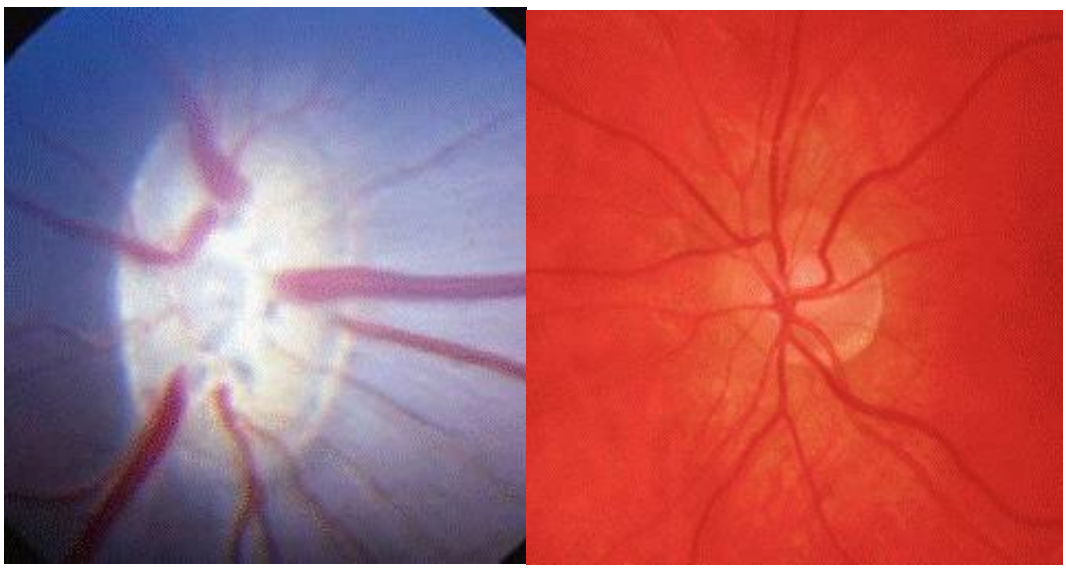


*Figure 108: Porcine (**LEFT**; Ruiz-Ederra et al., 2005) and human (**RIGHT**; Menolotto et al., 2019) optic nerve head.*

9.2.2 Porcine Eye Collection and Transportation

Fourteen enucleated porcine eyes were collected from a local abattoir. Given that biological eyes become cloudy with hours of enucleation, which gradually affects the image quality seen when imaging their retinas, they were collected as soon as possible following enucleation. In order to slow the progression of clouding, the eyes were immediately placed in containers filled with physiological saline solution, which were placed in a larger container filled with ice and water. The eyes were then transported immediately to the laboratory for imaging. Imaging likely began between 1 and 2 hours after enucleation. The eyes were kept within the containers during the imaging session, taking one out at a time for imaging.

9.2.3 Experimental Setup

The imaging system and experimental setup was unchanged from that detailed in the previous chapter (section 8.3 and subsubsection 8.4.2.1, respectively), apart from the fact that instead of an eye phantom, porcine eyes were held in its place. The imaging system was set to be infinity focused initially, allowing it to image an emmetropic eye. In this trial, a 3D printed mount, printed in blue PLA (RS Components Ltd, Corby, Northamptonshire, UK) on the Ultimaker 2+ 3D printer, was secured to the aluminium frame, and allowed for the eyes to be held in position for imaging. The mount featured a dimple shape, allowing the eyes to just sit in position. The magnification of the slit lamp was set to 30 initially. A number of different patterns were tested in this experiment, given the exploratory nature of it. The first to be tested was a striped pattern, therefore the projector was first set to project a striped pattern. A stereo image pair of the ONH of one of the eyes, taken at this magnification is shown at the top of Figure 109. At that setting, the ONH structure appeared larger than what

*Figure 109: Stereo image pairs obtained of the ONHs of porcine eyes with the slit lamp's magnification set to 30 (**TOP**) and 12 (**BOTTOM**).*

was expected. Moreover, the field of view was so narrow that locating the ONH, by rotating the eye, was extremely difficult. Complicating matters was the fact that, while rotating, the eye still had to be kept in the correct lateral and vertical position, within a fine margin, to keep the imaging system aligned with the approximate centre of the pupil, for the ONH to be seen. The magnification of the slit lamp was therefore reduced to 12 for the remainder of the imaging session which made locating the ONH easier. A stereo image captured at the new magnification is shown at the bottom of Figure 109.

9.2.4 ONH IMAGES

The eyes were then imaged, where the eyes had a large enough pupil size, were optically clear enough to be imaged and were unruptured from the enucleation process, with a number of different projection patterns being tested for each. The patterns tried were of different colours and type, such as line and speckle.

9.2.5 CALIBRATION IMAGES

In total 20 stereo calibration images were acquired using the life-sized eye phantom (detailed in the previous chapter: section 8.2). The reason for so few images was that calibration and reconstruction were performed during the trial to test if reconstructions could be obtained, and time was limited due to the progressive clouding of the eyes.

9.2.6 STEREO VISION PIPELINE

The stereo vision algorithm used here was different to the ones used in Chapters 6, 7 and 8 since they failed to return reasonable results, likely due to the poor image and projection quality seen in this trial. As such, it used a stereo matching algorithm which was developed prior to developing the coarse-to-fine algorithm first presented in Chapter 6 (subsubsection 6.4.3.3). The C++ code corresponding to the stereo vision algorithm used here is provided in Appendix 9A.

9.2.6.1 Calibration and Rectification

Calibration and rectification were performed in the same manner as in the previous chapter, using the camera calibration toolbox for MATLAB and the rectification functions of the OpenCV library, respectively. In this case however, given that few calibration images were taken, the coefficients and parameters were not estimated with incremental number of images to gauge if they stabilised. Calibration was performed on 20 images to begin with but

then those images where one or more data points corresponded to a reprojection error of greater than 3.8 pixels in any direction, were removed. 16 images remained after this.

9.2.6.2 Stereo Matching

The stereo matching algorithm used in this work performed only a forward matching step with a fixed block size. Figure 110 has been provided to complement the description of the algorithm that follows. An initial disparity was estimated using a selected large region of interest in the left image (as shown in solid blue in the top of Figure 110). The best matching region (shown with a dashed blue line in the top image of Figure 110) was searched for in the right image using the same template matching method as was detailed in step A in subsubsection 6.4.3.3 where the search was carried out across the whole image, in one dimension, on the same horizontal scanline as the region selected in the left image. 20 x 20 points, within the region of interest, in a grid, were then selected to be matched. The patch size (or support window size) was set to be equal to the point spacing, in the x and y directions. An example showing 8 x 8 points, and corresponding support windows, is shown in the bottom image of Figure 110. Each patch was then matched with a patch of the same

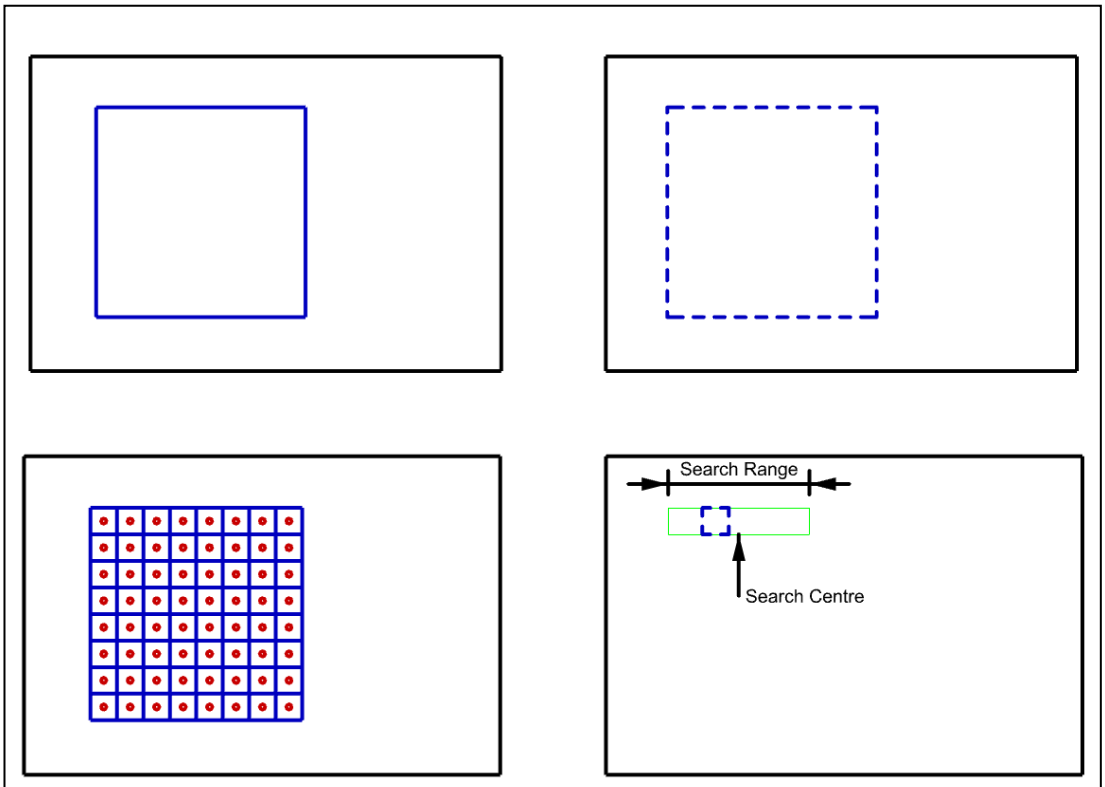


Figure 110: Illustrations to aid the description of the stereo matching algorithm used in this work. ***TOP****: Stereo images with an example region of interest selected in the reference image and potential matching patch in the right image for the search centre estimation step.* ***BOTTOM****: Stereo images with an example of a grid of patches to be matched in the reference image to patches in the right image, with the search range shown in the right image.*

size in the right image (shown in dashed blue in the bottom image of Figure 110) using the match template function used previously. The search range for each was set to be on the same horizontal scanline as the patch to be matched, of a 30-pixel width, and centred at the disparity estimated for the large region of interest initially selected, as shown in green. Sub-pixel disparity values were estimated in the same manner as was detailed in step E in subsubsection 6.4.3.3. The x, y values of the points, and their estimated sub-pixel disparities, were kept for the sparse point cloud where the NCC value associated with such disparity exceeded 0.8.

9.2.6.3 Reprojection, Interpolation and Texture Addition

Reprojection, interpolation and texture addition were performed in the same way as is detailed in subsubsection 7.4.3.3: interpolation was performed by fitting a mesh to the sparse point cloud points, and texture added by reprojecting the pixels of the left rectified image onto the estimated mesh.

9.3 Results and Discussion

9.3.1 Calibration

One of the stereo pairs in the final calibration was disabled automatically by the program because the left and right views were found to be inconsistent. As such, the final calibration coefficients and parameters were estimated on 15 image pairs. Reprojection error plots, shown in Figure 111, show the effect that the removal of bad images had on such errors. In doing so, it reduced the overall mean reprojection error from 1.25 pixels (3SD = 3.33) to 1.11 pixels (3SD = 2.13). The final mean error value and SD is, as could be expected due to their

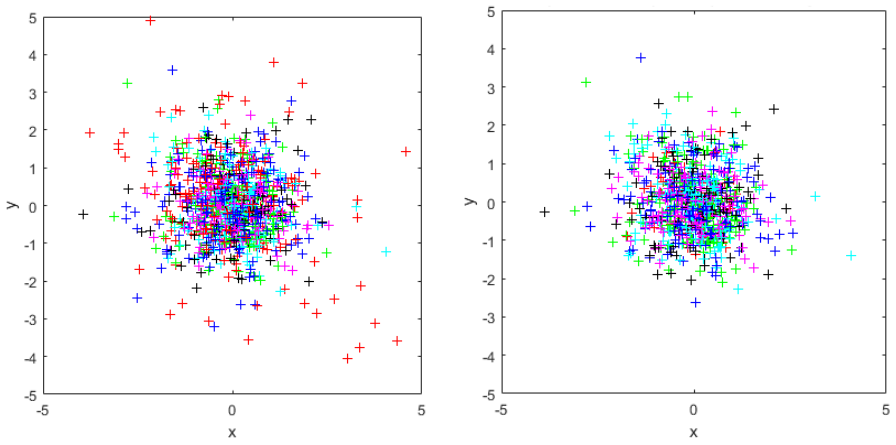


*Figure 111: Reprojection error plots from the calibration before (**LEFT**) and after (**RIGHT**) removing bad images.*

being much fewer images, greater than what was seen in the work of the previous chapter. Other factors could have contributed to this higher error value (e.g., image quality).

The calibration values and associated uncertainties are presented in Table 15. Compared to the uncertainties seen in the previous chapter, those for distortion here are in general lower. This is surprising considering that there are fewer images and that the overall mean reprojection error was seen to be higher. It must be noted that this is not the case for P_1 of the right camera, which shows a very large uncertainty. It could be said however that its value is small and its impact on distortion removal may be negligible. The fact that lower uncertainties were in general seen for the distortion coefficients seems to be reflected in the distortion maps obtained (shown in Figure 112 and Figure 113 for the left and right cameras, respectively). The distortion maps appear more similar to that which would be expected from a typical stereo camera system, compared to the work of the previous chapters, especially in the case of the right camera where the principal point is near the image centre and the distortion profiles look more typical. It is not clear, without investigation, why they look better in this case, it could be as a result of the system being in better alignment in this case. It was again seen that the focal length values had low uncertainties associated with them. The principal points showed a similar pattern. All extrinsic parameters, apart from the x-translation, showed relatively large uncertainties.

Table 15: Calibration coefficients and parameters estimated for the system. SD: Standard Deviation.

	Coefficient/Parameter	Value	Uncertainty (3SD)	% Uncertainty
Left	f_x (pixels)	1637.432	76.0726	4.65
	f_y (pixels)	1615.295	71.70071	4.44
	C_x (pixels)	295.5456	42.64302	14.43
	C_y (pixels)	456.8923	20.43336	4.47
	k_1	-0.76444	0.055744	-7.29
	k_1	0.788245	0.137252	17.41
	k_1	-0.43236	0.129225	-29.89
	P_1	0.010814	0.003164	29.26
	P_2	0.032238	0.01053	32.66
Right	f_x (pixels)	1664.809	72.81879	4.37
	f_y (pixels)	1655.742	71.20666	4.30
	C_x (pixels)	837.2353	7.235874	0.86
	C_y (pixels)	468.5889	9.28185	1.98
	k_1	-1.95552	0.176255	-9.01
	k_1	12.55151	2.495578	19.88
	k_1	-30.7207	9.204393	-29.96
	P_1	0.000213	0.004053	1902.82
	P_2	-0.01149	0.005064	-44.08
	R_1	0.039818	0.011858	29.78
	R_2	-0.21294	0.021548	-10.12
	R_3	0.006726	0.00128	19.03
	T_x (mm)	-1.46057	0.03966	-2.72
	T_y (mm)	-0.0371	0.02858	-77.05
	T_z (mm)	-0.50508	0.187558	-37.13

In this calibration, the low number of images was a limiting factor. Moreover, the poor image quality seen with this phantom, compared to the simple single lens air-filled phantoms used previously, was very likely a major contributor to error. The sources of error mentioned previously in subsection 6.5.1 (corner localisation errors, inaccurately fabricated checkerboard, inaccurate checkerboard square size estimation, sub-optimal camera alignment and inadequacy of distortion models) also apply.

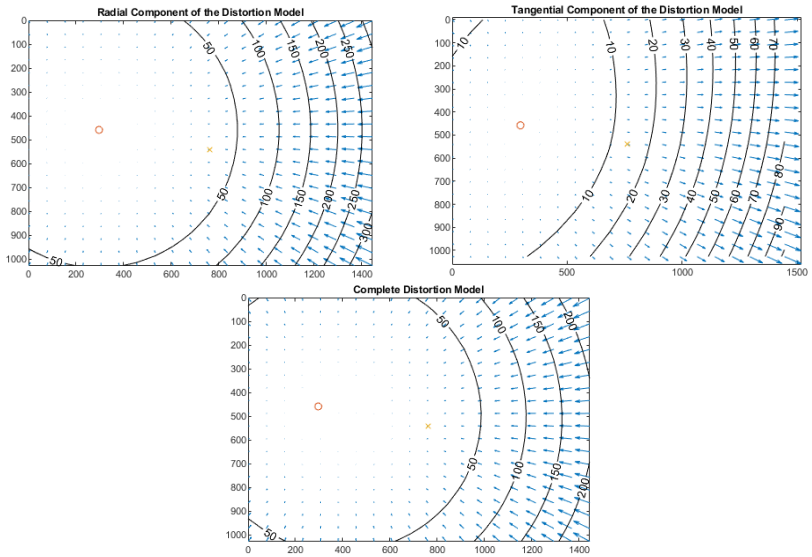


Figure 112: Distortion maps to visualise the distortion parameters estimated for the left camera.

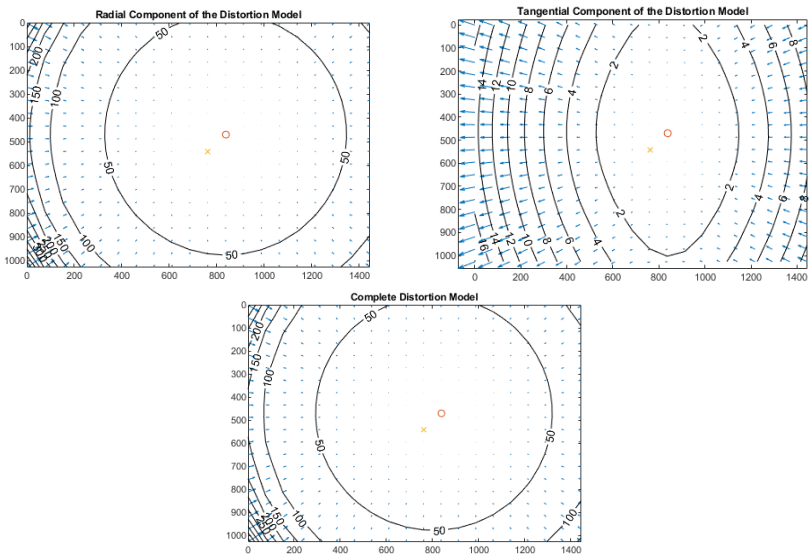


Figure 113: Distortion maps to visualise the distortion parameters estimated for the right camera.

9.3.2 RECONSTRUCTION

9.3.2.1 Results

A number of the porcine eyes were unusable due to their pupil being too small to stereo image, being ruptured during the enucleation process or not being optically clear enough. Moreover, in many of the images taken, the pattern projected did not cover a sufficient area of the ONH structures for reconstruction to be performed (examples are shown in Figure 114). Furthermore, after rectification, many of the images were cropped so heavily by the rectification function that regions very close to the ONH's dimple structure were removed (an example is shown in Figure 115). It can also be seen in this figure that the ONH in the left image looks apparently larger after rectification. It is not entirely clear why this is the case. Further investigation may be needed to identify the cause. In light of these issues, only one of the reconstructions obtained in the porcine eyes can be shown. It is shown in Figure 116. Also shown in this figure are the original and rectified stereo images used to generate the reconstruction. Cross-sectional slices of the reconstruction, obtained by extracting 0.05 mm slices, 0.2 mm apart, from the dense reconstruction, are also shown in the figure.

Figure 114: Stereo image pairs of porcine eye ONHs taken in this trial showing the projection pattern not fully covering the ONH structure.

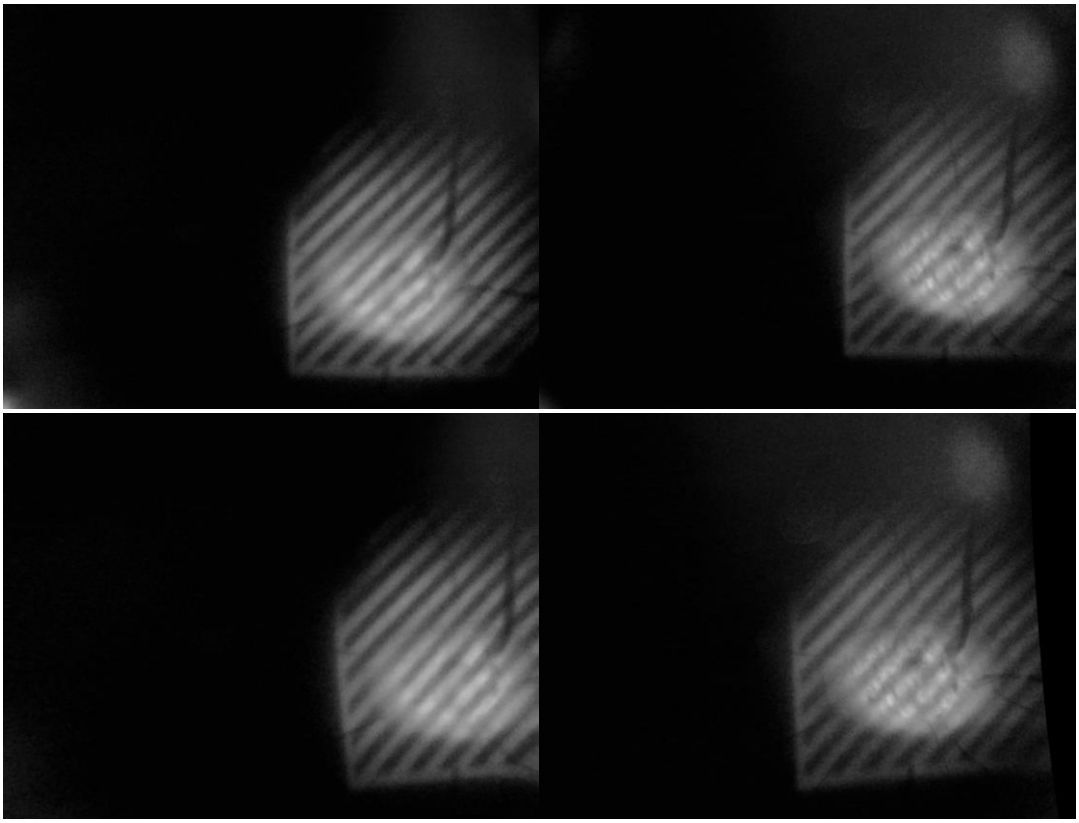


*Figure 115: Stereo image pair of one of the porcine eye's ONHs before (**TOP**) and after rectification (**BOTTOM**), taken in this trial.*

9.3.2.2 Discussion

Reconstruction shape: It appears that in the reconstruction that was obtained, the dimple-like ONH structure was recovered. Moreover, although not very easy to see, the shape of that structure appears to be very roughly similar to that which can be seen when viewing the structure stereoscopically through the rectified stereo images. When viewed stereoscopically, it can be seen that the slope of the dimple-like structure is relatively greater in its top left portion as compared to the other sections, which appear less steep. This pattern is reflected in the reconstruction of it, as is more apparent when viewing the colour heightmap textured 3D reconstruction.

Reconstruction size: The size of the ONH structure in the 3D reconstruction appears to be not too far from what would be considered reasonable. In order to roughly gauge its size, a line was drawn on the structure, as can be seen on the dense textured reconstruction in Figure 116, connecting two points opposite each other on the perceived edge of it, and its length measured using the CloudCompare software. While not very precise, this length should be a very rough estimation of the diameter of the ONH's disc. The distance of the line

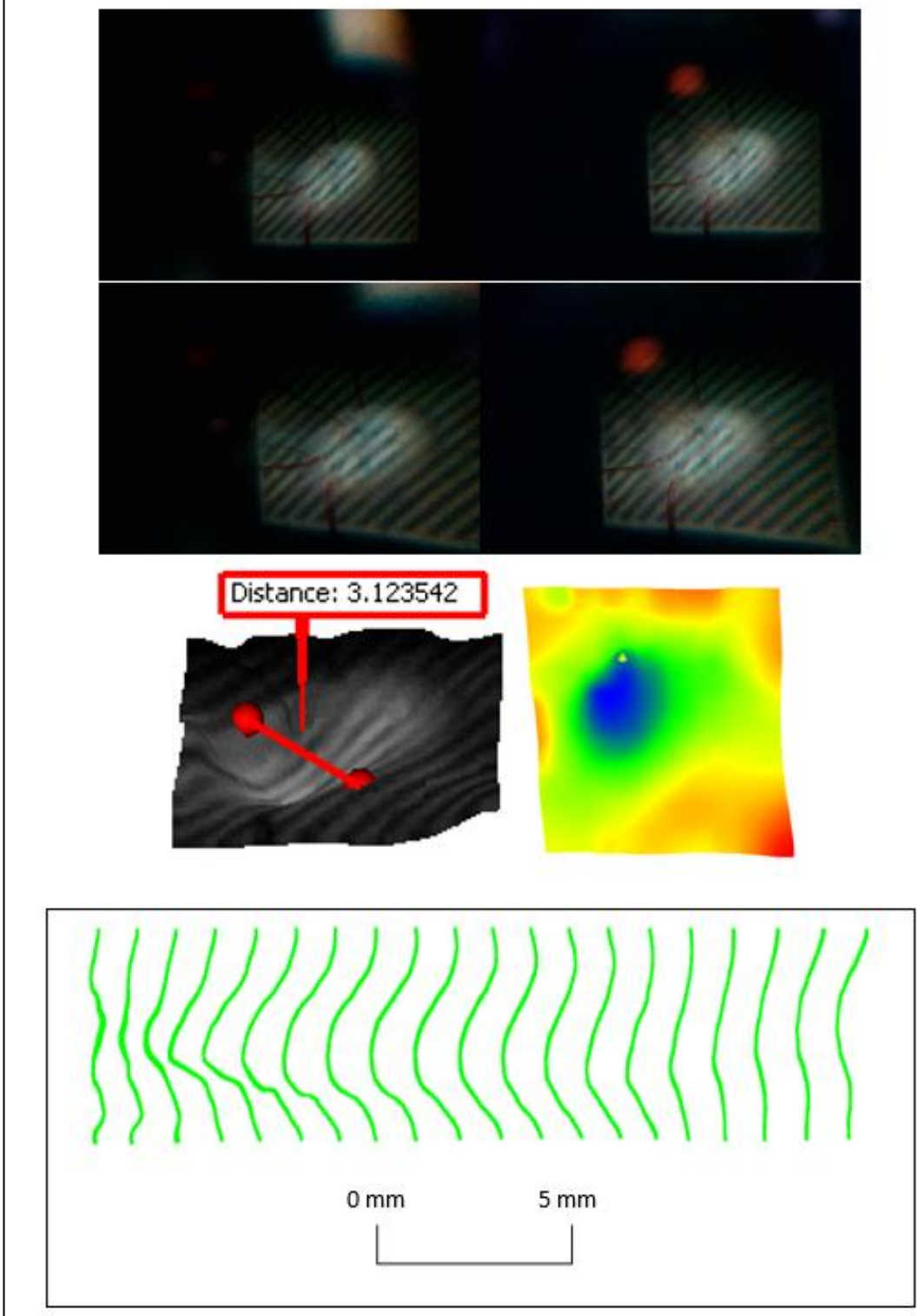


Figure 116: The stereo images and 3D reconstructions obtained of the ONH of a porcine eye in this trial. ***TOP TO BOTTOM****: Input stereo image pair; rectified stereo image pair; dense textured (left) and colour heightmap textured 3D reconstruction (right); and cross-sectional slices, created by extracting 0.05 mm slices, 0.2 mm apart, from the dense 3D reconstruction.*

was measured to be approximately 3 mm. A value for the optic disc diameter of the porcine eye could not be found in the literature for comparison. If it were assumed to be similar to that in the human eye, which is 1.85 mm to 1.95 mm on average, but in a wide range from 0.95 mm to 2.9 mm (Schacknow & Samples, 2010), it indicates that the reconstruction could be of roughly the correct size. If the ONH of the porcine eye was larger, this would explain why it appeared larger than expected, which led to the magnification needing to be changed. The other possibility is that the size was in fact overestimated. A potential reason for overestimation could be related to the axial length. It could be the case that the axial length of the porcine eye was less than that of a typical human eye, which is what the eye phantom used for calibration was designed to mimic, making its ONH appear larger. This was the case for Shields and colleagues in their eye phantom work with the Optic Nerve Head Analyser (Shields et al., 1989). They found that when the axial length was less than that of the typical adult human eye, for which the device would have been calibrated for, the program overestimated the size of the ONH, while underestimating it when the axial length was greater. They were able to correct for these errors by applying a correction based on the axial length. It would therefore likely make sense in future work, to also consider measuring axial length.

At this stage of the work, clinically useful parameters such as the cup-to-disc ratio, or volume parameters, were not determined. In order to do so, the cup margins would need to be determined which is not straight forward. In the HRT system, the cup margin is defined as where a reference plane, placed 50 µm below the retinal surface, intersects the structure. In order to be able to do this for this work, a definition of the cup margin would need to be decided upon and software created to perform the task. It would be useful at some stage in this work to implement such features.

Stereo matching and projection/image quality: This work gave an idea of how the projection would look, in terms of sharpness, on a biological ONH. It was seen that it was less sharp than what was seen on the ONH targets of the eye phantoms, particularly in the cup region. Comparing some of the images obtained in this trial, as are shown on the top row of Figure 117, with some obtained in the trial detailed in the previous two chapters, as are shown on the bottom row of Figure 117, indicates this. This could be due to a number of factors such as poor focus or clouding, or could be as a result of the biological ONH having different reflectance properties due to its inner layers being semi-transparent (Shields et al., 1989).

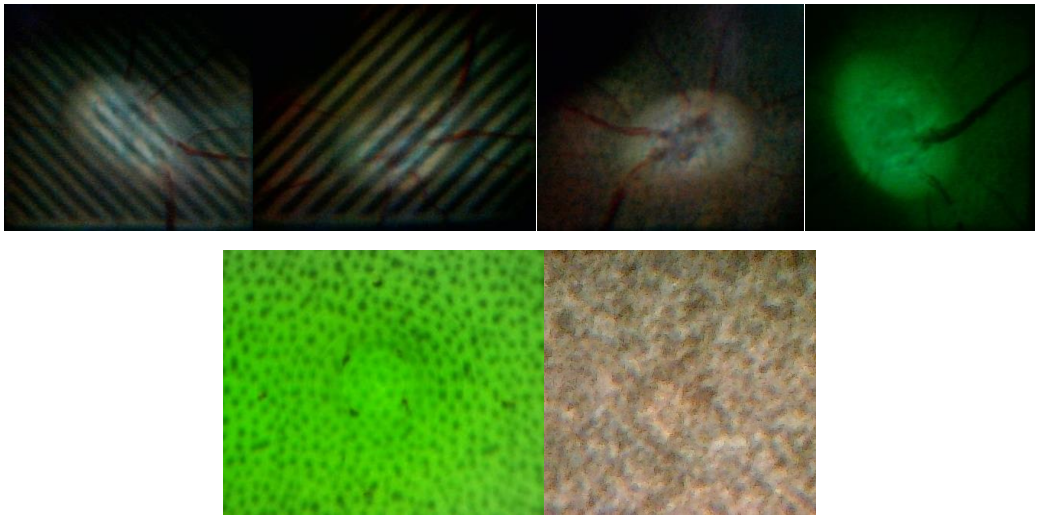


*Figure 117: Images obtained in this trial (**TOP**) and the trials detailed in the chapters 7 and 8 (**BOTTOM**).*

The sharpness limited the fineness of the pattern that could be imaged. This led to large support windows being used in stereo matching. In order to be able to use smaller patches, and therefore likely better reconstruct the structure, future work to try to improve pattern sharpness and project finer patterns, would likely be beneficial. Such improvements could be made by improving the optical design of the projection system and through optimisation of the projection colour. Regarding colour, it is known that shorter wavelengths tend to reflect more from the inner layers of the retina rather than longer wavelengths (Lee, 2019). Given this, green patterns were tested in this work however they resulted in dull and poor-quality images, as is evident in the top right example of Figure 117. It was for this reason that white patterns were also tried. It will likely be useful in future to explore this aspect further.

Imaging and projection issues: The issues experienced where the projection would not cover the ONH region well needs to be addressed before this type of imaging work is repeated.

9.3.3 Limitations of This Work

The core limitations that pertain to this trial are the lack of a ground truth and that the procedure followed did not allow precision to be gauged, which could have been assessed despite the lack of a ground truth. The lack of the ground truth prohibited the accuracy of reconstruction from being explored, and the fact that repeat imaging of specific eyes was not performed meant that precision could not be gauged.

The main reason for the lack of ground truth data was that access to an OCT machine for collection of such data was not possible at the time of the trial. The trial was preliminary

work, performed to evaluate if reconstructions could be obtained at all before the process of gaining access to an OCT machine was initiated. The preliminary nature of the work, and the lack of time available to conduct the study (limited as a result of the progressive clouding of the eyes), were the main reasons that repeat imaging was not performed.

A further limitation of this work is that there are differences between the ONH and eye's optical properties between porcine and human eyes.

9.4 Conclusion

The key target of this work was to test whether or not reconstruction of a biological ONH in a biological eye, which is similar to the human eye, could be obtained using this system and the associated stereo vision algorithms. It was shown, albeit only for one case, that reconstruction is possible but that more work is needed to improve image and projection quality, and thus hopefully reconstruction quality. Such improvements need to occur before considering moving to the next stage of accessing an OCT system to obtain ground truth data for comparison. The issues which were identified in this chapter and the previous three chapters, as well as their status, are provided in Table 16.

Table 16: Issues identified until this stage and their current status (chapter 9). ***Changes are shown in bold.***

Issue No.	Issue	Status
1	Stereo matching technique resulted in quantisation effect	The coarse-to-fine stereo matching algorithm did not suffer from this issue. Moreover, the algorithm used in chapter 7 also did not have this issue.
2	Low image resolution	The cameras used in chapters 6 and 7 (Raspberry Pi Cameras) were of a much higher resolution than those used in the work of chapter 5. The resolution of the cameras used in this work was lower than that of the raspberry pi cameras but did not appear to be a major source of reconstruction error.
3	Calibration performance sub-optimal	**The reprojection errors appeared large and the distortion maps abnormal in the previous chapters. Errors were still large in this trial however the distortion maps were less abnormal.**
4	Unstable rectification results	**Rectification was unstable using the process in chapter 5, but that implemented after (in chapter 6) does not suffer from such issues.**
5	Poor focus in some areas of the images	This is still an issue at this stage.
6	Validation was only qualitative since reconstructions are uncalibrated	**Reconstructions in chapters 6, 7 and 8 were calibrated and quantitative validation was performed. In this work the reconstructions were calibrated but quantitative validation could not be performed due to the lack of ground truth data.**
7	ONH target surface texture different to human ONH	**The texture density of the targets in the work of chapter 8 was very low, in line with that which is seen in many regions of the human ONH. Since this work was on a biological ONH, texture density was also very similar to the human ONH.**
8	ONH target optical properties different to human ONH	**The targets used in chapters 5, 6, 7 and 8 were opaque – different to the human ONH. In this work, a biological ONH was imaged and so this did not apply.**
9	ONH target shape different to human ONH	**The shapes of the targets in the work of chapters 6, 7, 8 and 9 were more similar to the human ONH, than those used in the work of Chapter 5, but were rotationally symmetrical - different to the human ONH. In this work, a biological ONH was imaged and so this did not apply.**
10	Illumination not provided from outside the eye	Illumination was provided from outside the eye using the projector-based illumination system.
11	Eye phantom optics different to human eye	**The eye phantom created and used in chapter 8 was closer, in terms of optical characteristics, to the human eye compared to the phantoms detailed in chapters 5, 6, and 7. The phantom could likely be improved through different corneal fabrication techniques and different optical designs. Regarding this work, the porcine eye is similar to the human eye in terms of size and optical configuration, but it is not entirely clear by how much they differ from the human eye.**
12	Image capture is not synchronous	The synchronicity of the cameras used was not tested in this work.
13	Stereo matching performance sub-optimal	There are many settings for the stereo matching algorithm developed which were not optimised.
14	Repeated measures not representative of imaging a human eye	**The precision of the method, where the target remained in the same position in the field of view of the cameras, was determined in the work of chapter 8. In this work, given that few reconstructions could be obtained, this was not possible.**
15	Unclear ground truth accuracy and precision	The work of chapter 8 used a highly accurate optical 3D measurement system for the acquisition of ground truth data. The work of the preceding chapters would benefit from also using this system for ground truth acquisition.
16	**Projection quality on biological ONHs needs to be improved**	**Projection quality on the ONHs of the porcine eyes was very poor. More work is needed to improve this before a study is set up where an OCT system is accessed to acquired ground truth data.**

CHAPTER 10

INITIAL FEASIBILITY OF A HANDHELD STEREO FUNDUS CAMERA FOR RECONSTRUCTION BY CALIBRATED RECONSTRUCTION IN AN EYE PHANTOM

10.1 Introduction

This chapter presents the initial work that was carried out to realise a low-cost handheld stereo fundus camera for use in stereo reconstruction of the ONH. In this, an optical design was carried out, a test setup created and an initial trial in an eye phantom run to test its performance. It must be noted that at this stage the system did not take the form of a handheld unit. The system created was for testing of the optical configuration only. This was done to allow for the optical design to be iteratively tested and developed before it is fixed and a handheld unit, featuring elements such as a battery and display, created.

The design created had a basic setup consisting of two low-cost cameras; a right-angled prism to create the stereo separation; an indirect lens; and a projection system to project texture onto the ONH during imaging. The stereo vision algorithms used here were from the slit lamp work. The performance trial was carried out on textureless ONH targets in a modified version of the cage system eye phantom presented in section 7.2. A series of ONH targets were created, with increasingly excavated dimple structures, to attempt to mimic the kind of structural changes that would occur over time with POAG, to get an indication on whether the system, in a simplified and controlled situation, could allow for detection of such changes.

10.2 System Optical Design

10.2.1 General Optical Configuration

This system uses the technique of indirect ophthalmoscopy to image the retina. The optical configuration of the system is depicted in Figure 118. In this figure the illumination component is shown to be in the same plane as the imaging component so that they could both be shown in 2D on the page. In reality the illumination system would be coming out of the page (i.e., perpendicular to the imaging component).

The design shows that a 20 D Lens (OICO, London, UK) was used to form an aerial image of the retina behind it. This image is then conjugated, using M12 camera lenses with focal lengths of 8 mm (Hongkong Ansice Industrial Co, Ltd, Hong Kong, China), to image sensors so that images of the retina can be obtained. The image sensors used were Sony IMX219 sensors which are used within the Raspberry Pi V2.1 Camera (Raspberry Pi Foundation, Cambridge, UK) and have a size of 3.68 x 2.76 mm. Regarding stereo separation of the cameras, they are

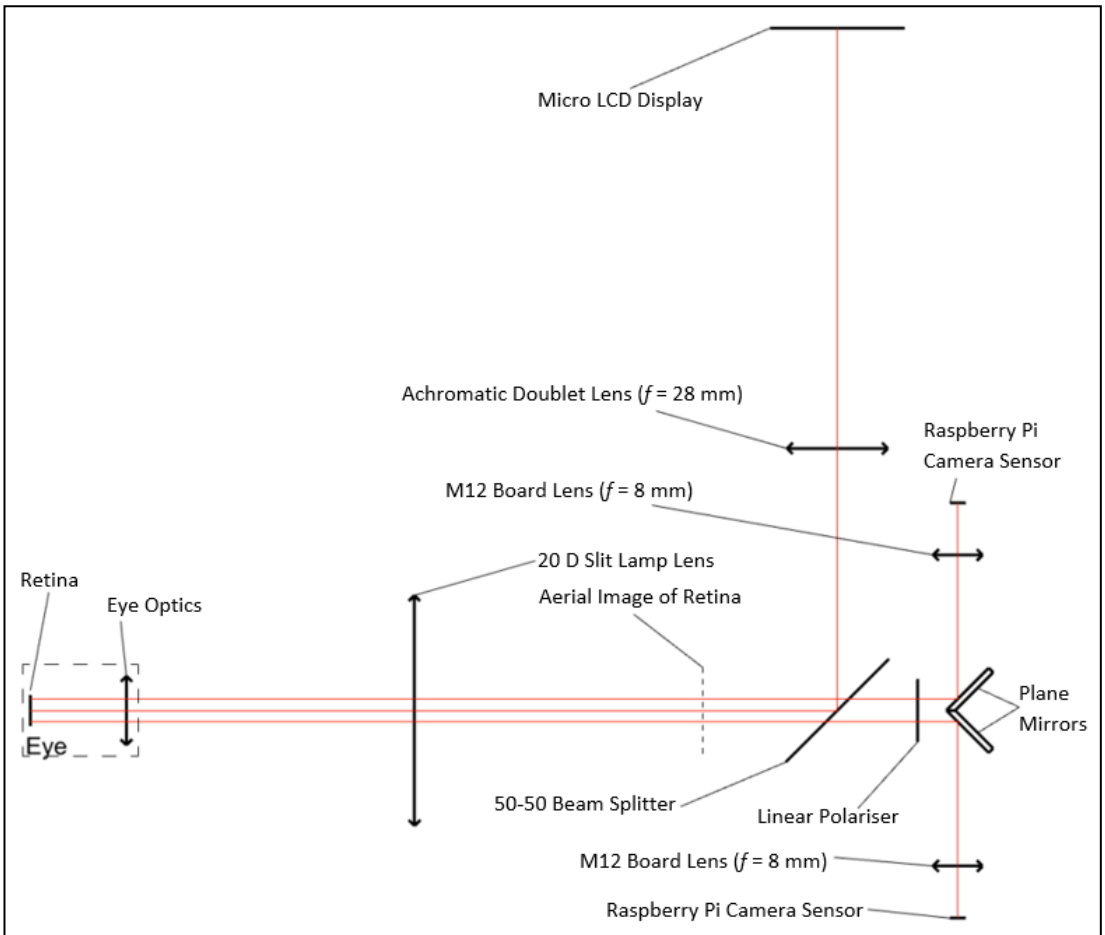


Figure 118: Schematic of the complete setup, combining both the imaging and illumination components. It must be noted that the schematic has been drawn in such a way so that it can be displayed two dimensionally. In actuality, the illumination system should be perpendicular to the plane of the page.

too large to be able to be placed side-by-side and be close enough to both be able to image an overlapping region on the retina. The separation required is on the order of 1 to 4 mm. Consequently, a right-angled mirror is exploited, with the configuration shown in Figure 118, to create the stereo separation. At this stage of the work, it was not possible to obtain a right-angled mirror, or an aluminised right-angled prism effectively, and so two plane mirrors (16MV10, Comar Optics Ltd, Linton, UK) were placed at 45° to each other to create this. In order to be able to get the mirrors to act as a right-angled mirror, one of the long edges of each needed to be sanded to a 45° angle.

In order to allow patterns to be projected onto the retina, a small liquid crystal display (LCD; Pimoroni Ltd, Sheffield, UK) was used. The display featured a 28 x 28 mm screen and had a resolution of 240 by 240 pixels. The screen was conjugated to the aerial image using an achromatic doublet lens with a focal length of 28 mm (28DC17, Comar Optics Ltd, Linton, UK). A beam splitter (25BN16, Comar Optics Ltd, Linton, UK) was used to direct the downward

projection from the LCD screen and doublet, towards the eye, and with a path parallel and close to the paths of the cameras.

In order to mitigate the amount of specular reflection reaching the image sensors, a linear polariser (Comar Optics Ltd, Linton, UK) was placed between the beam splitter and the right-angled mirror. A linear polariser was not required on the illumination component since the LCD screen already produces linearly polarised light.

10.2.2 Geometrical Optical Design

At this stage, the selection of optical elements and their spacing was carried out through a basic geometrical optical design, where lens thickness was ignored to simplify matters. No real consideration was made regarding aberrations at this stage since for many of the elements, such as the camera lenses and indirect lens, the optical specifications are not available from the manufacturers, prohibiting simulation in an optical design software package. The distances between the different elements, shown separately for the illumination and imaging components, are shown in the bottom and top left cells of Figure 119, respectively. It must be noted that this schematic does not show how the stereo separation was created, and features only one camera, for the purpose of clarity. Regarding the imaging component, the elements were placed such that a 7 mm by 7 mm square region on the retina (of a life-sized eye) would cast to a size of 2.76 mm by 2.76 mm on the image sensor. The coverage of this on the image sensor is shown as a solid line box in the top right cell of Figure 119. Also shown on this is a circle which would be cast onto the sensor from a circle with a diameter of 2 mm on the retina, to give an idea of the size that an ONH with a disc diameter of approximately 2 mm would take up in the resulting image. Regarding the illumination component, the elements were placed such that the entire screen would be cast onto a 5.4 x 5.4 mm region on the retina (of a life-sized eye). The resulting size of the projection is depicted as a dashed line box in the top right cell of Figure 119.

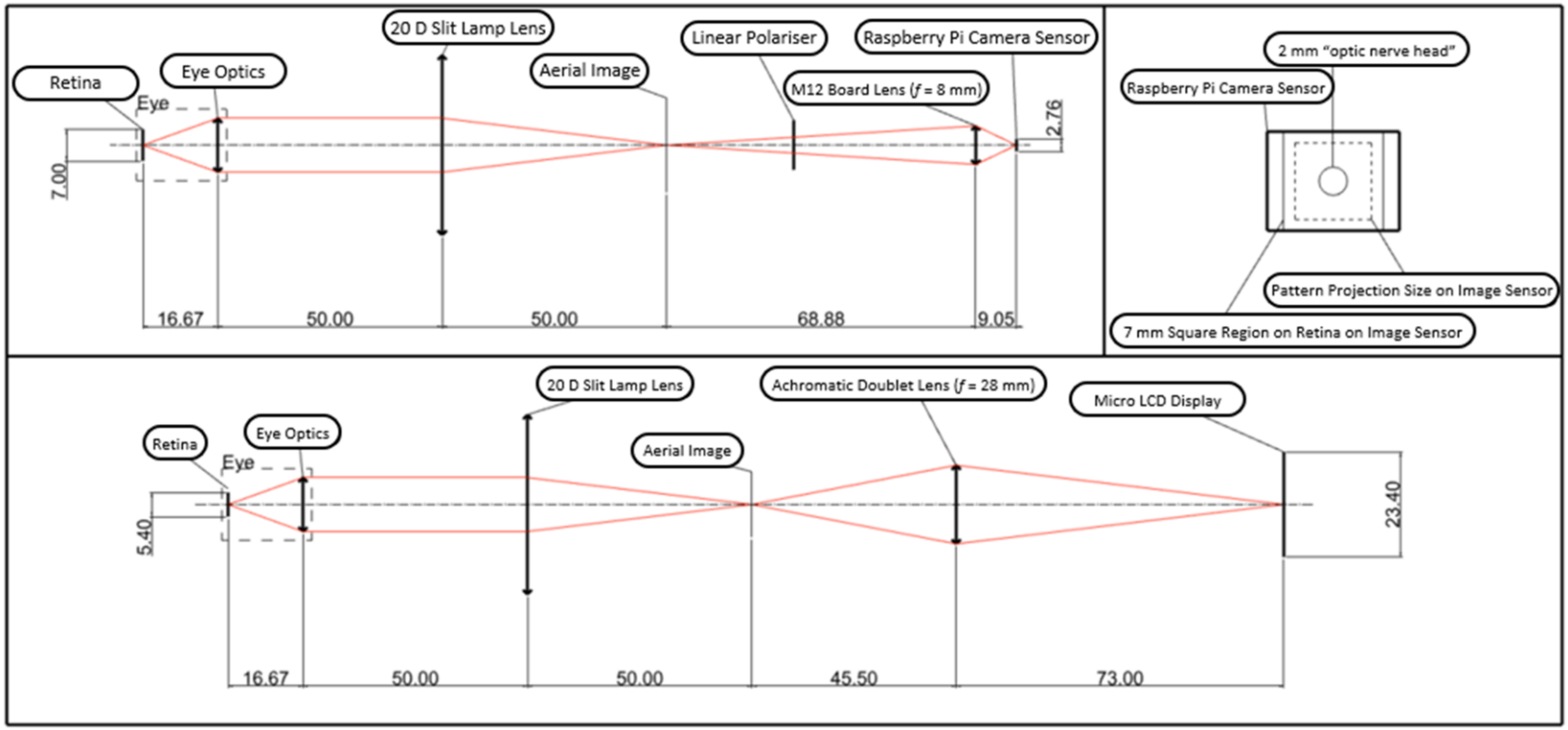


*Figure 119: Schematics of the optical configuration of the handheld imaging device, with the imaging (**TOP LEFT CELL**) and illumination systems (**BOTTOM CELL**) separated for clarity. It must be noted that the schematic of the imaging system does not show how the stereo separation was created, and features only one camera, also for the purpose of clarity. Also shown (**TOP RIGHT CELL**) is relative sizes of a 2 mm optic disc and the patterned illumination in the images captured of a life-sized eye using this design for the device.*

10.3 Experimental Setup

10.3.1 Brief Overall Description

CAD rendered views of the experimental setup, both full and sectional views, are provided in Figure 120, and photographs of it from different angles can be found in Figure 121. The various optical elements and electronic components of the system were held using mounts printed in black ecoPLA on the Creality Ender 5 FDM 3D Printer (Creality 3D, Shenzhen, China). The design featured a linear track onto which parts were bolted, allowing for their position to be varied. Also fixed to this was the eye phantom. It was fixed to this to ensure good alignment with the system.

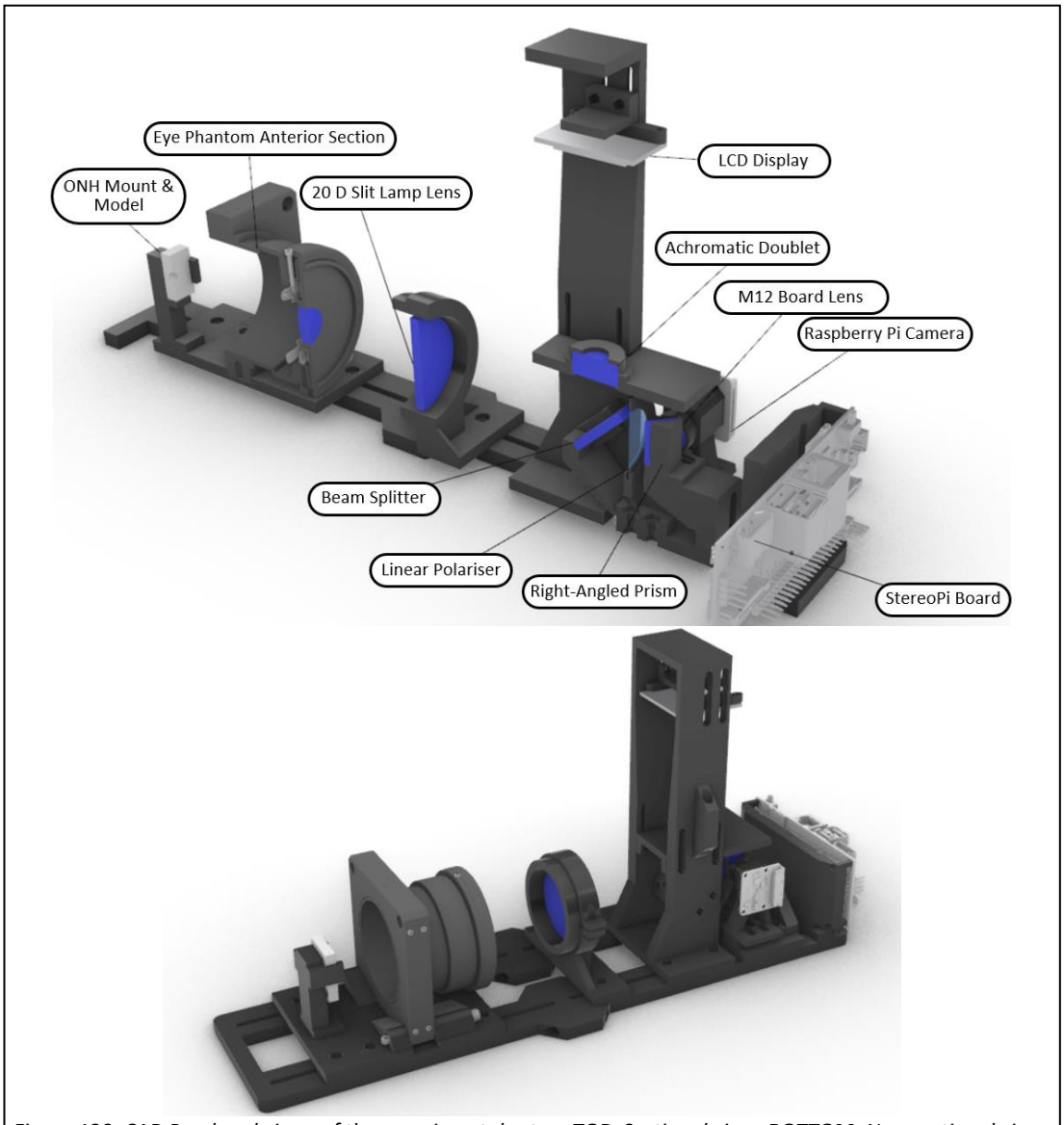


Figure 120: CAD Rendered views of the experimental setup. ***TOP****: Sectional view.* ***BOTTOM****: Non-sectional view.*

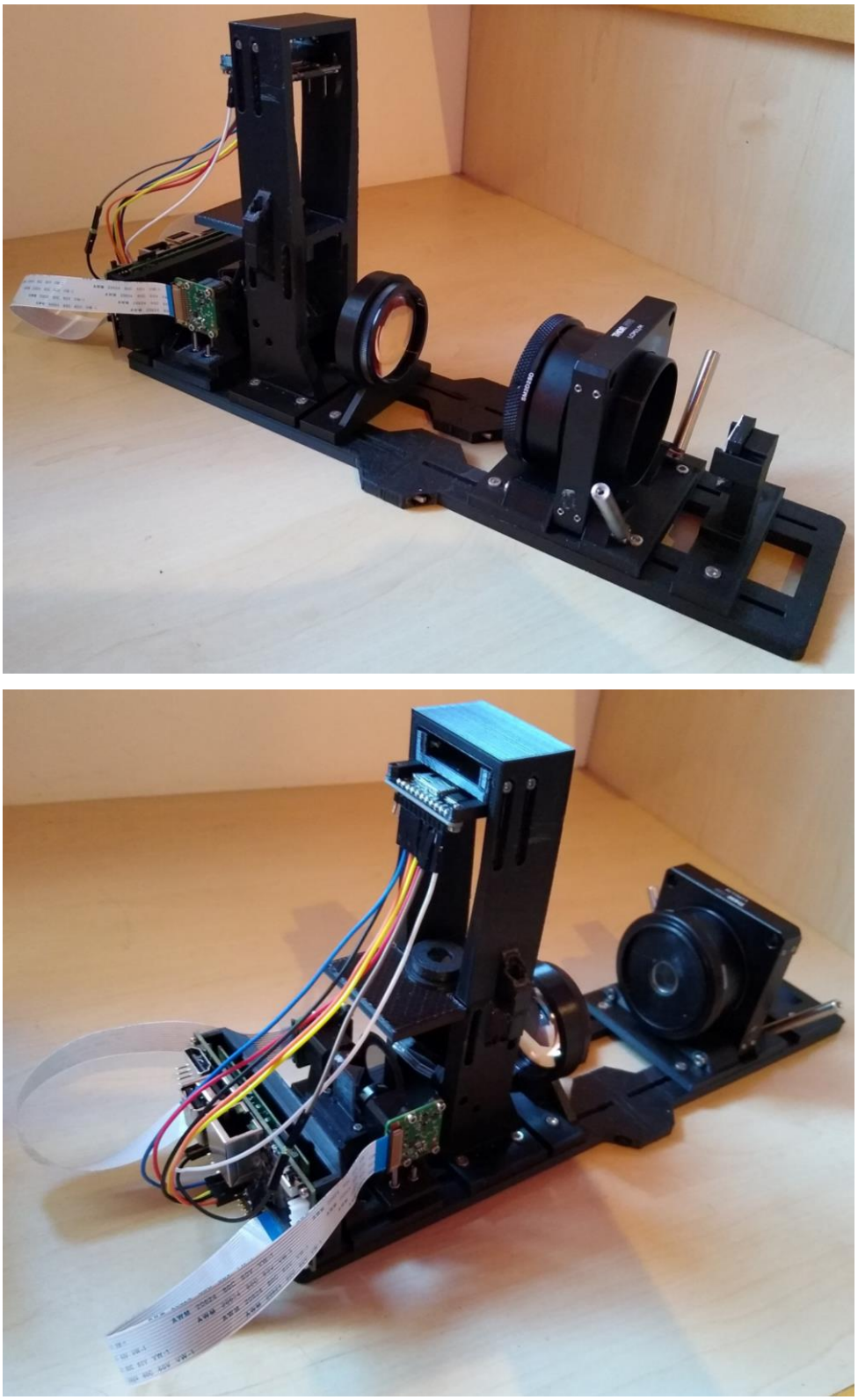


Figure 121: Photographs of the experimental setup.

10.3.2 Handheld System

On the linear track, the first part of the handheld system, was the 20 D slit lamp lens. It was held using a 3D printed mount which was itself bolted to the linear track. The mount held the lens in place by camping it down, where a bolt was used to tighten its grip.

After this part, the next on the linear track was a tower which held the different elements of the illumination component in place. The different elements were held using 3D printed mounts which bolted onto the two side walls of the tower. At the top of the tower, the LCD display was held. Vertical slots allowed for the display to the raised or lowered, to allow for the distance between it and the other elements to be varied. The achromatic doublet lens was held below the display. The tower also featured vertical slots at this point to allow for its height to be varied. The lens was placed with its convex side face up, towards the screen, and a 3D printed aperture, with a diameter of 9.7 mm, was placed on the doublet. In order to prevent light from reaching the right-angled prism and cameras directly from the LCD display, the mount for the doublet was extended to cover that area and block it from the screen. Below the achromatic doublet, the beam splitter was held, at an angle of 45°. It was placed such that the surface with the beam splitting coating was on top. The position and rotation of the beam splitter could not be adjusted – it was fixed in place. In order to mitigate reflection from the surface below the beam splitter, and prevent the light passing through the beam splitter from reflecting back up and then towards the cameras, the surface below was designed to be sloped and was painted with a highly absorbent black paint, black 2.0 (Culture Hustle, London, UK).

After the illumination tower, the next part on the linear track was a platform which held the cameras, right-angled mirror and polariser. An assembled and exploded 3D CAD rendered view of this part is provided in Figure 122. The linear polariser material was glued, using super glue (Loctite, Düsseldorf, Germany), to a 3D printed ring. The ring was placed in a 3D printed mount which itself bolted onto the main platform of this section. The design allowed for the ring to be turned to change the angle of polarisation. The raspberry pi cameras, with stock lenses removed, and the M12 lenses, were fit to 3D printed mounts, which also bolted onto the main platform. The mounts contained slots to allow for the distance between the cameras and right-angled mirror to be varied. In order to create the right-angled prism, the two plane mirrors, with a long edge sanded to be at 45°, were glued, using super glue, onto a 3D printed piece, with the mirror coated faces facing outward. This mount was bolted onto

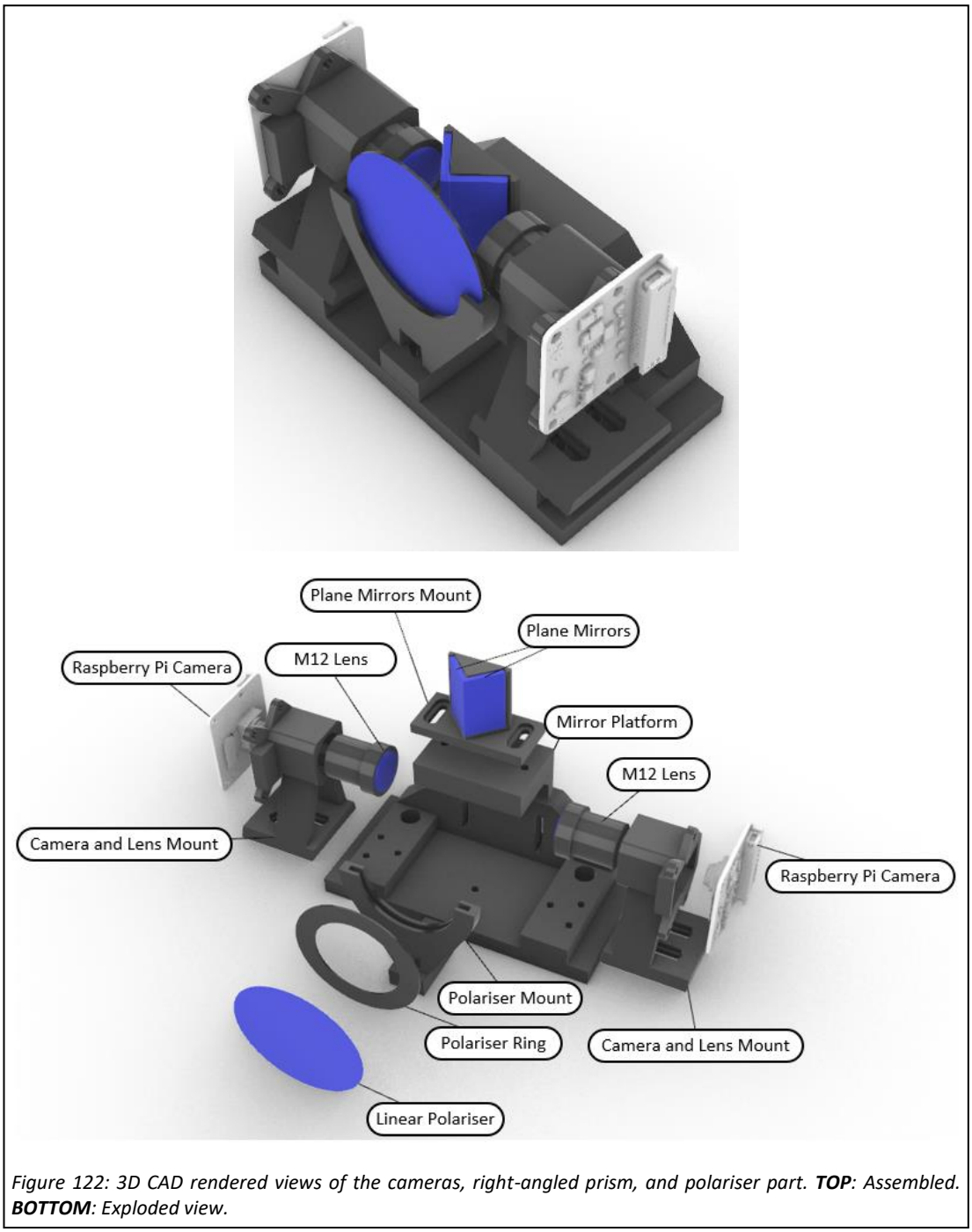


Figure 122: 3D CAD rendered views of the cameras, right-angled prism, and polariser part. ***TOP****: Assembled.* ***BOTTOM****: Exploded view.*

a smaller 3D printed platform, which was itself bolted to the main platform of this part. Slots in the two platforms made it possible for the right-angled mirror to be moved up and down and backwards and forwards.

At the back of the linear track, a StereoPi board was mounted using a 3D printed mount. The cameras were connected to this using ribbon cables, and the LCD display was connected to its general-purpose input-output (GPIO) pins using jumper wires (B071JDZXTH, Amazon .com, Inc, Seattle, Washington, US). The board was used to control the cameras and the LCD

display. A PC was connected to it using an ethernet cable so that commands could be sent to it. Scripts were written in the Python programming language to allow for a video stream from the cameras to be seen on a monitor and for still image capture to be initiated by a button press on a keyboard.

10.3.3 Eye Phantom

At the front end of the setup, the eye phantom was held. The eye phantom used in this work used parts of the cage system eye phantom, which is detailed in section 7.2. The anterior section of this phantom was used but the posterior section was changed to allow for ONH targets to be interchanged more easily and for it to be used for calibration without the need to remove a cage plate. The anterior section was held on a 3D printed platform which was itself bolted onto the linear track. A separate platform, also bolted to the linear track, was used for mounting of the ONH targets. A mount for the ONH targets was magnetically attached to this other platform, using four neodymium magnets, each 2 mm in diameter and 1 mm thick. The ONH targets were press fit into the ONH target mount.

10.3.3.1 ONH Targets

A series of ONH targets, with white top surfaces, were created. CAD rendered cross-sections and photographs of each of these are shown in Figure 123. These were very similar to those used before, as presented in subsubsection 7.2.3.2, however in this case they have a shape which is characteristic of increasing glaucomatous damage.

The targets were created in a similar manner to what is detailed in subsubsection 6.2.2: a half-sectional profile was created, scaled appropriately, revolved around an axis, embedded into a square prism, 3D printed and then painted. The half-sectional profiles used are shown

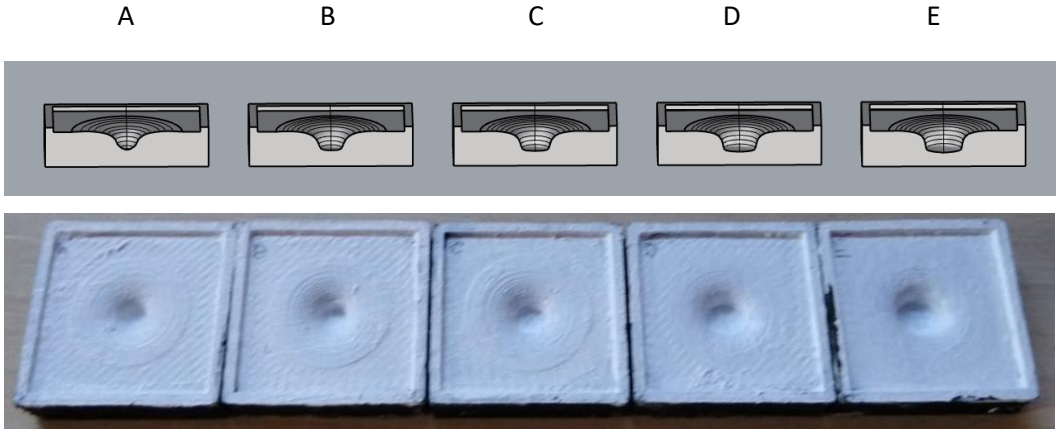


Figure 123: Series of ONH targets created for this work.

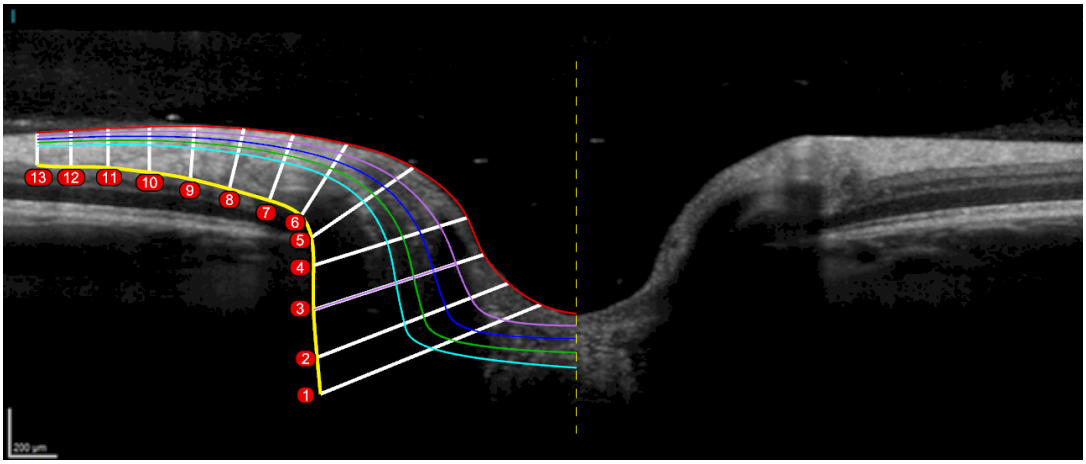


Figure 124: Half-sectional profiles used to design the series of ONH targets created for this work. ***Red*** *– no damage,* ***Purple*** *– 10% damage,* ***Blue*** *– 20% damage,* ***Green*** *– 30% damage,* ***Cyan*** *– 40% damage. (Heidelberg Engineering, n.d.).*

as different coloured lines in Figure 124. They were created with the intention of being characteristic of progression from 0% axonal loss to 40%, in increments of 10%. The reason for this is that visual symptoms are only detectable after around 30-50% of RGCs, and therefore axons in the RNFL, have been lost (Alencar & Medeiros, 2011). The target characterising no damage was created by tracing the ILM of the OCT image shown in Figure 124. In order to create the other targets, the RNFL was manually segmented, shown as the region bounded by the yellow (solid) and red lines. Lines, shown in white, were drawn to be roughly perpendicular to the direction of axons within the RNFL. These lines were then revolved around the yellow dotted line (the rough ONH centre). The resulting surfaces have cross-sectional areas equal to those in the RNFL at the ONH. Each surface was reduced in area, such that the cross-sectional areas were decreased by the various damage percentages used. The cross-sections at each stage were used to construct the half-sectional lines, to be used for revolution into a surface. The lines corresponding to 10, 20, 30 and 40% damage are shown in purple, blue, green, and cyan, respectively, in Figure 124. As a rough guide for the validity of this method to design the targets' profiles, the profile corresponding to 40% damage (cyan) looks similar to the profile of the ILM in a damaged ONH, as is shown in Figure 125.

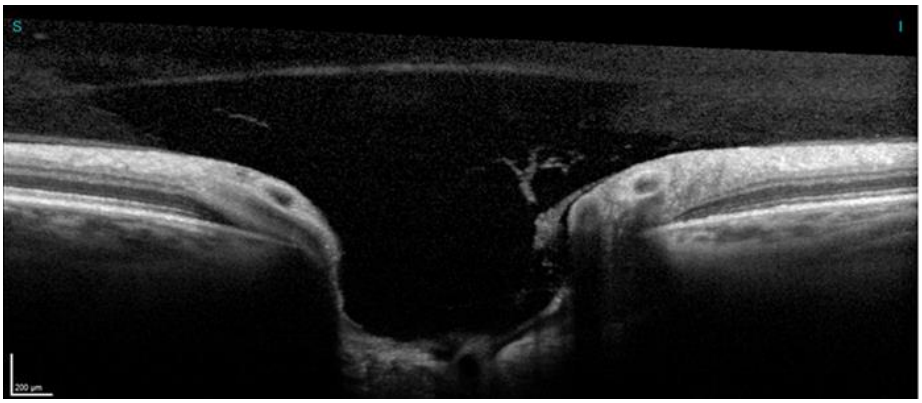


Figure 125: OCT b-scan of an ONH with glaucomatous damage. (Heidelberg Engineering, n.d.).

Once the profiles were created, they were revolved around the yellow dotted line, as shown in Figure 124, before being embedded into square prisms. Using the resulting designs, the targets were printed in black ecoPLA on a Creality Ender 5 3D Printer. Following printing, the top surfaces of them were brush painted with a few coats of titanium white heavy body acrylic paint to yield the targets shown in Figure 123. The targets were labelled from A to E, as is indicated in this figure.

10.3.3.2 Calibration Target

In order to perform calibration, the mount for the ONH targets needed to be removed. Since it was magnetically attached this was very straightforward. The calibration target used in this work was that which was used in Chapter 5, as is detailed in subsubsection 5.2.3.4, which is shown in Figure 39. While it contained fewer squares in its pattern, it may be more precise than the one used in the works of Chapters 6, 7 and 8 since it was created by a professional photomask service rather than by photographing a monitor with a camera which may have some degree of distortion.

10.4 Performance Evaluation

10.4.1 Imaging

Prior to imaging the ONH targets, images of the calibration target were acquired. The calibration target was connected to a power supply and held on a flexible helping-hand. 50 images were then acquired of it, where it was held in different positions and tilts within the field of view of both cameras, making sure to maintain the focus on the target as best as practically possible.

Following the acquisition of the calibration images, images of the ONH targets were acquired. The system was set to project the speckle pattern shown in Figure 126. An image of each target was then acquired sequentially, and this was repeated 3 times such that in total 3 images were acquired of each target. It was found that the light intensity of the illumination system was so low that dark images were obtained with normal capture settings. In light of this, the images of the ONH targets were taken with an exposure time of 10 seconds.

Following this, the calibration procedure was carried out again to obtain a second set of 50 calibration images.

Then finally, the ONH targets were imaged again in the same manner as the first but with the targets in a slightly different position. Figure 127 is a CAD rendered view of the ONH target mount which shows how the ONH targets were mounted for both imaging rounds. The image on the right shows its position during this second imaging round, 2 mm higher than its position in the first. A 3D printed piece was used to increase the height.

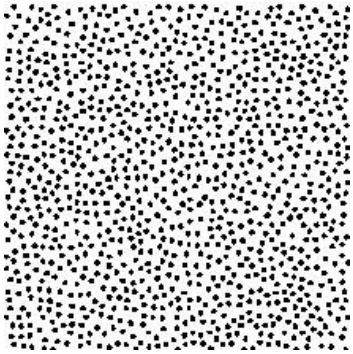


Figure 126: Pattern projected onto the ONH targets in this work.

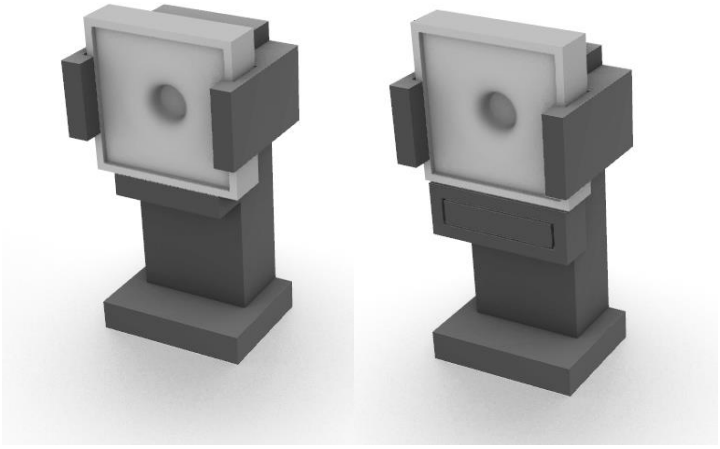


*Figure 127: Position of the ONH targets on the mount during the first (**LEFT**) and second (**RIGHT**) imaging rounds.*

10.4.2 Ground Truth Acquisition

The ground truth scans were acquired in the same way as in the work of Chapter 7 (detailed in subsubsection 7.4.4.1): the top surfaces were marked using pencil lead, imaged from many different viewpoints, reconstructed using 3DF Zephyr Free, and scaled to the correct scale. Given the time-consuming nature of this, ground truth data was acquired for targets A, C and E only.

10.4.3 Data Analysis

10.4.3.1 Calibration

Calibration was carried out, in the same way as is detailed in subsubsection 7.4.3.1, on each set of 50 images. In each case, calibration was performed and then the images which corresponded to a reprojection error greater than 1.8 pixels were deactivated. The results from both calibrations were then compared and the one which had the lowest overall mean reprojection error was used in the reconstruction process.

The square size of the calibration pattern was measured in the same way as is detailed in subsubsection 6.4.3.2, but using 2 measurements only. The square size was determined to be 2.17 mm.

10.4.3.2 Stereo Vision Pipeline

The stereo vision pipeline used was the same as that which is detailed in subsection 6.4.3: featuring the coarse-to-fine stereo matching algorithm and B-Spline interpolation. In this work, in order for the stereo matching algorithm to be able to return results, some changes were made to the thresholds. The consistency check threshold was set to 1.5 pixels, the minimum number of patch size levels to 2 and the texture threshold removed completely.

10.4.3.3 Quantitative Comparison

Precision where target position is unchanged: the range of the distance values between reconstructions of the same target, held in the first position, was determined through cloud-to-cloud comparison and output as a map. A map showing the variation for the targets was produced. This was done to give an indication on the precision where the ONH was imaged at roughly the same location at different times. This was assessed using targets A, C and E only, due to the time-consuming nature of this and also that performing this with targets B and D as well is unlikely to offer much additional insight.

Precision where target position was changing: Maps were produced in a similar manner to above however for each target, the comparison was made between one reconstruction at position 1 versus at position 2. This was assessed using targets A and E only, for the same reasons as mentioned above.

Ability to detect glaucomatous change: In order to get some indication on whether or not the structural changes which are characteristic of glaucomatous damage can be followed using this technique, cross-sectional slices of each of the reconstructions, for all targets, at position 1, were superimposed such that they could be subjectively compared.

Accuracy assessment: Accuracy was assessed using targets A, C and E only, for the same reasons as mentioned above. Cloud-to-cloud comparison was made, for each of the targets assessed, between the 3 reconstructions obtained of individual targets and the ground truth data. Accuracy maps were then obtained using the mean of the error maps obtained for each target.

10.5 Results and Discussion

10.5.1 Calibration

The calibration parameters obtained using each of the 50 calibration image sets are tabulated in Table 17. The table also shows the percentage difference between the values obtained for each of the parameters using each image set. It can be seen that the focal length values were very stable, with very similar values being obtained. Many of the other parameters differed by significant amounts.

The parameters which appeared to be less stable appear to, in most cases, have had a large uncertainty associated with their estimation. Table 18 shows the values estimated using the first set of images, along with their associated uncertainties (expressed as three SDs and as a percentage of the parameter values). The parameters such as focal lengths and principal point coordinates can be seen to have low uncertainties associated with them, which is in line with the fact that they were more stable. As could be expected, the uncertainties associated with the distortion coefficients are much larger in general.

It was seen that, on looking at how the parameters varied while an incremental number of the images were used in calibration, using the first image set, that some of the parameters

Table 17: Comparison of Calibration Parameters Obtained using Different Image Sets.

	Parameter	Calibration Image Set One	Calibration Image Set Two	% Difference
Left	f_x (pixels)	6356	6352	-0.063
	f_y (pixels)	6445	6482	0.561
	C_x (pixels)	1671	1514	-10.4
	C_y (pixels)	1196	1356	11.8
	k_1	-1.198	-1.115	-7.38
	k_2	-0.757	-2.246	66.3
	k_3	14.886	38.369	61.2
	P_1	0.048	0.026	-86.1
	P_2	0.028	0.053	47.8
Right	f_x (pixels)	6396	6402	0.089
	f_y (pixels)	6494	6529	0.530
	C_x (pixels)	1059	845	-25.3
	C_y (pixels)	1497	1625	7.87
	k_1	-1.167	-1.197	2.50
	k_2	-5.202	0.271	2020
	k_3	0.022	0.006	-3220
	P_1	0.023	0.049	-296
	P_2	86.121	2.592	53.7
	R_1	0.030	0.026	-16.4
	R_2	0.053	0.060	11.9
	R_3	0.026	0.025	-1.04
	T_x (mm)	4.052	4.110	1.43
	T_y (mm)	-0.139	-0.147	5.31
	T_z (mm)	0.322	0.504	35.9

Table 18: Values and Uncertainties for Calibration Parameters Estimated using Calibration Image Set 1. SD: Standard Deviation.

	Parameter	Value	Uncertainty (3SD)	% Uncertainty
Left	f_x (pixels)	6356	53.938275	0.85
	f_y (pixels)	6445	53.482703	0.83
	C_x (pixels)	1671	34.228842	2.05
	C_y (pixels)	1196	44.592433	3.73
	k_1	-1.198	0.065944	-5.51
	k_2	-0.757	1.837982	-242.88
	k_3	14.886	0.006915	14.52
	P_1	0.048	0.005176	18.59
	P_2	0.028	16.655205	111.89
Right	f_x (pixels)	6396	52.868329	0.83
	f_y (pixels)	6494	53.155452	0.82
	C_x (pixels)	1059	37.831105	3.57
	C_y (pixels)	1497	33.441174	2.23
	k_1	-1.167	0.079915	-6.85
	k_2	-5.202	3.028899	-58.23
	k_3	0.022	4.87E-03	21.86
	P_1	0.023	0.005744	25.28
	P_2	86.121	35.422529	41.13
	R_1	0.030	0.029863	0.74
	R_2	0.053	0.024091	-17.30
	R_3	0.026	0.246088	76.24
	T_x (mm)	4.052	0.00641	21.44
	T_y (mm)	-0.139	0.00486	9.17
	T_z (mm)	0.322	2.28E-04	0.89

appeared to stabilise. More images may not have led to very different results for these but may have for others, in particular the distortion parameters. The graphs showing how the parameters evolved with more and more images are provided in appendix 10A.

The distortion graphs, which are shown in Figure 128, appear more similar to that which would be expected for a typical camera than what was seen in the majority of the work with the slit lamp system. In this case the principal points are much closer to the image centres and the distortion profiles do not look to be particularly abnormal. This could have been as a result of a simpler and better aligned imaging system, and/or due to the calibration pattern being of a higher precision. Despite this, based on the uncertainties associated with the distortion parameters, there is likely still room for improvement.

The overall mean reprojection errors associated with the calibrations using the first and second set of images were 0.64 and 0.69 pixels, respectively. In light of this, the parameters obtained using the first set of images were used for reconstruction of the ONH targets. The potential sources of calibration error in this case are unchanged from those detailed previously in subsection 6.5.1 (corner localisation errors, inaccurately fabricated checkerboard, inaccurate checkerboard square size estimation, sub-optimal camera alignment and inadequacy of distortion models).

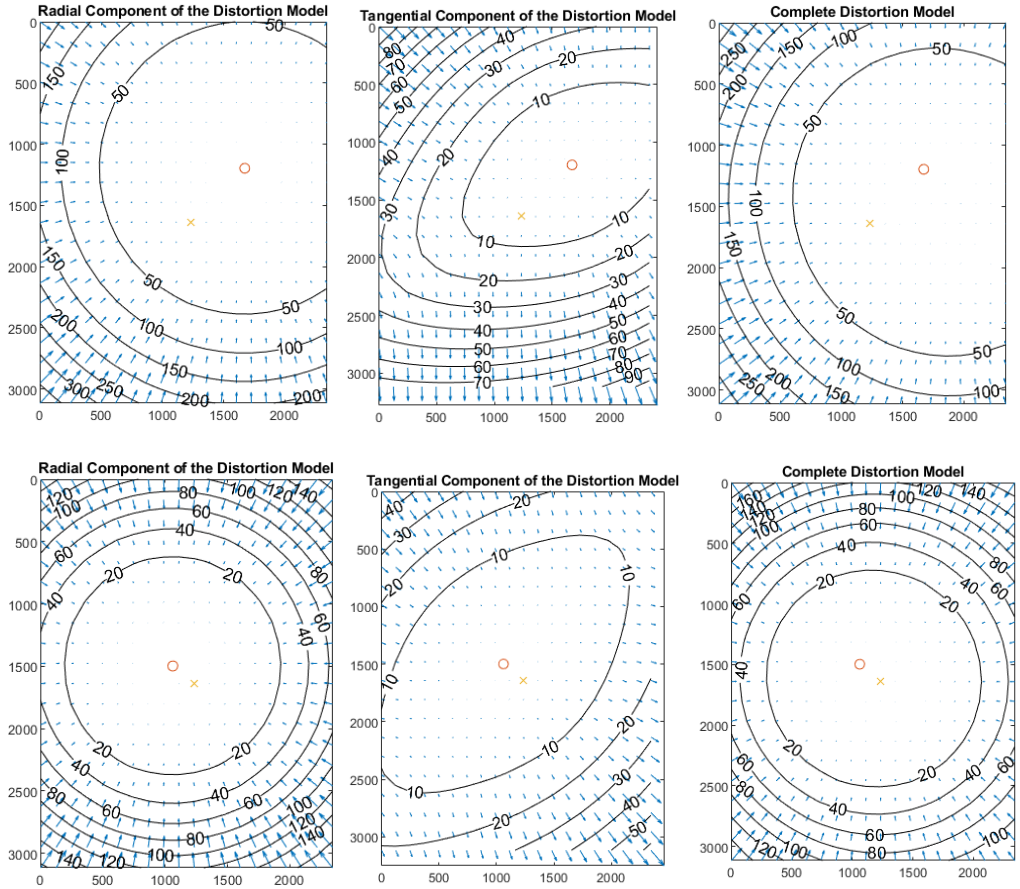


*Figure 128: Distortion maps obtained for the left (**TOP**) and right (**BOTTOM**) cameras of the handheld imaging system.*

10.5.2 RECONSTRUCTION

10.5.2.1 Precision of Reconstruction where Target Position was Unchanged

The reconstructions, shown as depth maps and textured reconstructions, obtained for target A when it was imaged in a fixed position at the back of the eye phantom, are shown in Figure 129. This figure also shows the stereo images from which the reconstructions were obtained, and a map showing the deviation between all of the reconstructions of this target. The same are also shown for targets C and E, where they were also held in a fixed position, in Figure 130 and Figure 131, respectively. It can be seen in the depth and deviation maps in each of these cases that the reconstructions did not deviate significantly. This result indicates that if the position of the target is fixed, reconstruction using this system and technique is precise. The deviation maps appear to show a random pattern. It is not entirely clear what is the cause of these local errors, but it is likely texture related and therefore potentially due to the projection pattern and stereo matching algorithm.

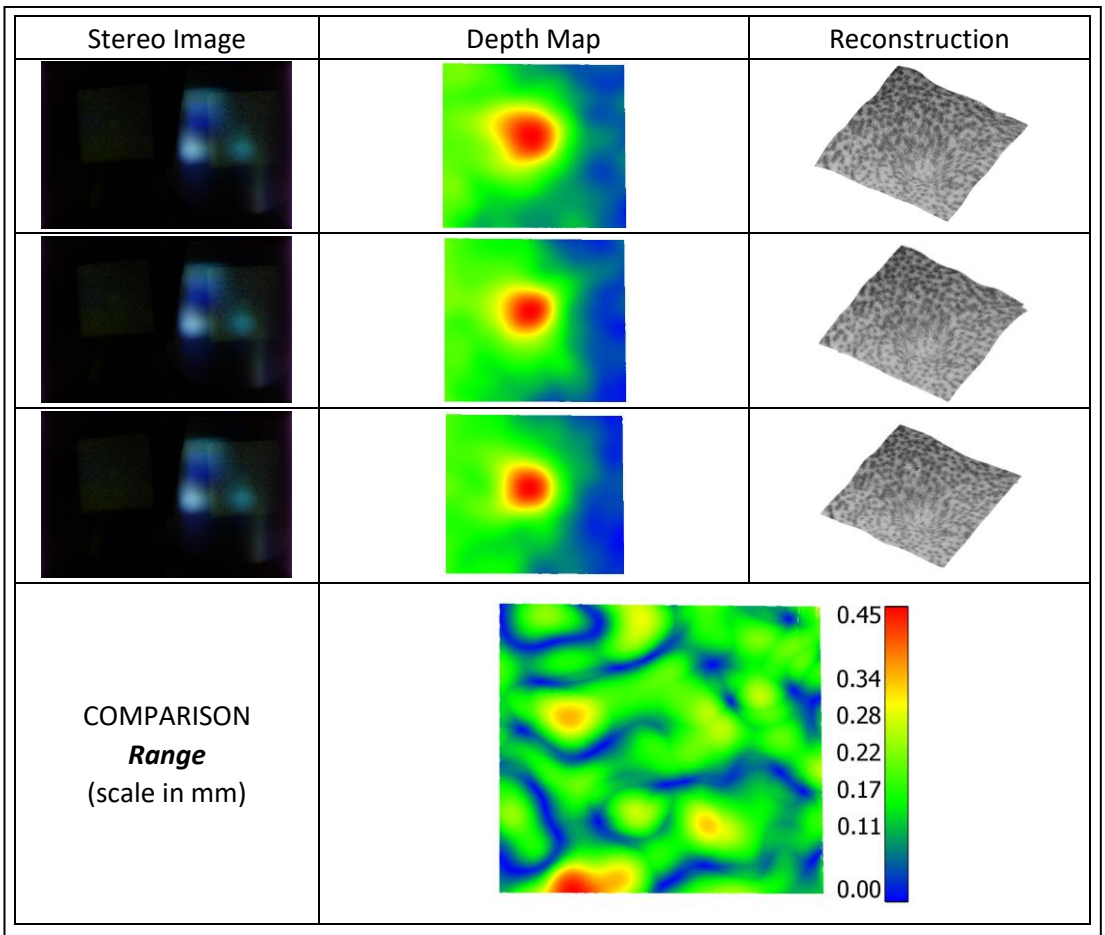


*Figure 129: Reconstructions (shown as depth maps and textured reconstruction) of **target A** held in practically the same position. The captured stereo image corresponding to each reconstruction is shown to its left. The comparison depth map shown shows the maximum deviation between the reconstructions.*

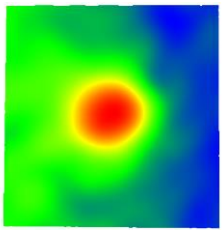
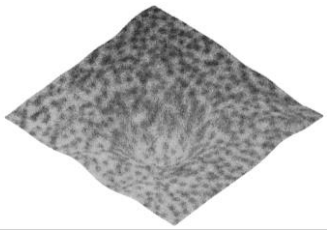
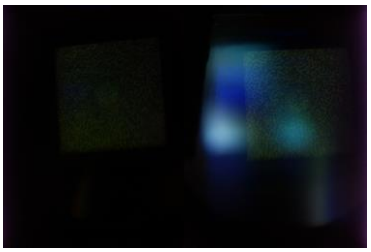
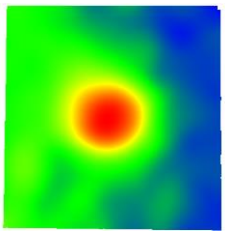
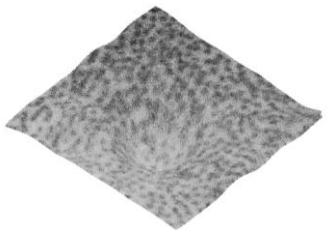
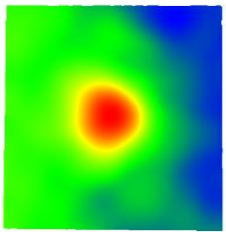
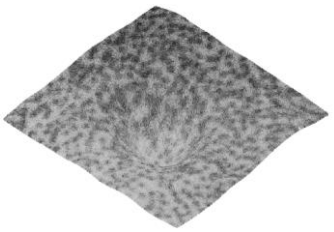
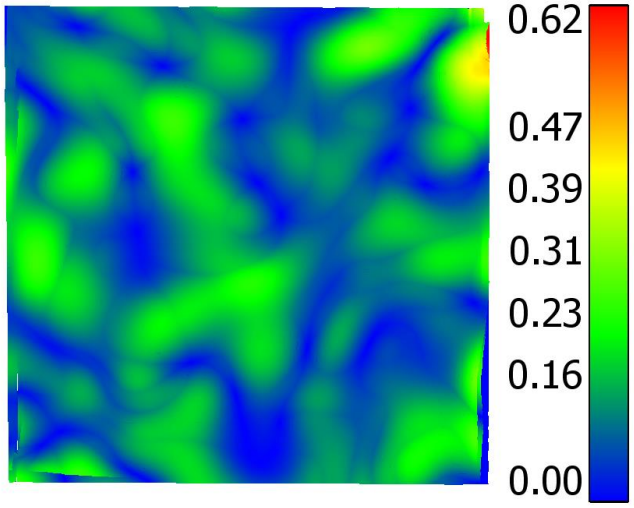
Stereo Image	Depth Map	Reconstruction
		
		
		
COMPARISON ***Range*** (scale in mm)		

*Figure 130: Reconstructions (shown as depth maps and textured reconstruction) of **target C** held in practically the same position. The captured stereo image corresponding to each reconstruction is shown to its left. The comparison depth map shown shows the maximum deviation between the reconstructions.*

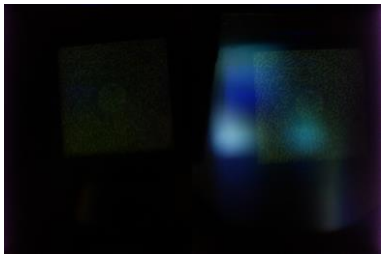
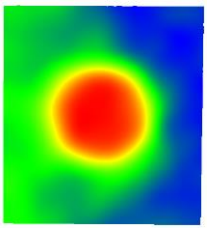
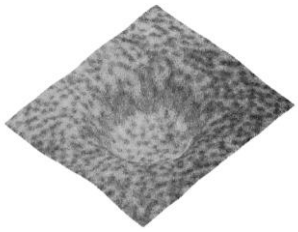
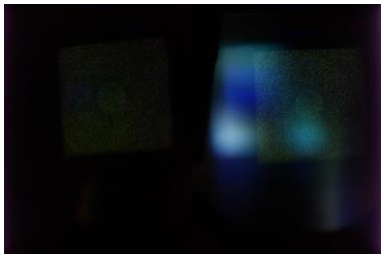
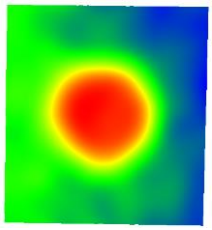
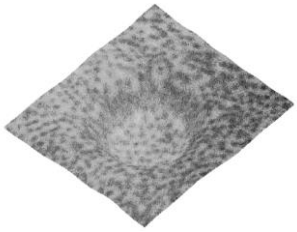
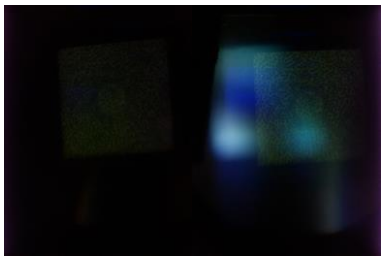
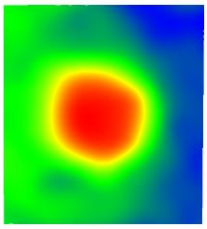
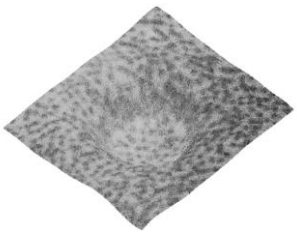
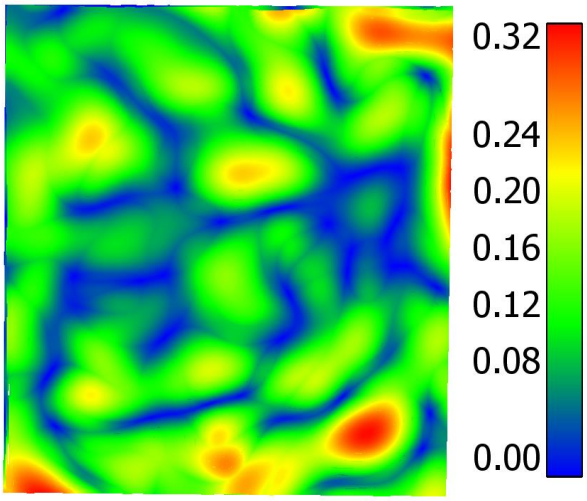
Stereo Image	Depth Map	Reconstruction
		
		
		
COMPARISON ***Range*** (scale in mm)		

*Figure 131: Reconstructions (shown as depth maps and textured reconstruction) of **target E** held in practically the same position. The captured stereo image corresponding to each reconstruction is shown to its left. The comparison depth map shown shows the maximum deviation between the reconstructions.*

10.5.2.2 Precision of Reconstruction where Target Position was Changed

The reconstruction results, shown in the same format as above, for target A, where it was imaged in two different positions, are shown in Figure 132. The same for target E is shown in Figure 133. It can be seen, in the depth maps and deviation map, that changing position did not have a significant effect on the reconstruction results obtained. This indicates that the precision of the system and algorithms is not noticeably affected by this type and amount of positional change of the target. However, it does not allow for the same to be said for changes in different directions, tilts and more extreme displacements. Again, the deviation maps show random patterns.

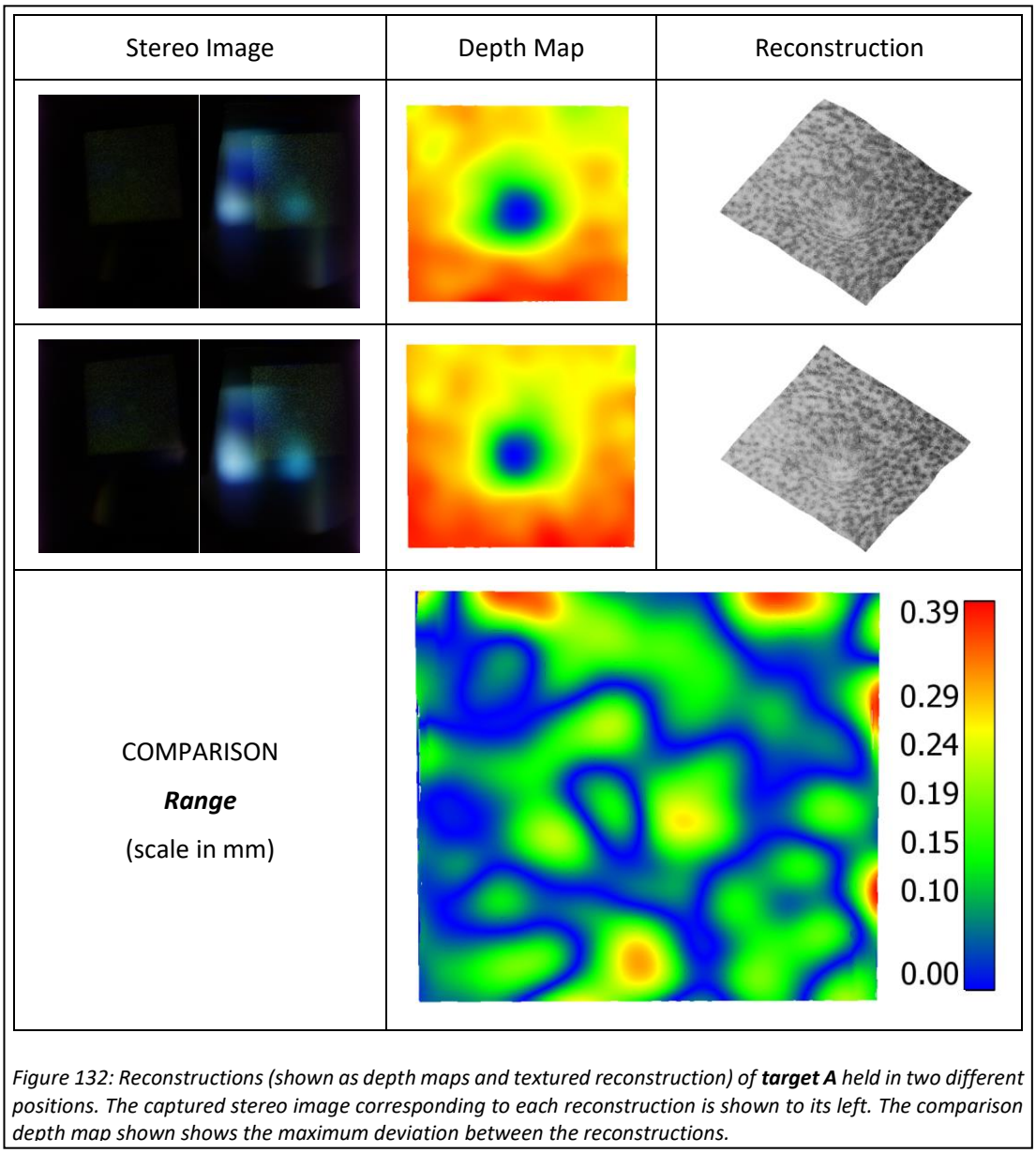


*Figure 132: Reconstructions (shown as depth maps and textured reconstruction) of **target A** held in two different positions. The captured stereo image corresponding to each reconstruction is shown to its left. The comparison depth map shown shows the maximum deviation between the reconstructions.*

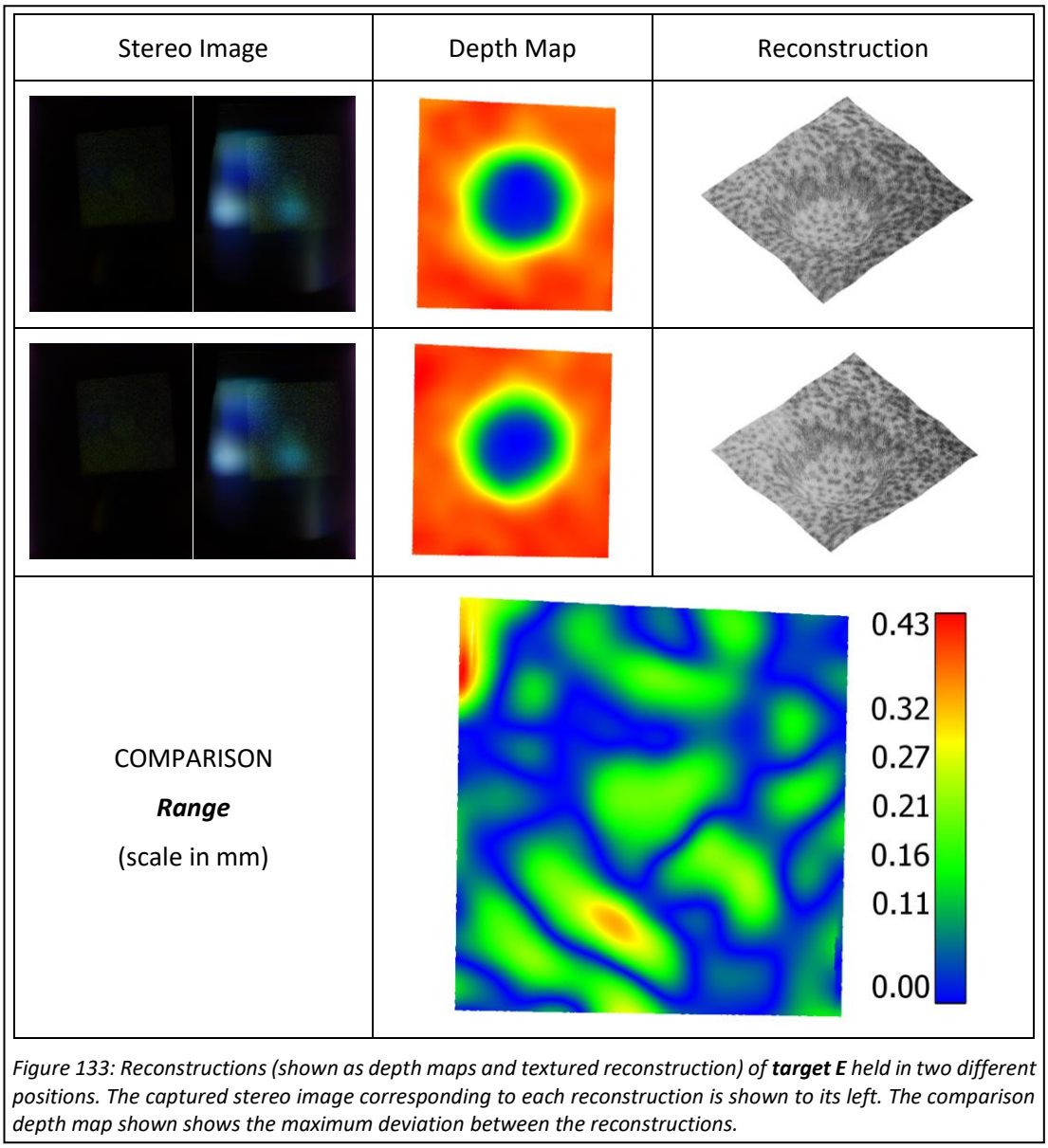


*Figure 133: Reconstructions (shown as depth maps and textured reconstruction) of **target E** held in two different positions. The captured stereo image corresponding to each reconstruction is shown to its left. The comparison depth map shown shows the maximum deviation between the reconstructions.*

10.5.2.3 Comparison of Reconstructions between Targets

Cross-sectional slices taken from the reconstructions obtained of each target in position 1, are shown in Figure 134. The cross-sectional slices of the reconstructions of each individual target also show that reconstruction appears to be precise. Moreover, it appears that, qualitatively, it is possible to discern between each of the targets. However, this is not the case for between targets D and E, where the shape change is the least dramatic. This result indicates that it may be possible to identify the kind of glaucomatous changes which would occur before an individual shows any detectable functional loss, provided the first scan occurred early in the course of POAG.

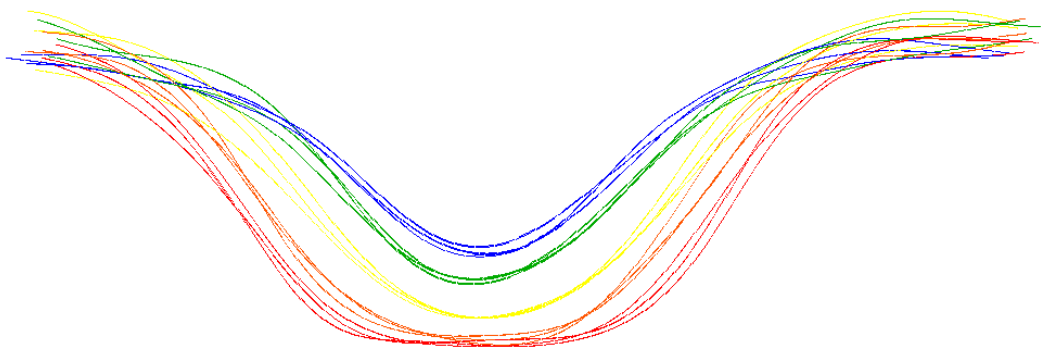


Figure 134: Superimposed cross-sectional slices of reconstructions of all of the targets, A to E, held in practically the same position. ***Blue*** *(A) – normal ONH,* ***Green*** *(B) – 10% damage,* ***Yellow*** *(C) – 20%,* ***Orange*** *(D) – 30%,* ***Red*** *(E) – 40%.*

10.5.2.4 Accuracy of Reconstruction

A map showing the mean error of the reconstructions obtained of target A in the first position, compared to ground truth data, is shown in Figure 135. Also shown in this figure are the input stereo images captured of the target for each reconstruction, the obtained textured reconstructions and depth maps. The same for targets C and E are provided in Figure 136 and Figure 137, respectively.

In all cases, the reconstructions appear to resemble the ground truth data relatively well. The depth of the dimple like structures appears to be well captured: the error maps show low error at the bottom of the dimples and on the top surface of the targets. One of the most noticeable errors is at the slopes/walls of the target, which when they get progressively steeper, as is the case when going from target A to E, they are less and less well recovered – indicated by increased error in this region. Another noticeable error is that the reconstructions appear less smooth than the ground truth scans. This is particularly noticeable on the top surfaces of the targets which, in reality, are practically flat.

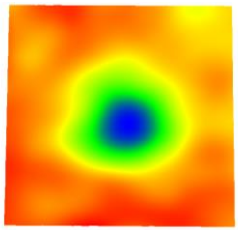
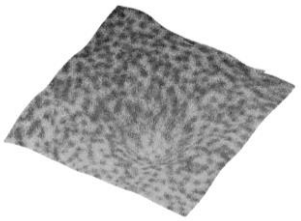
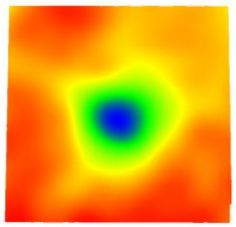
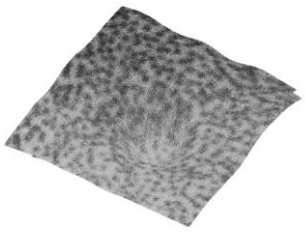
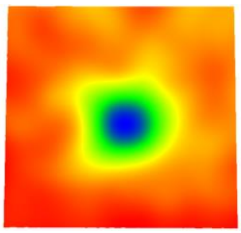
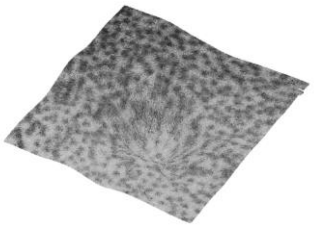
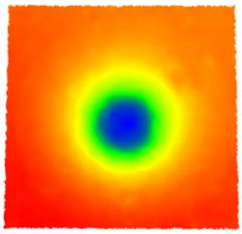
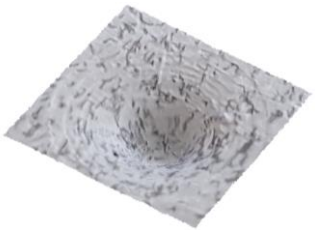
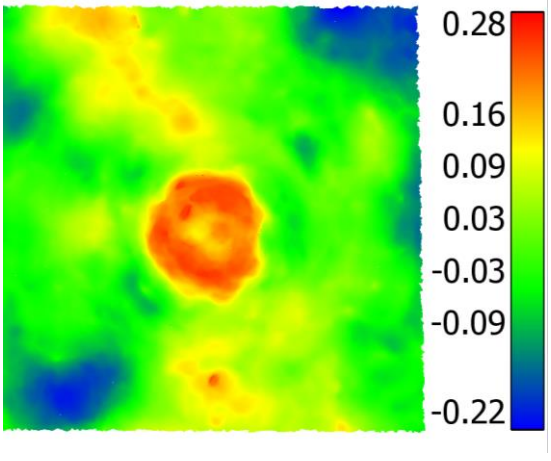
Stereo Image	Depth Map	Reconstruction
		
		
		
GROUND TRUTH		
COMPARISON ***Average Error*** (scale in mm)		

*Figure 135: Comparison of reconstructions obtained of **target A** in position 1 against its ground truth scan. Captured stereo images, depth maps and textured reconstructions are also shown.*

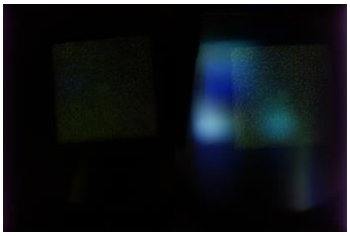
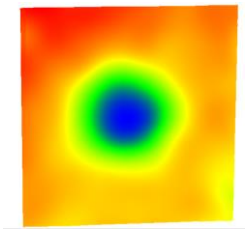
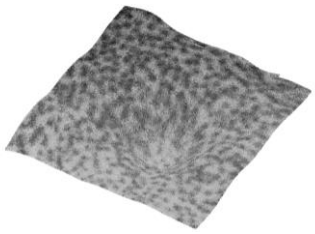
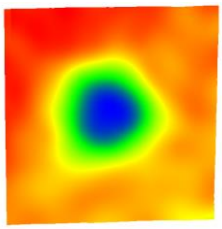
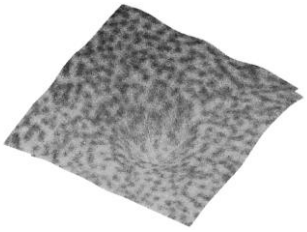
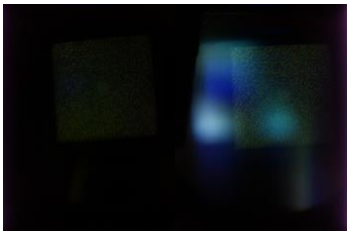
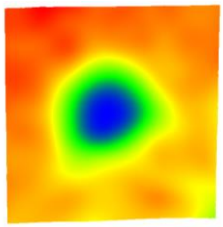
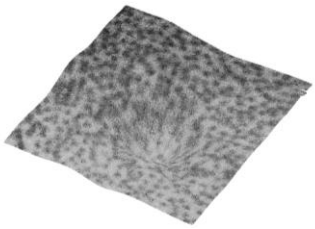
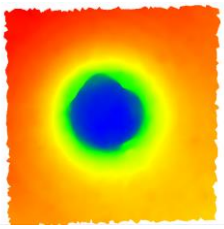
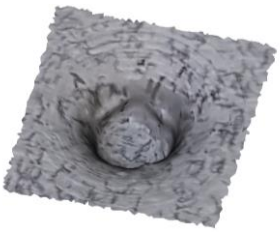
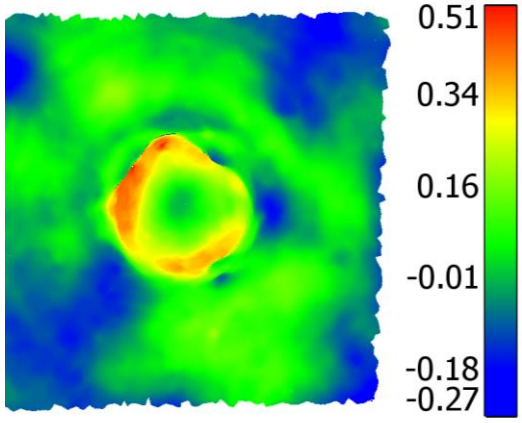
Stereo Image	Depth Map	Reconstruction
		
		
		
GROUND TRUTH		
COMPARISON ***Average Error*** (scale in mm)		

*Figure 136: Comparison of reconstructions obtained of **target C** in position 1 against its ground truth scan. Captured stereo images, depth maps and textured reconstructions are also shown.*

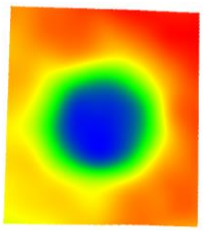
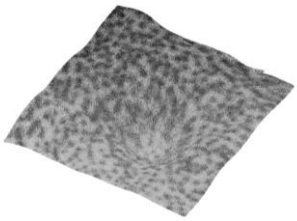
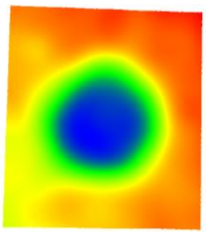
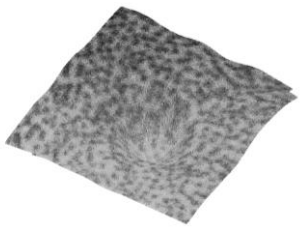
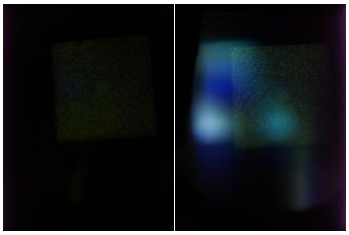
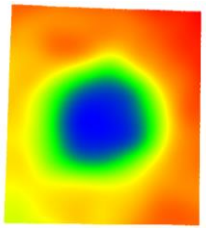
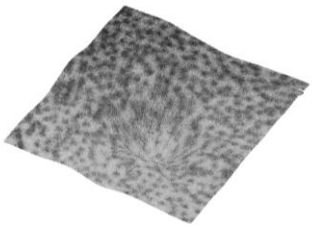
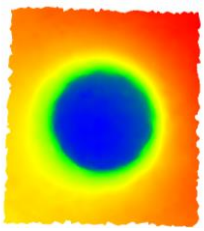
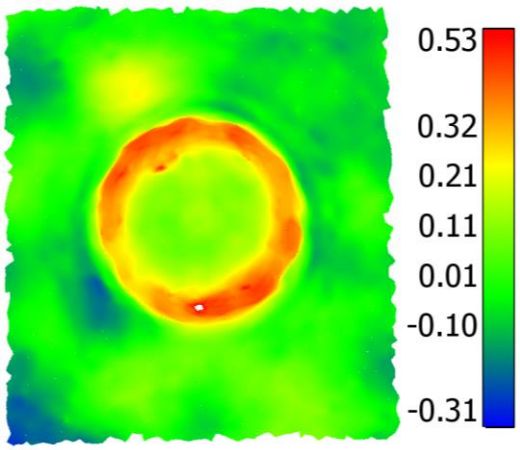
Stereo Image	Depth Map	Reconstruction
		
		
		
GROUND TRUTH		
COMPARISON ***Average Error*** (scale in mm)		

*Figure 137: Comparison of reconstructions obtained of **target E** in position 1 against its ground truth scan. Captured stereo images, depth maps and textured reconstructions are also shown.*

10.5.2.5 Theoretically Obtainable Depth Resolution

In order to gauge how this system compares to the other key technologies in this area, a calculation of the theoretically obtainable depth resolution can be informative. In the same manner as was detailed in subsubsection 8.5.2.2, a calculation of this was made. The calculation (provided in Appendix 10B) takes into account the scale factor of the eye phantom and has been corrected to what it would be for a life-sized eye. It was determined that the depth resolution (per matching accuracy in pixels) is in the range of 37.4 µm pixel-1 to 33.6 µm pixel-1 over the range of the ONH structure. Again, as mentioned in subsubsection 8.5.2.2, the depth resolution depends heavily on the matching accuracy possible. Subpixel accurate matching would enable for the system to exceed an axial resolution of between 37.4 µm to 33.6 µm. If matching accuracy was on the level of 1 pixel, then theoretically a depth resolution better than the slit lamp-based instrument (73.8 µm pixel-1 to 63.6 µm pixel-1), HRT (62.5 µm; Balasubramanian et al., 2011) but not OCT (20 µm to 5 µm; Aumann et al., 2019) could be possible.

10.5.3 Main Sources of Reconstruction Error

Projection, image quality and stereo matching: It is likely that the most major source of error, and area with the most room for improvement, relates to projection, image quality and stereo matching. The image quality and fineness of the projection pattern are important aspects which dictate the lower limit of the patch size that can be reliably matched. The focus of the left camera was not as good as the right camera. A stereo image pair obtained in this work is provided in Figure 138 to show this. This limited the scale/fineness of the pattern used. It is not entirely clear what caused this however it could have been partly due to damage to the camera's lenses. Although, it was noticed that the mirrors had some damage to them from the sanding process. Another aspect which likely affected stereo matching performance are the reflections that were seen in the right image. Moreover, there was a definite brightness discrepancy between the left and right images. Further work on improving the focus of the cameras, achieving equal brightness, and mitigating reflections would likely boost reconstruction performance.

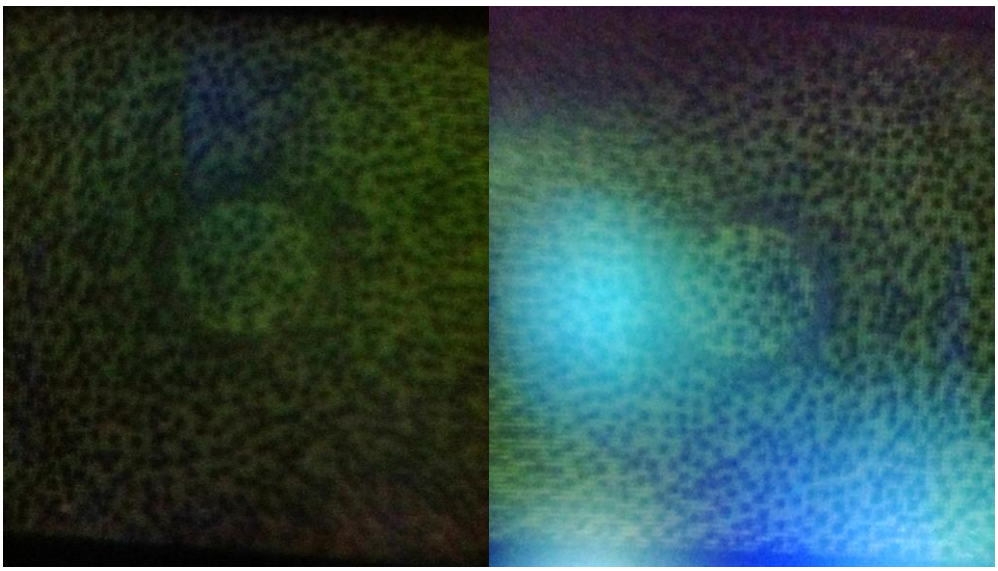


Figure 138: A stereo image pair obtained in this work. Both the left and right images have been cropped to shown only the target.

Calibration: Accurate calibration is vital for accurate stereo reconstruction. The calibration performance in this work, while it does appear to be much better than it was for the work with the slit lamp system, could still likely be improved. In particular, the distortion coefficients obtained showed the greatest uncertainties in general. Given that distortion has a significant impact on reconstruction performance, this is an important aspect to explore and improve in future work.

10.5.4 Limitations of This Work

One of the core limitations of this work is the lack of similarity of the eye phantom and targets to the human eye and its ONH. As is discussed in detail in subsubsection 5.6.4.3, the optical properties of the ONH targets differ from that of the human ONH. Moreover, as discussed in subsubsection 5.6.4.6, the optical characteristics of the eye phantom used differ from those of the human eye. A further related limitation, also mentioned in subsection 6.5.3, is that the ONH targets used had shapes which were rotationally symmetrical. Moreover, the way in which the targets were designed to mimic progressive POAG-like structural damage may not be valid. In order to achieve a more realistic shape, future work may benefit from designing the targets based on patient OCT data.

A further limitation of this work relates to the ground truth accuracy. It is not clear how accurate the photogrammetry-based method used in this work was. Future work would

benefit from using the optical measurement system that was used in the work detailed in Chapter 8 (subsubsection 8.4.4.1).

Another limitation of this work is the fact that, when capturing images of the ONH targets, the exposure time of the cameras had to be set to 10 seconds. Future work will need to work on increasing the illumination intensity of the projection system prior to any in vivo work such that capture times can be reduced significantly. Although not a problem for this work, the synchronicity of the cameras will need to be investigated prior to use of this system in vivo, as was discussed in subsection 6.5.3.

10.6 CONCLUSION

The core goal of this work was to design and test an optical configuration for a handheld stereo fundus camera and associated algorithms, in its ability to perform reconstruction of the ONH. Such initial testing involved performing reconstruction in an eye phantom. Another aspect to this testing was to see if the system could detect the kind of ONH structural changes which are characteristic of POAG.

This work showed that with a low-cost setup for a handheld unit, it was possible to reconstruct ONH targets in an eye phantom precisely and relatively accurately. Moreover, it was indicated, albeit in a simple and heavily controlled situation, that the kind of changes that occur to the ONH during POAG are detectable using this system. It was however seen that the greater the slope of the surface that is being reconstructed, the worse that system appears to perform. It is still not clear if the precision and accuracy are of a level needed for clinical usefulness, but the work indicates that future work on this unit should continue. The issues of this system/work, as well as their status, are provided in Table 19.

Table 19: Issues identified for the handheld unit.

Issue No.	Issue	Status
1	Image quality	Image quality was not the same between the left and right cameras, which is likely to have had significant effects on stereo matching performance.
2	Calibration performance sub-optimal	The calibration performance seen in this work appeared to be better than in all of the slit lamp works but could still be improved.
3	ONH target optical properties different to human ONH	The targets used in this work were opaque – different to the human ONH.
4	ONH target shape different to human ONH	The ONH targets in this work were rotationally symmetric and the POAG-like structural changes were created mathematically and may not be accurate to the actual shape of the human ONH and the structural changes which take place in the progression of POAG.
5	Eye phantom optics different to human eye	The optics of the eye phantom used in this work are hugely simplified compared to those of the human eye.
6	Image capture is not synchronous	The synchronicity of the cameras used was not tested in this work.
7	Stereo matching performance sub-optimal	There are many settings for the stereo matching algorithm developed which were not optimised.
8	Unclear ground truth accuracy and precision	The photogrammetry-based method used for ground truth acquisition is not ideal due to its precision and accuracy being unknown. Future work would benefit from using a system with known scanning performance.
9	Illumination intensity low	The illumination intensity was so low that the exposure time of the cameras needed to be set to 10 seconds.

CHAPTER 11

CONCLUSIONS AND RECOMMENDATIONS

11.1 CONCLUSIONS

Quantitative 3D imaging of the ONH, if it was available to more clinics, has the potential to reduce the number of missed cases of POAG. Unfortunately, it is prohibitively expensive for many clinics, particularly for those in the worst affected countries – those of low- and middle-income. Some work has been carried out by others (Carvalho & Carvalho, 2014) on low-cost stereo imaging-based reconstruction of the ONH, using computer stereo vision techniques. The overall goal of the work of this thesis was to develop instruments that operate by applying computer stereo vision techniques to stereo images obtained using them, which could be even lower in cost than the low-cost stereo fundus camera detailed by those researchers, and initially test their feasibility of performing 3D surface reconstruction of the ONH. The first aim of this work was to develop a slit lamp-based instrument, consisting of low-cost off-the-shelf components fitted to a slit lamp, and initially test its feasibility. The second aim, developed later in the work, was to see if a simple handheld unit, composed of a small number of low-cost components, together with the appropriate stereo vision algorithms, could be developed and used for the same task. Such a system would be more appropriate for remote and rural areas where existing ophthalmic equipment, including slit lamps, is unavailable.

At the beginning of this work, an initial simple version of the slit lamp-based instrument was developed, along with a simple eye phantom containing highly textured retinal targets, and uncalibrated reconstruction was demonstrated through the use of standard stereo vision algorithms. In doing so, it showed that no fundamental flaw in the idea existed and allowed for the first objective of this work to be met. After this initial demonstration, improvements were made to the instrument, and custom stereo vision algorithms created, which allowed for calibrated reconstructions of highly textured ONH targets in the simple eye phantom to be obtained. This allowed for the second objective of this work to be met. At this stage, the performance trial conducted was more involved than many of those seen in the literature. Firstly, calibration performance was analysed in depth, which is not explored in any of the works within the literature. Secondly, the work involved an assessment of precision, and accuracy against ground truth data – a level of assessment not seen in any of the works within the literature. This level of performance trial was also a feature of the later trials of this work, apart from in the work with porcine eyes since a ground truth could not be obtained at that stage. Once calibrated reconstruction was demonstrated with highly textured ONH targets,

the next objective was to obtain calibrated reconstruction of poorly textured ONH targets, which matches the case of the human ONH more closely. In order to allow for the stereo vision algorithms to operate, to make reconstruction possible, a projection system was added to project texture onto the surfaces of the ONH targets – a feature not seen in the systems used in other works within the literature. In successfully achieving reconstruction of these targets, the third objective of this work was met. Proceeding forward from this, the next objective sought to demonstrate reconstruction of ONH targets with little texture within an eye phantom which was more similar in terms of optical characteristics to the human eye. In order to meet this, a life-sized fluid-filled eye was developed, and reconstructions were demonstrated, allowing for the fourth objective of this work to be met. The final objective relating to the slit lamp-based instrument was to demonstrate reconstruction within a biological eye. In this case porcine eyes were used, and reconstruction of a porcine ONH was successfully demonstrated, although due to an inability to obtain ground truth data at this stage, the result could not be quantitatively validated. Despite this, the fifth objective of this work was achieved. In meeting these objectives, the aim of developing the instrument and initially testing its feasibility was met. On top of this, the results indicated promising levels of precision and accuracy, but it is not clear from this work if the instrument is at a level needed in order to be clinically useful. Further work is therefore warranted, and eventual clinical testing.

Regarding the handheld unit, the only objective that was set involved developing the instrument and performing calibrated reconstruction of textureless ONH targets in a simple eye phantom. In this work, an initial setup for the instrument, which also featured a projection system, was created. Using the system, calibrated reconstruction on textureless ONH targets was demonstrated, meeting the objective and aim. This work also showed that the instrument was capable of promising levels of precision and accuracy. Moreover, the experiment was extended slightly to see if the instrument may be capable of detecting the kind and degree of structural changes which occur to the ONH during the progression of POAG prior to detectable vison loss. Using a series of ONH targets with progressively changing structures, it was shown, in a simplified and controlled situation, that the instrument may be capable of this. Therefore, further work of this instrument is also warranted, and eventual clinical testing.

The core contributions of this work were as follows:

1. The creation and initial performance assessment of two low-cost systems, together with corresponding stereo vision algorithms, for 3D surface reconstruction of the ONH: a slit lamp-based and standalone handheld system.

2. The addition of projection systems to such systems to artificially increase texture density on the ONH's surface to boost stereo matching performance.

3. The creation of a life-sized fluid-filled eye phantom which can be created at low-cost; is close, in terms of optical characteristics, to the human eye; and can be used for calibration of retinal imaging systems and ONH surface imaging studies.

4. The scoping reviews conducted to map the research which has been conducted on stereo vision based ONH surface reconstruction, and on fluid-filled optical eye phantoms.

11.2 Issues and Recommendations for Future Work

Going forward with this project, it is clear that the systems require further work to their hardware and software to improve their performance, and the level of performance evaluation needs to be advanced, eventually to the point where the systems are tested in vivo. The issues with the systems, limiting their performance, which were identified in this work, are listed in Table 16 and Table 19. Many suggestions to boost performance, to address such issues, have been detailed in the discussions of the relevant chapters. Given the importance of calibration on reconstruction accuracy, and that a relatively poor level of calibration performance was seen in this work, it would likely be very beneficial to address this in future work. Suggestions to improve calibration performance were given in subsection 6.5.1.

Aside from the performance level of the systems, it could be beneficial in future work to improve the test subjects used (e.g., the eye phantoms). There were issues identified with the life-sized fluid-filled eye phantom, in terms of optical quality that needs to be addressed to begin with. Apart from this, one change which would allow for a much more realistic test scenario would be the use of semi-transparent ONH targets. In having a closer match to the biological ONH, it would enable faster development of the projection and imaging systems such that it would perform better on the biological ONH. Another recommendation for future work is to investigate the light levels and wavelengths illuminating the ONH structures by the projection systems to allow for this to be optimised while also ensuring that they remain

within safe limits according to ISO 15004-2:2007 (International Organization for Standardization, 2007).

Once sufficient performance was obtained with such ONH targets, further porcine eye work would be permitted. If performance was adequate, then it would make sense to perform in vivo work on human eyes. Successful trials in vivo could provide an impetus to realise these systems commercially, which could broaden access to 3D ONH imaging in low- and middle-income settings, where they are needed most.

REFERENCES

Abe, R. Y., Gracitelli, C. P. B., Diniz-Filho, A., Tatham, A. J., & Medeiros, F. A. (2015). Lamina Cribrosa in Glaucoma: Diagnosis and Monitoring. *Current Ophthalmology Reports*, *3*(2), 74–84. https://doi.org/10.1007/s40135-015-0067-7

Adelson, J. D., Bourne, R. R. A., Briant, P. S., Flaxman, S. R., Taylor, H. R. B., Jonas, J. B., Abdoli, A. A., Abrha, W. A., Abualhasan, A., Abu-Gharbieh, E. G., Adal, T. G., Afshin, A., Ahmadieh, H., Alemayehu, W., Alemzadeh, S. A. S., Alfaar, A. S., Alipour, V., Androudi, S., Arabloo, J., … Vos, T. (2021). Causes of blindness and vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to VISION 2020: The Right to Sight: an analysis for the Global Burden of Disease Study. *The Lancet Global Health*, *9*(2), e144–e160. https://doi.org/10.1016/S2214-109X(20)30489-7

Adhi, M., & Duker, J. S. (2013). Optical coherence tomography – current and future applications. *Current Opinion in Ophthalmology*, *24*(3), 213–221. https://doi.org/10.1097/ICU.0b013e32835f8bf8

Agrawal, A., Baxi, J., Calhoun, W., Chen, C.-L., Ishikawa, H., Schuman, J. S., Wollstein, G., & Hammer, D. X. (2016). Optic Nerve Head Measurements With Optical Coherence Tomography: A Phantom-Based Study Reveals Differences Among Clinical Devices. *Investigative Ophthalmology & Visual Science*, *57*(9), OCT413–OCT420. https://doi.org/10.1167/iovs.15-18738

Agrawal, A., Connors, M., Beylin, A., Liang, C.-P., Barton, D., Chen, Y., Drezek, R. A., & Pfefer, T. J. (2012). Characterizing the point spread function of retinal OCT devices with a model eye-based phantom. *Biomedical Optics Express*, *3*(5), 1116–1126. https://doi.org/10.1364/BOE.3.0011163

Albon, J., Purslow, P. P., Karwatowski, W. S. S., & Easty, D. L. (2000). Age related compliance of the lamina cribrosa in human eyes. *British Journal of Ophthalmology*, *84*(3), 318–323. https://doi.org/10.1136/bjo.84.3.318

Alencar, L. M., & Medeiros, F. A. (2011). The role of standard automated perimetry and newer functional methods for glaucoma diagnosis and follow-up. *Indian Journal of Ophthalmology*, *59*(Suppl1), S53–S58. https://doi.org/10.4103/0301-4738.73694

Ara, M., Ferreras, A., Pajarin, A. B., Calvo, P., Figus, M., & Frezzotti, P. (2015, June 22). *Repeatability and Reproducibility of Retinal Nerve Fiber Layer Parameters Measured by Scanning Laser Polarimetry with Enhanced Corneal Compensation in Normal and Glaucomatous Eyes* [Clinical Study]. BioMed Research International; Hindawi. https://doi.org/10.1155/2015/729392

Arianpour, A., Tremblay, E. J., Stamenov, I., Ford, J. E., Schanzlin, D. J., & Lo, Y. (2013). An optomechanical model eye for ophthalmological refractive studies. *Journal of Refractive Surgery*, *29*(2), 126–132. https://doi.org/10.3928/1081597X-20130117-08

Arora, S., Chung, H., & Damji, K. (2015). *Optic Disk Size Assessment Techniques: Photo Essay*. https://doi.org/10.23937/2378-346X/1410009

Artal, P. (2016). The Eye as an Optical Instrument. In M. D. Al-Amri, M. El-Gomati, & M. S. Zubairy (Eds.), *Optics in Our Time* (pp. 285–297). Springer International Publishing. https://doi.org/10.1007/978-3-319-31903-2_12

Atchison, D. A., & Smith, G. (2000). *Optics of the Human Eye*. Elsevier. https://doi.org/10.1016/B978-0-7506-3775-6.X5001-9

Atchison, D. A., & Thibos, L. N. (2016). Optical models of the human eye. *Clinical and Experimental Optometry*, *99*(2), 99–106. https://doi.org/10.1111/cxo.12352

Aumann, S., Donner, S., Fischer, J., & Müller, F. (2019). Optical Coherence Tomography (OCT): Principle and Technical Realization. In J. F. Bille (Ed.), *High Resolution Imaging in Microscopy and Ophthalmology: New Frontiers in Biomedical Optics* (pp. 59–85). Springer International Publishing. https://doi.org/10.1007/978-3-030-16638-0_3

Bader, J., & Havens, S. J. (2020). Tonometry. In *StatPearls*. StatPearls Publishing. http://www.ncbi.nlm.nih.gov/books/NBK493225/

Bahrami, M., Heidari, A., & Pierscionek, B. K. (2016). Alteration in refractive index profile during accommodation based on mechanical modelling. *Biomedical Optics Express*, *7*(1), 99–110. https://doi.org/10.1364/BOE.7.000099

Bakaraju, R. C., Ehrmann, K., Falk, D., Ho, A., & Papas, E. (2010). Physical human model eye and methods of its use to analyse optical performance of soft contact lenses. *Optics Express*, *18*(16), 16868–16882. https://doi.org/10.1364/OE.18.016868

Balasubramanian, M., Bowd, C., Weinreb, R. N., & Zangwill, L. M. (2011). Agreement between Heidelberg Retina Tomograph-I and -II in detecting glaucomatous changes using topographic change analysis. *Eye*, *25*(1), 31–42. https://doi.org/10.1038/eye.2010.124

Ballew, B. S. (2009). Elsevier's Scopus® Database. *Journal of Electronic Resources in Medical Libraries*, *6*(3), 245–252. https://doi.org/10.1080/15424060903167252

Bansal, M., Sizintsev, M., Eledath, J., Sawhney, H., Pearson, D. J., & Stone, R. A. (2013). 3D optic disc reconstruction via a global fundus stereo algorithm. *Conference Proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, *2013*, 5877–5882. https://doi.org/10.1109/EMBC.2013.6610889

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded Up Robust Features. In A. Leonardis, H. Bischof, & A. Pinz (Eds.), *Computer Vision – ECCV 2006* (pp. 404–417). Springer. https://doi.org/10.1007/11744023_32

Berestov, A. L. (2000). *Stereo fundus photography: Automatic evaluation of retinal topography*. *3957*, 50–59. https://doi.org/10.1117/12.384479

Bhowmik, D., & Appiah, K. (2018). Embedded Vision Systems: A Review of the Literature. *ARC*. https://doi.org/10.1007/978-3-319-78890-6_17

Birkner, S., Einighammer, J., Oltrup, T., Bende, T., & Jean, B. (2011). Biometric measurements inside the model eye using a two wavelengths Fourier domain low coherence interferometer. *Biomedizinische Technik. Biomedical Engineering*, *56*(1), 65–71. https://doi.org/10.1515/BMT.2010.059

Bouguet, J.-Y. (2015). *Camera Calibration Toolbox for Matlab*. http://www.vision.caltech.edu/bouguetj/calib_doc/index.html

Bradski, G., & Kaehler, A. (2000). *OpenCV*. *3*.

Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc.

Brand, U., Mengedoht, K., Pessel, M., Paul, M., & Trier, H. G. (1994). Measurement of vibration induced by diagnostical ultrasound in an eye model with simulated retina. *Microscopy, Holography, and Interferometry in Biomedicine*, *2083*, 347–355. https://doi.org/10.1117/12.167409

Callan, T., Durbin, M., Chhagan, S., Yu, S., & Abunto, T. (2012). Repeatability and Reproducibility of Optic Nerve Head Measurements on the Cirrus HD-OCT. *Investigative Ophthalmology & Visual Science*, *53*(14), 687–687.

Cao, Z., Ding, Z., Hu, Z., Qiao, W., Liu, W., & Chen, X. (2018). Model eyes with curved multilayer structure for the axial resolution evaluation of an ophthalmic optical

coherence tomography device. *Journal of Innovative Optical Health Sciences*, *11*(03), 1850013. https://doi.org/10.1142/S179354581850013X

Carvalho, L. A. V. de, & Carvalho, V. (2014). Cost-effective instrumentation for quantitative depth measurement of optic nerve head using stereo fundus image pair and image cross correlation techniques. *Ophthalmic Technologies XXIV*, *8930*, 89301C. https://doi.org/10.1117/12.2035727

Chan, H. H. L., Ong, D. N., Kong, Y. X. G., O'Neill, E. C., Pandav, S. S., Coote, M. A., & Crowston, J. G. (2014). Glaucomatous optic neuropathy evaluation (GONE) project: The effect of monoscopic versus stereoscopic viewing conditions on optic nerve evaluation. *American Journal of Ophthalmology*, *157*(5), 936–944. https://doi.org/10.1016/j.ajo.2014.01.024

Chen, P. P. (2004). Risk and risk factors for blindness from glaucoma. *Current Opinion in Ophthalmology*, *15*(2), 107–111. https://doi.org/10.1097/00055735-200404000-00009

Christopher, M., Belghith, A., Bowd, C., Proudfoot, J. A., Goldbaum, M. H., Weinreb, R. N., Girkin, C. A., Liebmann, J. M., & Zangwill, L. M. (2018). Performance of Deep Learning Architectures and Transfer Learning for Detecting Glaucomatous Optic Neuropathy in Fundus Photographs. *Scientific Reports*, *8*(1), 16685. https://doi.org/10.1038/s41598-018-35044-9

Clarivate Analytics. (1997). *Web of Science*. https://www.webofknowledge.com/

CloudCompare. (2020). http://www.cloudcompare.org/

Corcoran, A., Muyo, G., van Hemert, J., Gorman, A., & Harvey, A. R. (2015). Application of a wide-field phantom eye for optical coherence tomography and reflectance imaging. *Journal of Modern Optics*, *62*(21), 1828–1838. https://doi.org/10.1080/09500340.2015.1045309

Corona, E., Mitra, S., Wilson, M., Krile, T., Kwon, Y. H., & Soliz, P. (2002). Digital stereo image analyzer for generating automated 3-D measures of optic disc deformation in glaucoma. *IEEE Transactions on Medical Imaging*, *21*(10), 1244–1253. https://doi.org/10.1109/TMI.2002.806293

Corona, E., Mitra, S., Wilson, M., & Soliz, P. (2002). Digital stereo-optic disc image analyzer for monitoring progression of glaucoma. *Medical Imaging 2002: Image Processing*, *4684*, 82–93. https://doi.org/10.1117/12.467043

Crock, G., & Parel, J. M. (1969). Stereophotogrammetry of fluorescein angiographs in ocular biometrics. *The Medical Journal of Australia*, *2*(12), 586–590.

Dabasia, P. L., Edgar, D. F., Garway-Heath, D. F., & Lawrenson, J. G. (2014). A survey of current and anticipated use of standard and specialist equipment by UK optometrists. *Ophthalmic & Physiological Optics: The Journal of the British College of Ophthalmic Opticians (Optometrists)*, *34*(5), 592–613. https://doi.org/10.1111/opo.12150

Damji, K. F., Nazarali, S., Giorgis, A., Kiage, D., Marco, S., Philippin, H., Daniel, N., & Amin, S. (2017). STOP Glaucoma in Sub Saharan Africa: Enhancing awareness, detection, management, and capacity for glaucoma care. *Expert Review of Ophthalmology*, *12*(3), 197–206. https://doi.org/10.1080/17469899.2017.1295848

Delgado, M. F., Abdelrahman, A. M., Terahi, M., Miro Quesada Woll, J. J., Gil-Carrasco, F., Cook, C., Benharbit, M., Boisseau, S., Chung, E., Hadjiat, Y., & Gomes, J. A. (2019). Management Of Glaucoma In Developing Countries: Challenges And Opportunities For Improvement. *ClinicoEconomics and Outcomes Research: CEOR*, *11*, 591–604. https://doi.org/10.2147/CEOR.S218277

Devalla, S. K., Chin, K. S., Mari, J.-M., Tun, T. A., Nicholas, G., Strouthidis, Aung, T., Thiery, A. H., & Girard, M. J. A. (2017). *Glaucoma A Deep Learning Approach to Digitally Stain Optical Coherence Tomography Images of the Optic Nerve Head*. /paper/Glaucoma-

A-Deep-Learning-Approach-to-Digitally-of-Devalla-Chin/50838b9cd72ae6d384eac001a95cb37c3686f88d

Dreher, A. W., Tso, P. C., & Weinreb, R. N. (1991). Reproducibility of topographic measurements of the normal and glaucomatous optic nerve head with the laser tomographic scanner. *American Journal of Ophthalmology*, *111*(2), 221–229. https://doi.org/10.1016/s0002-9394(14)72263-9

Elbassiouny, O., Awadallah, A. K., Sallam, M. A., & Soliman, F. E.-S. (2018). *Swept-source-optical coherence tomography study of choroidal thickness in maculopathy type two diabetes mellitus patients*. https://doi.org/10.15406/aovs.2018.08.00311

Escudero-Sanz, I., & Navarro, R. (1999). Off-axis aberrations of a wide-angle schematic eye model. *Journal of the Optical Society of America. A, Optics, Image Science, and Vision*, *16*(8), 1881–1891. https://doi.org/10.1364/josaa.16.001881

Falconer, D. G. (1973). Digital Stereophotogrammetry of the Ocular Fundus. *Applied Optics*, *12*(7), 1388–1389. https://doi.org/10.1364/AO.12.001388

Fenner, B. J., Wong, R. L. M., Lam, W.-C., Tan, G. S. W., & Cheung, G. C. M. (2018). Advances in Retinal Imaging and Applications in Diabetic Retinopathy Screening: A Review. *Ophthalmology and Therapy*, *7*(2), 333–346. https://doi.org/10.1007/s40123-018-0153-7

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395. https://doi.org/10.1145/358669.358692

Fujimoto, J. G., Pitris, C., Boppart, S. A., & Brezinski, M. E. (2000). Optical Coherence Tomography: An Emerging Technology for Biomedical Imaging and Optical Biopsy. *Neoplasia (New York, N.Y.)*, *2*(1–2), 9–25.

Garway-Heath, D. F., Ruben, S. T., Viswanathan, A., & Hitchings, R. A. (1998). Vertical cup/disc ratio in relation to optic disc size: Its value in the assessment of the glaucoma suspect. *British Journal of Ophthalmology*, *82*(10), 1118–1124. https://doi.org/10.1136/bjo.82.10.1118

Gatti, V., Hill, J., Mitra, S., & Nutter, B. (2014). Optic disc boundary segmentation from diffeomorphic demons registration of monocular fundus image sequences versus 3D visualization of stereo fundus image pairs for automated early stage glaucoma assessment. *Medical Imaging 2014: Biomedical Applications in Molecular, Structural, and Functional Imaging*, *9038*, 90380H. https://doi.org/10.1117/12.2044291

Gazzard, G., Morgan, W., Devereux, J., Foster, P., Oen, F., Seah, S., Khaw, P. T., & Chew, P. (2003). Optic disc hemorrhage in Asian glaucoma patients. *Journal of Glaucoma*, *12*(3), 226–231. https://doi.org/10.1097/00061198-200306000-00008

Geiger, A., Roser, M., & Urtasun, R. (2011). Efficient Large-Scale Stereo Matching. In R. Kimmel, R. Klette, & A. Sugimoto (Eds.), *Computer Vision – ACCV 2010* (pp. 25–38). Springer. https://doi.org/10.1007/978-3-642-19315-6_3

Giaconi, J. A., Law, S. K., Nouri-Mahdavi, K., Coleman, A. L., & Caprioli, J. (Eds.). (2016a). *Pearls of Glaucoma Management* (2nd ed.). Springer-Verlag. https://www.springer.com/gp/book/9783662490402

Giaconi, J. A., Law, S. K., Nouri-Mahdavi, K., Coleman, A. L., & Caprioli, J. (2016b). *Pearls of Glaucoma Management*. Springer.

Gramatikov, B. (2014). Modern technologies for retinal scanning and imaging: An introduction for the biomedical engineer. *Biomedical Engineering Online*, *13*, 52. https://doi.org/10.1186/1475-925X-13-52

Gray's Anatomy. Descriptive and Applied. (1913). Philadelphia, Lea & Febiger.

Guo, F., Zhao, X., Zou, B., & Ouyang, P. (2018). 3D Reconstruction and Registration for Retinal Image Pairs. *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, 364–368. https://doi.org/10.1109/ICIVC.2018.8492769

Haag-Streit Group. (n.d.). *Haag-Streit Avanti Glaucoma Module Information Flyer*. Retrieved 31 January 2021, from https://www.haag-streit.com/haag-streit-uk/products/optovue/oct/avanti/

Hale, G. M., & Querry, M. R. (1973). Optical Constants of Water in the 200-nm to 200-µm Wavelength Region. *Applied Optics*, *12*(3), 555–563. https://doi.org/10.1364/AO.12.000555

Hartley, R., & Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Harwerth, R. S., Carter-Dawson, L., Shen, F., Smith, E. L., & Crawford, M. L. (1999). Ganglion cell losses underlying visual field defects from experimental glaucoma. *Investigative Ophthalmology & Visual Science*, *40*(10), 2242–2250.

Heidelberg Engineering. (n.d.). *Patient Flyer. Was ist ein Glaukom?* Retrieved 11 August 2020, from https://media.heidelbergengineering.com/downloads/practice-marketing/practice-tools/93939-001_PatientFlyer-OCT-Glaucoma_DE_RGB.pdf

Heidelberg Engineering. (1999). *Quantitative Threedimensional Imaging of the Posterior Segment with the Heidelberg Retina Tomograph.* http://www.oftech.com.ar/Lecturas/heidelberg.pdf

Hirschmuller, H. (2005). Accurate and efficient stereo processing by semi-global matching and mutual information. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, *2*, 807–814 vol. 2. https://doi.org/10.1109/CVPR.2005.56

Hu, Z., Liu, W., Hong, B., Hao, B., Wang, L., & Li, J. (2014). A physical model eye with 3D resolution test targets for optical coherence tomography. *Twelfth International Conference on Photonics and Imaging in Biology and Medicine (PIBM 2014)*, *9230*, 92301H. https://doi.org/10.1117/12.2067974

International Organization for Standardization. (2007). *ISO 15004-2:2007*. https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/03/89/38952.html

Jayabalan, G. S., Kessler, R., Fischer, J., & Bille, J. F. (2019). Compact Adaptive Optics Scanning Laser Ophthalmoscope with Phase Plates. In J. F. Bille (Ed.), *High Resolution Imaging in Microscopy and Ophthalmology: New Frontiers in Biomedical Optics* (pp. 377–394). Springer International Publishing. https://doi.org/10.1007/978-3-030-16638-0_18

Jiang, X., Torres, M., & Varma, R. (2018). Variation in Intraocular Pressure and the Risk of Developing Open-angle Glaucoma: The Los Angeles Latino Eye Study. *American Journal of Ophthalmology*, *188*, 51–59. https://doi.org/10.1016/j.ajo.2018.01.013

Jonas, J. B., Aung, T., Bourne, R. R., Bron, A. M., Ritch, R., & Panda-Jonas, S. (2017). Glaucoma. *The Lancet*, *390*(10108), 2183–2193. https://doi.org/10.1016/S0140-6736(17)31469-1

Kai, Z., Yuzhou, W., & Guoping, W. (2006). Hierarchical Stereo Matching: From Foreground to Background. In J. Blanc-Talon, W. Philips, D. Popescu, & P. Scheunders (Eds.), *Advanced Concepts for Intelligent Vision Systems* (pp. 632–643). Springer. https://doi.org/10.1007/11864349_58

Kamal, M. A., Alhomida, A. S., Al-Rajhi, A. A., & Al-Jafari, A. A. (2000). Thermodynamic analysis of human retinal acetylcholinesterase inhibition using an anti-Alzheimer's drug, tacrine, through the development of a dual substrate and temperature model.

Proceedings of the National Science Council, Republic of China. Part B, Life Sciences, *24*(3), 108–115.

Kern, C., Fu, D. J., Kortuem, K., Huemer, J., Barker, D., Davis, A., Balaskas, K., Keane, P. A., McKinnon, T., & Sim, D. A. (2020). Implementation of a cloud-based referral platform in ophthalmology: Making telemedicine services a reality in eye care. *British Journal of Ophthalmology*, *104*(3), 312–317. https://doi.org/10.1136/bjophthalmol-2019-314161

Khaw, P. T., Shah, P., & Elkington, A. R. (2004). Glaucoma—1: Diagnosis. *BMJ : British Medical Journal*, *328*(7431), 97–99.

Killer, H. E., & Pircher, A. (2020). What is the optimal glaucoma treatment: Reducing aqueous humour production or facilitating its outflow? *Eye*, 1–3. https://doi.org/10.1038/s41433-020-0862-8

Kingman, S. (2004). Glaucoma is second leading cause of blindness globally. *Bulletin of the World Health Organization*, *82*(11), 887–888.

Kong, H. J., Kim, S. K., Seo, J. M., Park, K. H., Chung, H., Park, K. S., & Kim, H. C. (2004). Three dimensional reconstruction of conventional stereo optic disc image. *Conference Proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, *2*, 1229–1232. https://doi.org/10.1109/IEMBS.2004.1403391

Kwartz, A. J., Henson, D. B., Harper, R. A., Spencer, A. F., & McLeod, D. (2005). The effectiveness of the Heidelberg Retina Tomograph and laser diagnostic glaucoma scanning system (GDx) in detecting and monitoring glaucoma. *Health Technology Assessment (Winchester, England)*, *9*(46), 1–132, iii. https://doi.org/10.3310/hta9460

Lee, J.-S. (2019). *Primary Eye Examination: A Comprehensive Guide to Diagnosis*. Springer.

Leung, C. K. S. (2015). Detecting optic nerve head deformation and retinal nerve fiber layer thinning in glaucoma progression. *Taiwan Journal of Ophthalmology*, *5*(2), 50–55. https://doi.org/10.1016/j.tjo.2015.04.003

Lin, S. C., Singh, K., Jampel, H. D., Hodapp, E. A., Smith, S. D., Francis, B. A., Dueker, D. K., Fechtner, R. D., Samples, J. S., Schuman, J. S., & Minckler, D. S. (2007). Optic Nerve Head and Retinal Nerve Fiber Layer Analysis. *Ophthalmology*, *114*(10), 1937–1949. https://doi.org/10.1016/j.ophtha.2007.07.005

Lorenz, B., & Borruat, F.-X. (2007). *Pediatric Ophthalmology, Neuro-Ophthalmology, Genetics*. Springer Science & Business Media.

Lusk, J. A., & Nutter, B. (2009). Automated 3-D reconstruction of stereo fundus images via camera calibration and image rectification. *2009 22nd IEEE International Symposium on Computer-Based Medical Systems*, 1–7. https://doi.org/10.1109/CBMS.2009.5255339

Marriner, H. (n.d.). https://www.pngitem.com/middle/iTxRbTm_human-eye-png-eye-ball-cross-section-transparent/

Maslin, J. S., Mansouri, K., & Dorairaj, S. K. (2015). HRT for the Diagnosis and Detection of Glaucoma Progression. *The Open Ophthalmology Journal*, *9*, 58–67. https://doi.org/10.2174/1874364101509010058

Matlach, J., Bender, S., König, J., Binder, H., Pfeiffer, N., & Hoffmann, E. M. (2018). Investigation of intraocular pressure fluctuation as a risk factor of glaucoma progression. *Clinical Ophthalmology (Auckland, N.Z.)*, *13*, 9–16. https://doi.org/10.2147/OPTH.S186526

McKelvie, J., Ku, J. Y., McArdle, B., & McGhee, C. (2009). Wavefront aberrometry: Comparing and profiling higher-order aberrations produced by intraocular lenses in vitro using

a physical model eye system and Hartman-Shack aberrometry. *Journal of Cataract and Refractive Surgery*, *35*(3), 547–555. https://doi.org/10.1016/j.jcrs.2008.11.048

Menolotto, M., Giardini, M. E., Jordan, K. C., & Coghill, I. (2019). *Comprehensive Ophthalmic Research Database—CORD*. https://doi.org/10.15129/39bcd12d-0677-4cf3-a099-b763fbb7d3c4

Middleton, S. (2010). Porcine Ophthalmology. *Veterinary Clinics of North America: Food Animal Practice*, *26*(3), 557–572. https://doi.org/10.1016/j.cvfa.2010.09.002

Motta, D. A., Serillo, A., Matos, L. de, Yasuoka, F. M. M., Bagnato, V. S., & Carvalho, L. A. V. (2014). 3D papillary image capturing by the stereo fundus camera system for clinical diagnosis on retina and optic nerve. *Design and Quality for Biomedical Technologies VII*, *8936*, 893614. https://doi.org/10.1117/12.2038435

Motta, D., de Matos, L., de Souza, A. C., Marcato, R., Paiva, A., & de Carvalho, L. A. V. (2015). All-in-focus imaging technique used to improve 3D retinal fundus image reconstruction. *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, 26–31. https://doi.org/10.1145/2695664.2695845

Muramatsu, C., Hatanaka, Y., Ishida, K., Sawada, A., Yamamoto, T., & Fujita, H. (2014). Preliminary study on differentiation between glaucomatous and non-glaucomatous eyes on stereo fundus images using cup gradient models. *Medical Imaging 2014: Computer-Aided Diagnosis*, *9035*, 903533. https://doi.org/10.1117/12.2043409

Muramatsu, C., Nakagawa, T., Sawada, A., Hatanaka, Y., Hara, T., Yamamoto, T., & Fujita, H. (2009). Determination of cup-to-disc ratio of optical nerve head for diagnosis of glaucoma on stereo retinal fundus image pairs. *Medical Imaging 2009: Computer-Aided Diagnosis*, *7260*, 72603L. https://doi.org/10.1117/12.811461

Muramatsu, C., Nakagawa, T., Sawada, A., Hatanaka, Y., Yamamoto, T., & Fujita, H. (2011). Automated determination of cup-to-disc ratio for classification of glaucomatous and

normal eyes on stereo retinal fundus images. *Journal of Biomedical Optics*, *16*(9), 096009. https://doi.org/10.1117/1.3622755

Murray, R. M., Li, Z., Sastry, S. S., & Sastry, S. S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press.

Nakagawa, T., Hayashi, Y., Hatanaka, Y., Aoyama, A., Hara, T., Fujita, A., Kakogawa, M., Fujita, H., & Yamamoto, T. (2007). Three-Dimensional Reconstruction of Optic Nerve Head from Stereo Fundus Images and Its Quantitative Estimation. *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 6747–6750. https://doi.org/10.1109/IEMBS.2007.4353910

Nakagawa, T., Hayashi, Y., Hatanaka, Y., M.d, A. A., Hara, T., Kakogawa, M., Fujita, H., & M.d, T. Y. (2007). Comparison of the depth of an optic nerve head obtained using stereo retinal images and HRT. *Medical Imaging 2007: Physiology, Function, and Structure from Medical Images*, *6511*, 65112M. https://doi.org/10.1117/12.706237

Nakagawa, T., Suzuki, T., Hayashi, Y., Mizukusa, Y., Hatanaka, Y., Ishida, K., Hara, T., Fujita, H., & Yamamoto, T. (2008). Quantitative depth analysis of optic nerve head using stereo retinal fundus image pair. *Journal of Biomedical Optics*, *13*(6), 064026. https://doi.org/10.1117/1.3041711

Noecker, R. J. (2006). The management of glaucoma and intraocular hypertension: Current approaches and recent advances. *Therapeutics and Clinical Risk Management*, *2*(2), 193–206.

Norouzifard, M., Dawda, A., Abdul-Rahman, A., GholamHosseini, H., & Klette, R. (2018). Superpixel Segmentation Methods on Stereo Fundus Images and Disparity Map for Glaucoma Detection. *2018 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, 1–6. https://doi.org/10.1109/IVCNZ.2018.8634732

Ohnuma, K., Shiokawa, Y., Hirayama, N., & Hua, Q. (2000). Eye Model Using a CCD Camera for Observing the Images Constructed by IOLs. In C. Fotakis, T. G. Papazoglou, & C. Kalpouzos (Eds.), *Optics and Lasers in Biomedicine and Culture* (pp. 336–339). Springer. https://doi.org/10.1007/978-3-642-56965-4_66

Okutomi, M., Yoshizaki, O., & Tomita, G. (1992). Color stereo matching and its application to 3-D measurement of optic nerve head. *11th IAPR International Conference on Pattern Recognition [1992] Proceedings*, 509–513. https://doi.org/10.1109/ICPR.1992.201611

Optics Incorporated. (n.d.). *Ultra M2 Slit Lamp, Unit Module*. Optics Incorporated. Retrieved 11 August 2020, from https://opticsinc.com/products/ultra-m2-slit-lamp-unit-module

Otto, T. P., Burk, R. O. W., Bille, J. F., & Dengler, J. J. (1994). Comparison of laser tomographic fundus mapping to digital analysis of stereo photographs. *Laser Applications in Ophthalmology*, *2079*, 71–75. https://doi.org/10.1117/12.168723

Paquet, D. (2020). *The Retina*. https://thebrain.mcgill.ca/flash/i/i_02/i_02_cl/i_02_cl_vis/i_02_cl_vis.html

Parihar, J. K. S. (2016). Glaucoma: The 'Black hole' of irreversible blindness. *Medical Journal, Armed Forces India*, *72*(1), 3–4. https://doi.org/10.1016/j.mjafi.2015.12.001

Prata, T. S., Meira-Freitas, D., Lima, V. C., Guedes, L. M., Magalhães, F. P., & Paranhos Junior, A. (2010). Factors affecting the variability of the Heidelberg Retina Tomograph III measurements in newly diagnosed glaucoma patients. *Arquivos Brasileiros de Oftalmologia*, *73*(4), 354–357. https://doi.org/10.1590/S0004-27492010000400011

Quigley, H. A. (2011). Glaucoma. *Lancet (London, England)*, *377*(9774), 1367–1377. https://doi.org/10.1016/S0140-6736(10)61423-7

Quigley, H. A., Brown, A. E., Morrison, J. D., & Drance, S. M. (1990). The Size and Shape of the Optic Disc in Normal Human Eyes. *Archives of Ophthalmology*, *108*(1), 51–57. https://doi.org/10.1001/archopht.1990.01070030057028

Ramirez, J. M., Mitra, S., & Morales, J. (1999). Visualization of the three-dimensional topography of the optic nerve head through a passive stereo vision model. *J. Electronic Imaging*. https://doi.org/10.1117/1.482687

Ramirez, M., Mitra, S., Kher, A., & Morales, J. (1992). 3-D digital surface recovery of the optic nerve head from stereo fundus images. *[1992] Proceedings Fifth Annual IEEE Symposium on Computer-Based Medical Systems*, 284–291. https://doi.org/10.1109/CBMS.1992.244937

Ramírez-Hernández, L. R., Rodríguez-Quiñonez, J. C., Castro-Toscano, M. J., Hernández-Balbuena, D., Flores-Fuentes, W., Rascón-Carmona, R., Lindner, L., & Sergiyenko, O. (2020). Improve three-dimensional point localization accuracy in stereo vision systems using a novel camera calibration method. *International Journal of Advanced Robotic Systems*, *17*(1), 1729881419896717. https://doi.org/10.1177/1729881419896717

Reis, A. S. C., O'Leary, N., Yang, H., Sharpe, G. P., Nicolela, M. T., Burgoyne, C. F., & Chauhan, B. C. (2012). Influence of clinically invisible, but optical coherence tomography detected, optic disc margin anatomy on neuroretinal rim evaluation. *Investigative Ophthalmology & Visual Science*, *53*(4), 1852–1860. https://doi.org/10.1167/iovs.11-9309

Reus, N. J., de Graaf, M., & Lemij, H. G. (2007). Accuracy of GDx VCC, HRT I, and clinical assessment of stereoscopic optic nerve head photographs for diagnosing glaucoma. *The British Journal of Ophthalmology*, *91*(3), 313–318. https://doi.org/10.1136/bjo.2006.096586

Rohrschneider, K., Burk, R. O., Kruse, F. E., & Völcker, H. E. (1994). Reproducibility of the optic nerve head topography with a new laser tomographic scanning device. *Ophthalmology*, *101*(6), 1044–1049. https://doi.org/10.1016/s0161-6420(94)31220-6

Ruiz-Ederra, J., García, M., Hernández, M., Urcola, H., Hernández-Barbáchano, E., Araiz, J., & Vecino, E. (2005). The pig eye as a novel model of glaucoma. *Experimental Eye Research*, *81*(5), 561–569. https://doi.org/10.1016/j.exer.2005.03.014

Rusu, R. B., & Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). *2011 IEEE International Conference on Robotics and Automation*, 1–4. https://doi.org/10.1109/ICRA.2011.5980567

Sanchez, I., Martin, R., Ussa, F., & Fernandez-Bueno, I. (2011). The parameters of the porcine eyeball. *Graefe's Archive for Clinical and Experimental Ophthalmology = Albrecht Von Graefes Archiv Fur Klinische Und Experimentelle Ophthalmologie*, *249*(4), 475–482. https://doi.org/10.1007/s00417-011-1617-9

Schacknow, P. N., & Samples, J. R. (2010). *The Glaucoma Book: A Practical, Evidence-Based Approach to Patient Care*. Springer.

Schneider, C. A., Rasband, W. S., & Eliceiri, K. W. (2012). NIH Image to ImageJ: 25 years of image analysis. *Nature Methods*, *9*(7), 671–675. https://doi.org/10.1038/nmeth.2089

Scottish Intercollegiate Guidelines Network (SIGN). (2015). *Glaucoma referral and safe discharge* (No. 144). http://www.sign.ac.uk

Seymenoğlu, G., Başer, E., & Oztürk, B. (2013). Comparison of spectral-domain optical coherence tomography and Heidelberg retina tomograph III optic nerve head parameters in glaucoma. *Ophthalmologica. Journal International D'ophtalmologie.*

International Journal of Ophthalmology. Zeitschrift Fur Augenheilkunde, *229*(2), 101–105. https://doi.org/10.1159/000341574

Shaarawy, T. M., Sherwood, M. B., Hitchings, R. A., & Crowston, J. G. (2014). *Glaucoma E-Book*. Elsevier Health Sciences.

Sharma, A., Corona, E., Mitra, S., & Nutter, B. S. (2006). Early detection of glaucoma using fully automated disparity analysis of optic nerve head (ONH) from stereo fundus images. *Medical Imaging 2006: Image Processing*, *6144*, 61440H. https://doi.org/10.1117/12.654019

Shields, M. B., Tiedeman, J. S., Miller, K. N., Hickingbotham, D., & Ollie, A. R. (1989). Accuracy of topographic measurements with the Optic Nerve Head Analyzer. *American Journal of Ophthalmology*, *107*(3), 273–279. https://doi.org/10.1016/0002-9394(89)90312-7

Song, B. J., & Caprioli, J. (2014). New directions in the treatment of normal tension glaucoma. *Indian Journal of Ophthalmology*, *62*(5), 529–537. https://doi.org/10.4103/0301-4738.133481

Sultanova, N., Kasarova, S. N., & Nikolov, I. (2009). *Dispersion Properties of Optical Polymers*. https://doi.org/10.12693/aphyspola.116.585

Susanna, R., De Moraes, C. G., Cioffi, G. A., & Ritch, R. (2015). Why Do People (Still) Go Blind from Glaucoma? *Translational Vision Science & Technology*, *4*(2). https://doi.org/10.1167/tvst.4.2.1

Szeliski, R. (2011). *Computer Vision: Algorithms and Applications*. Springer-Verlag. https://doi.org/10.1007/978-1-84882-935-0

Taketani, F., & Hara, Y. (2011). Characteristics of spherical aberrations in 3 aspheric intraocular lens models measured in a model eye. *Journal of Cataract and Refractive Surgery*, *37*(5), 931–936. https://doi.org/10.1016/j.jcrs.2010.12.044

Tang, L., Garvin, M. K., Lee, K., Alward, W. L. M., Kwon, Y. H., & Abràmoff, M. D. (2011). Robust Multiscale Stereo Matching from Fundus Images with Radiometric Differences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(11), 2245–2258. https://doi.org/10.1109/TPAMI.2011.69

Tang, L., Kardon, R. H., Wang, J.-K., Garvin, M. K., Lee, K., & Abràmoff, M. D. (2012). Quantitative evaluation of papilledema from stereoscopic color fundus photographs. *Investigative Ophthalmology & Visual Science*, *53*(8), 4490–4497. https://doi.org/10.1167/iovs.12-9803

Tang, L., Scheetz, T. E., Mackey, D. A., Hewitt, A. W., Fingert, J. H., Kwon, Y. H., Quellec, G., Reinhardt, J. M., & Abràmoff, M. D. (2010). Automated quantification of inherited phenotypes from color images: A twin study of the variability of optic nerve head shape. *Investigative Ophthalmology & Visual Science*, *51*(11), 5870–5877. https://doi.org/10.1167/iovs.10-5527

Tatham, A. J., Weinreb, R. N., Zangwill, L. M., Liebmann, J. M., Girkin, C. A., & Medeiros, F. A. (2013). The Relationship Between Cup-to-Disc Ratio and Estimated Number of Retinal Ganglion Cells. *Investigative Ophthalmology & Visual Science*, *54*(5), 3205–3214. https://doi.org/10.1167/iovs.12-11467

Tham, Y.-C., Li, X., Wong, T. Y., Quigley, H. A., Aung, T., & Cheng, C.-Y. (2014). Global Prevalence of Glaucoma and Projections of Glaucoma Burden through 2040: A Systematic Review and Meta-Analysis. *Ophthalmology*, *121*(11), 2081–2090. https://doi.org/10.1016/j.ophtha.2014.05.013

Thomas, D., & Duguid, G. (2004). Optical coherence tomography—A review of the principles and contemporary uses in retinal investigation. *Eye*, *18*(6), 561–570. https://doi.org/10.1038/sj.eye.6700729

Tran, K., Mendel, T. A., Holbrook, K. L., & Yates, P. A. (2012). Construction of an Inexpensive, Hand-Held Fundus Camera through Modification of a Consumer "Point-and-Shoot" Camera. *Investigative Ophthalmology & Visual Science*, *53*(12), 7600–7607. https://doi.org/10.1167/iovs.12-10449

Tricco, A. C., Lillie, E., Zarin, W., O'Brien, K. K., Colquhoun, H., Levac, D., Moher, D., Peters, M. D. J., Horsley, T., Weeks, L., Hempel, S., Akl, E. A., Chang, C., McGowan, J., Stewart, L., Hartling, L., Aldcroft, A., Wilson, M. G., Garritty, C., … Straus, S. E. (2018). PRISMA Extension for Scoping Reviews (PRISMA-ScR): Checklist and Explanation. *Annals of Internal Medicine*, *169*(7), 467–473. https://doi.org/10.7326/M18-0850

U.S. National Library of Medicine. (1988). *PubMed*. https://pubmed.ncbi.nlm.nih.gov/

Vasudevan, R., Kurillo, G., Lobaton, E., Bernardin, T., Kreylos, O., Bajcsy, R., & Nahrstedt, K. (2011). High-Quality Visualization for Geographically Distributed 3-D Teleimmersive Applications. *IEEE Transactions on Multimedia*, *13*(3), 573–584. https://doi.org/10.1109/TMM.2011.2123871

Vessani, R. M., Moritz, R., Batis, L., Zagui, R. B., Bernardoni, S., & Susanna, R. (2009). Comparison of quantitative imaging devices and subjective optic nerve head assessment by general ophthalmologists to differentiate normal from glaucomatous eyes. *Journal of Glaucoma*, *18*(3), 253–261. https://doi.org/10.1097/IJG.0b013e31818153da

Wang, J., & Barton, K. (2017). Aqueous shunt implantation in glaucoma. *Taiwan Journal of Ophthalmology*, *7*(3), 130–137. https://doi.org/10.4103/tjo.tjo_35_17

Weinreb, R. N., Aung, T., & Medeiros, F. A. (2014). The Pathophysiology and Treatment of Glaucoma. *JAMA*, *311*(18), 1901–1911. https://doi.org/10.1001/jama.2014.3192

Wünstel, M., & Schumann, H. (2002). *Automatic 3D-reconstruction of the ocular fundus from stereo images*.

Xie, P., Hu, Z., Zhang, X., Li, X., Gao, Z., Yuan, D., & Liu, Q. (2014). Application of 3-Dimensional Printing Technology to Construct an Eye Model for Fundus Viewing Study. *PLOS ONE*, *9*(11), e109373. https://doi.org/10.1371/journal.pone.0109373

Xu, J., & Chutatape, O. (2004). 3-D optic disk reconstruction from low-resolution stereo images based on combined registrations and sub-pixel matching. *Conference Proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, *3*, 1825–1828. https://doi.org/10.1109/IEMBS.2004.1403544

Xu, J., & Chutatape, O. (2005). 3-d optic disk reconstruction via combined registration and inclusion of eye optical effects. *Conference Proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, *3*, 3284–3287. https://doi.org/10.1109/IEMBS.2005.1617178

Xu, J, Chutatape, O., Zheng, C., & Kuan, P. C. T. (2006). Three dimensional optic disc visualisation from stereo images via dual registration and ocular media optical correction. *The British Journal of Ophthalmology*, *90*(2), 181–185. https://doi.org/10.1136/bjo.2005.082313

Xu, Juan, & Chutatape, O. (2006). Auto-adjusted 3-D optic disk viewing from low-resolution stereo fundus image. *Computers in Biology and Medicine*, *36*(9), 921–940. https://doi.org/10.1016/j.compbiomed.2005.05.001

Xu, Juan, Ishikawa, H., Wollstein, G., Bilonick, R. A., Sung, K. R., Kagemann, L., Townsend, K. A., & Schuman, J. S. (2008). Automated assessment of the optic nerve head on stereo disc photographs. *Investigative Ophthalmology & Visual Science*, *49*(6), 2512–2517. https://doi.org/10.1167/iovs.07-1229

Yang, J., Wang, H., Ding, Z., Lv, Z., Wei, W., & Song, H. (2016). Local Stereo Matching Based on Support Weight With Motion Flow for Dynamic Scene. *IEEE Access*, *4*, 4840–4847. https://doi.org/10.1109/ACCESS.2016.2601069

Yoo, C., Amoozgar, B., Yang, K.-S., Park, J.-H., & Lin, S. C. (2018). Glaucoma severity and intraocular pressure reduction after cataract surgery in eyes with medically controlled glaucoma. *Medicine*, *97*(42). https://doi.org/10.1097/MD.0000000000012881

Zhang, K., Xi, X., Li, Z., & Wang, G. (2005). Stereo Matching and 3-D Reconstruction for Optic Disk Images. *CVBIA*. https://doi.org/10.1007/11569541_52

Zhu, Y.-M. (2007). A Java program for stereo retinal image visualization. *Computer Methods and Programs in Biomedicine*, *85*(3), 214–219. https://doi.org/10.1016/j.cmpb.2006.11.007

APPENDICES

APPENDIX 3A – PREFERRED REPORTING ITEMS FOR SYSTEMATIC REVIEWS AND META-ANALYSES EXTENSION FOR SCOPING REVIEWS (PRISMA-SCR) CHECKLIST (USED IN STEREO VISION SCOPING REVIEW)

SECTION	ITEM	PRISMA-ScR CHECKLIST ITEM	REPORTED ON PAGE #
TITLE			
Title	1	Identify the report as a scoping review.	38
ABSTRACT			
Structured summary	2	Provide a structured summary that includes (as applicable): background, objectives, eligibility criteria, sources of evidence, charting methods, results, and conclusions that relate to the review questions and objectives.	39
INTRODUCTION			
Rationale	3	Describe the rationale for the review in the context of what is already known. Explain why the review questions/objectives lend themselves to a scoping review approach.	40
Objectives	4	Provide an explicit statement of the questions and objectives being addressed with reference to their key elements (e.g., population or participants, concepts, and context) or other relevant key elements used to conceptualize the review questions and/or objectives.	41
METHODS			
Protocol and registration	5	Indicate whether a review protocol exists; state if and where it can be accessed (e.g., a Web address); and if available, provide registration information, including the registration number.	41
Eligibility criteria	6	Specify characteristics of the sources of evidence used as eligibility criteria (e.g., years considered, language, and publication status), and provide a rationale.	41
Information sources*	7	Describe all information sources in the search (e.g., databases with dates of coverage and contact with authors to identify additional sources), as well as the date the most recent search was executed.	42
Search	8	Present the full electronic search strategy for at least 1 database, including any limits used, such that it could be repeated.	42
Selection of sources of evidence†	9	State the process for selecting sources of evidence (i.e., screening and eligibility) included in the scoping review.	43
Data charting process‡	10	Describe the methods of charting data from the included sources of evidence (e.g., calibrated forms or forms that have been tested by the team before their use, and whether data charting was done independently or in duplicate) and any processes for obtaining and confirming data from investigators.	43
Data items	11	List and define all variables for which data were sought and any assumptions and simplifications made.	43
Critical appraisal of individual sources of evidence§	12	If done, provide a rationale for conducting a critical appraisal of included sources of evidence; describe the methods used and how this information was used in any data synthesis (if appropriate).	44
Synthesis of results	13	Describe the methods of handling and summarizing the data that were charted.	44
RESULTS			
Selection of sources of evidence	14	Give numbers of sources of evidence screened, assessed for eligibility, and included in the review, with reasons for exclusions at each stage, ideally using a flow diagram.	44
Characteristics of sources of evidence	15	For each source of evidence, present characteristics for which data were charted and provide the citations.	45

SECTION	ITEM	PRISMA-ScR CHECKLIST ITEM	REPORTED ON PAGE #
Critical appraisal within sources of evidence	16	If done, present data on critical appraisal of included sources of evidence (see item 12).	45
Results of individual sources of evidence	17	For each included source of evidence, present the relevant data that were charted that relate to the review questions and objectives.	45
Synthesis of results	18	Summarize and/or present the charting results as they relate to the review questions and objectives.	45
DISCUSSION			
Summary of evidence	19	Summarize the main results (including an overview of concepts, themes, and types of evidence available), link to the review questions and objectives, and consider the relevance to key groups.	50
Limitations	20	Discuss the limitations of the scoping review process.	51
Conclusions	21	Provide a general interpretation of the results with respect to the review questions and objectives, as well as potential implications and/or next steps.	51
FUNDING			
Funding	22	Describe sources of funding for the included sources of evidence, as well as sources of funding for the scoping review. Describe the role of the funders of the scoping review.	51

JBI = Joanna Briggs Institute; PRISMA-ScR = Preferred Reporting Items for Systematic reviews and Meta-Analyses extension for Scoping Reviews.

* Where sources of evidence (see second footnote) are compiled from, such as bibliographic databases, social media platforms, and Web sites.

† A more inclusive/heterogeneous term used to account for the different types of evidence or data sources (e.g., quantitative and/or qualitative research, expert opinion, and policy documents) that may be eligible in a scoping review as opposed to only studies. This is not to be confused with information sources (see first footnote).

‡ The frameworks by Arksey and O'Malley (6) and Levac and colleagues (7) and the JBI guidance (4, 5) refer to the process of data extraction in a scoping review as data charting.

§ The process of systematically examining research evidence to assess its validity, results, and relevance before using it to inform a decision. This term is used for items 12 and 19 instead of "risk of bias" (which is more applicable to systematic reviews of interventions) to include and acknowledge the various sources of evidence that may be used in a scoping review (e.g., quantitative and/or qualitative research, expert opinion, and policy document).

From: (Tricco et al., 2018)

APPENDIX 3B – SEARCH STRINGS USED IN STEREO VISION SCOPING REVIEW

Scopus

((TITLE(stereo)) *OR* (TITLE(stereo-matching)) *OR* (TITLE(stereo AND matching)) *OR* (TITLE(stereophotogrammetry)) *OR* (TITLE(stereo-photogrammetry)) *OR* (TITLE(stereo AND photogrammetry)) *OR* (TITLE(stereoscopic)) *OR* (TITLE(3D AND reconstruction)) *OR* (TITLE(3-D AND reconstruction)) *OR* (TITLE(3-dimensional AND reconstruction)) *OR* (TITLE(three-dimensional AND reconstruction)) *OR* (TITLE(three dimensional AND reconstruction)) *OR* (TITLE(3D AND visualization)) *OR* (TITLE(3-dimensional AND visualization)) *OR* (TITLE(three-dimensional AND visualization)) *OR* (TITLE(three dimensional AND visualization)) *OR* (TITLE(3D AND visualisation)) *OR* (TITLE(3-D AND visualisation)) *OR* (TITLE(3-dimensional AND visualisation)) *OR* (TITLE(three-dimensional AND visualisation)) *OR* (TITLE(three dimensional AND visualisation)) *OR* (TITLE(depth AND analysis)) *OR* (TITLE(topographic AND measurements)) *OR* (TITLE(topographic AND measurement))) **AND** ((TITLE(optic AND nerve AND head)) *OR* (TITLE(optic AND nerve)) *OR* (TITLE(optic-nerve)) *OR* (TITLE(optic AND disk)) *OR* (TITLE(optic AND disks)) *OR* (TITLE(optic-disk)) *OR* (TITLE(optic AND disc)) *OR* (TITLE(optic AND discs)) *OR* (TITLE(optic-disc)) *OR* (TITLE(optic AND cup)) *OR* (TITLE(optic AND cups)) *OR* (TITLE(optic-cup)) *OR* (TITLE(fundus)) *OR* (TITLE(retina)) *OR* (TITLE(retinal)))

Web of Science

((TI=(stereo)) *OR* (TI=(stereo-matching)) *OR* (TI=(stereo AND matching)) *OR* (TI=(stereophotogrammetry)) *OR* (TI=(stereo-photogrammetry)) *OR* (TI=(stereo AND photogrammetry)) *OR* (TI=(stereoscopic)) *OR* (TI=(3D AND reconstruction)) *OR* (TI=(3-D AND reconstruction)) *OR* (TI=(3-dimensional AND reconstruction)) *OR* (TI=(three-dimensional AND reconstruction)) *OR* (TI=(three dimensional AND reconstruction)) *OR* (TI=(3D AND visualization)) *OR* (TI=(3-dimensional AND visualization)) *OR* (TI=(three-dimensional AND visualization)) *OR* (TI=(three dimensional AND visualization)) *OR* (TI=(3D AND visualisation)) *OR* (TI=(3-D AND visualisation)) *OR* (TI=(3-dimensional AND visualisation)) *OR* (TI=(three-dimensional AND visualisation)) *OR* (TI=(three dimensional AND visualisation)) *OR* (TI=(depth AND analysis)) *OR* (TI=(topographic AND measurements)) *OR* (TI=(topographic AND measurement))) **AND** ((TI=(optic AND nerve AND head)) *OR* (TI=(optic AND nerve)) *OR* (TI=(optic-nerve)) *OR* (TI=(optic AND disk)) *OR* (TI=(optic AND disks)) *OR* (TI=(optic-disk)) *OR* (TI=(optic AND disc)) *OR* (TI=(optic AND discs)) *OR* (TI=(optic-disc)) *OR* (TI=(optic AND cup))

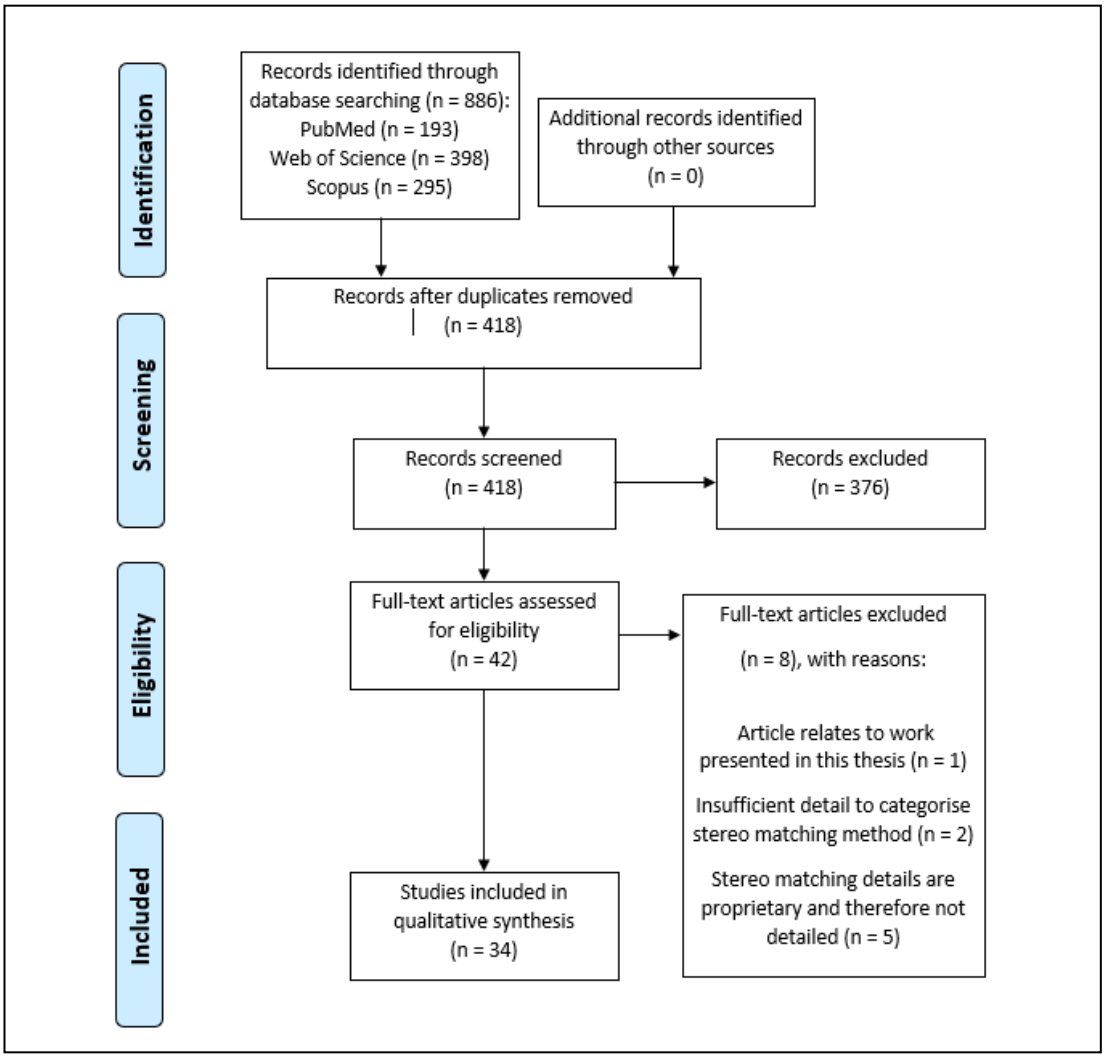
OR (TI=(optic AND cups)) *OR* (TI=(optic-cup)) *OR* (TI=(fundus)) *OR* (TI=(retina)) *OR* (TI=(retinal)))

PubMed

(((stereo[Title])) *OR* ((stereo-matching[Title])) *OR* ((stereo[Title] AND matching [Title])) *OR* ((stereophotogrammetry[Title])) *OR* ((stereo-photogrammetry[Title])) *OR* ((stereo[Title] AND photogrammetry[Title])) *OR* ((stereoscopic[Title])) *OR* ((3D[Title] AND reconstruction[Title])) *OR* ((3-D[Title] AND reconstruction[Title])) *OR* ((3-dimensional[Title] AND reconstruction[Title])) *OR* ((three-dimensional[Title] AND reconstruction[Title])) *OR* ((three dimensional[Title] AND reconstruction[Title])) *OR* ((3D[Title] AND visualization[Title])) *OR* ((3-dimensional[Title] AND visualization[Title])) *OR* ((three-dimensional[Title] AND visualization[Title])) *OR* ((three dimensional[Title] AND visualization[Title])) *OR* ((3D[Title] AND visualisation[Title])) *OR* ((3-D[Title] AND visualisation[Title])) *OR* ((3-dimensional[Title] AND visualisation[Title])) *OR* ((three-dimensional[Title] AND visualisation[Title])) *OR* ((three dimensional[Title] AND visualisation [Title])) *OR* ((depth[Title] AND analysis[Title])) *OR* ((topographic[Title] AND measurements[Title])) *OR* ((topographic[Title] AND measurement[Title]))) **AND** (((optic[Title] AND nerve[Title] AND head[Title])) *OR* ((optic[Title] AND nerve[Title])) *OR* ((optic-nerve [Title])) *OR* ((optic[Title] AND disk[Title])) *OR* ((optic[Title] AND disks[Title])) *OR* ((optic-disk[Title])) *OR* ((optic[Title] AND disc[Title])) *OR* ((optic[Title] AND discs[Title])) *OR* ((optic-disc[Title])) *OR* ((optic[Title] AND cup[Title])) *OR* ((optic[Title] AND cups[Title])) *OR* ((optic-cup[Title])) *OR* ((fundus[Title])) *OR* ((retina[Title])) *OR* ((retinal[Title])))

APPENDIX 3C – SEARCH FLOW DIAGRAM OF STEREO VISION SCOPING REVIEW

According to the PRISMA Statement (Tricco et al., 2018). n: number of articles.



APPENDIX 3D – CHARTING RESULTS OF STEREO VISION SCOPING REVIEW

Articles detailing global stereo matching algorithms

Study	Imaging Device	Stereo Matching Algorithm	Testing Performed
(Bansal et al., 2013)	1. Monoscopic Fundus Camera. [Commercial Device] (Asyncronous) 2. Nidek 3Dx Stereo Fundus Camera. [*Commercial Device*] (**Simultaneous**)	Global function minimised using graph cuts. Data term a function of the distance from the ONH centre.	1. In vivo (no validation performed). 2. In vivo (quantitative validation performed against OCT data).

(Zhang et al., 2005)	Not detailed.	Global function minimised using loopy belief propagation. Interpolation used to create final dense map.	In vivo (no validation performed).
(Otto et al., 1994)	Topcon TRC-SS2. [*Commercial Device*] (**Simultaneous**)	Matching algorithm based on model of the elastic membrane. Implemented using multigrid methods with the dynamic pyramid.	In vivo (quantitative comparison of retinal height along a defined contour line against a confocal laser scanning device).

Articles detailing hybrid stereo matching algorithms

Study	Imaging Device	Stereo Matching Algorithm	Testing Performed
(Norouzifard et al., 2018)	Not detailed.	Semi-global block matching.	In vivo (no validation performed).
(Gatti et al., 2014)	Not detailed.	Coarse-to-fine scheme. Patches matched using cross-correlation. Matching took place on images processed to have binary and grey-level features. Interpolation was used to obtain dense disparity information.	In vivo (cup-to-disc ratio measures determined by an automatic method on the 3D reconstructions were compared against ratios determined by an expert).
(Tang et al., 2012)	Topcon TRC-50DX Monoscopic Fundus Camera. [*Commercial Device*] (**Asynchronous**)	Fixed support window size but varied image resolution. Lower resolutions constrained search at higher resolutions. Intensity- and feature-based similarity metric. Disparity maps obtained using both left and right images as reference and fused to obtain final map.	In vivo, papilledema condition (compared reconstruction and volumetric measure against OCT data).
(Tang et al., 2011)	Nidek 3Dx Stereo Fundus Camera. [*Commercial Device*] (**Simultaneous**)	Fixed support window size but varied image resolution. Lower resolutions constrained search at higher resolutions. Intensity- and feature-based similarity metric. Disparity maps obtained using both left and right images as reference and fused to obtain final map.	In vivo (quantitative validation performed against OCT data).
(Tang et al., 2010)	Nidek 3Dx Stereo Fundus Camera. [*Commercial Device*] (**Simultaneous**)	Fixed support window size but varied image resolution. Lower resolutions constrained search at higher resolutions. Intensity- and feature-based similarity metric. Disparity maps obtained using both left and right images as reference and fused to obtain final map.	In vivo (quantitative validation performed against OCT data).
(Lusk & Nutter, 2009)	Not detailed.	Coarse-to-fine scheme. Patches matched using cross-correlation. Matching took place on images processed to have binary and grey-level features. Interpolation was used to obtain dense disparity information.	In vivo (compared cup-to-disc ratios determined by applying automatic margin segmentation method on reconstructions obtained using reconstruction algorithm before and after changing rectification method).
(Zhu, 2007)	Nidek 3Dx Stereo Fundus Camera. [*Commercial Device*] (**Simultaneous**) and Monoscopic Fundus	Coarse-to-fine scheme. Patches matched using cross-correlation. Matching took place on images processed to have binary and grey-level features. Interpolation was used to obtain dense disparity information.	In vivo (no validation performed).

	Camera [*Commercial Device*] (**Asynchronous**)		
(Sharma et al., 2006)	Monoscopic Fundus Camera. [*Commercial Device*] (**Asynchronous**)	Coarse-to-fine scheme. Patches matched using cross-correlation. Matching took place on images processed to have binary and grey-level features. Interpolation was used to obtain dense disparity information.	In vivo (cup and disc margins determined using their automatic margin segmentation method were compared with those marked by an expert).
(Juan Xu & Chutatape, 2006)	Not detailed.	Initial coarse disparity map (used to constrain the search of disparities at finer levels) obtained using large windows. Matching was then performed for small to large window sizes, until a reliable match was found for 80% of the points. Reliable matches were those that gave disparity values, using two different metrics (intensity- and feature-based), that were within 1 pixel of each other. Interpolation was used to obtain dense disparity information.	In vivo (validation not performed).
(Kong et al., 2004)	Topcon TRC-SS2. [*Commercial Device*] (**Simultaneous**)	Patches matched using cross-correlation of binarized images. Patch size was varied from small to large, and the final disparity map was generated from the results at all levels.	In vivo (no validation performed).
(Corona, Mitra, Wilson, Krile, et al., 2002)	Not detailed.	Coarse-to-fine scheme. Patches matched using cross-correlation (only applied for most probable shifts - as determined from cepstral analysis). Matching took place on images processed to have binary and grey-level features. Interpolation was used to obtain dense disparity information.	In vivo (cup-to-disc ratio measures determined by a semi-automatic method on the 3D reconstructions were compared against ratios determined by an experienced ophthalmologist).
(Corona, Mitra, Wilson, & Soliz, 2002)	Not detailed.	Coarse-to-fine scheme. Patches matched using cross-correlation (only applied for most probable shifts - as determined from cepstral analysis). Matching took place on images processed to have binary and grey-level features. Interpolation was used to obtain dense disparity information.	In vivo (cup-to-disc ratio measures determined by a semi-automatic method on the 3D reconstructions were compared against ratios determined by an experienced ophthalmologist).
(J. M. Ramirez et al., 1999)	Not detailed.	Coarse-to-fine scheme. Patches are matched using the cepstral transformation – applied to binary images to obtain an impulse train whereby the disparity is determined using the first signal peak. Interpolation was used to obtain dense disparity information.	1. Stereogram with half sphere structure (quantitative validation performed). 2. In vivo (no validation performed).

(M. Ramirez et al., 1992)	Not detailed.	Coarse-to-fine scheme. Patches are matched using the cepstral transformation – applied to binary images to obtain an impulse train whereby the disparity is determined using the first signal peak. Interpolation was used to obtain dense disparity information.	1. Concave ball, not in eye phantom (no validation performed) 2. In vivo (no validation performed).
(Okutomi et al., 1992)	Not detailed.	Initial disparity map estimated using sum of the squared differences metric summed over all three colour channels. Then matching is performed again iteratively from this initial map using intensity and disparity information within support regions until disparity values converge.	1. In vivo (no validation) 2. Monkey (no validation).

Articles detailing local stereo matching algorithms

Study	Imaging Device	Stereo Matching Algorithm	Testing Performed
(Guo et al., 2018)	Not detailed.	Sum of absolute differences metric.	In vivo (quantitative validation performed against OCT data).
(D. Motta et al., 2015)	Developed Low-Cost Stereo Fundus Camera. [*Research Prototype*] (**Simultaneous**)	Performed patched based matching for every 4th pixel using a correlation-based similarity metric which used data from all three RGB colour channels and the edge images of those colour channels. Multiple stereo image pairs were taken, and an all-in-focus technique was used to fuse the best focus regions of them to create a single image pair with superior focus (or quality). Smoothed resulting maps with a DBSCAN filtering method.	ONH models, not inside eye phantom (physically measured ONH model and made the comparison against the 3D reconstruction).
(Carvalho & Carvalho, 2014)	Developed Low-Cost Stereo Fundus Camera. [*Research Prototype*] (**Simultaneous**)	Combined window and feature based method.	In vivo (no validation performed).
(D. A. Motta et al., 2014)	Developed Low-Cost Stereo Fundus Camera. [*Research Prototype*] (**Simultaneous**)	Performed patched based matching for every 4th pixel using a correlation-based similarity metric which used data from all three RGB colour channels and the edge images of those colour channels. Smoothed resulting maps with a DBSCAN filtering method.	ONH models, not inside eye phantom (compared cross-sectional profile of reconstruction against profile created by manual measurement).
(Muramatsu et al., 2014)	Kowa WX-1. [*Commercial Device*] (**Almost Simultaneous**)	Performed patched based matching for every 4th pixel using a correlation-based similarity metric which used data from all three RGB colour channels and the edge images of those colour channels. Smoothed resulting maps with a noise reduction step.	In vivo (no validation of reconstruction).

(Muramatsu et al., 2011)	Kowa WX-1. [*Research Prototype*] (**Almost Simultaneous**)	Performed patched based matching for every 4th pixel using a correlation-based similarity metric which used data from all three RGB colour channels and the edge images of those colour channels. Smoothed resulting maps with a noise reduction step.	In vivo (quantitative comparison between cup and disc areas determined by an automatic method on the 3D reconstructions and that obtained by an experienced ophthalmologist).
(Muramatsu et al., 2009)	Kowa WX-1. [*Research Prototype*] (**Almost Simultaneous**)	Performed patched based matching for every 4th pixel using a correlation-based similarity metric which used data from all three RGB colour channels and the edge images of those colour channels. Smoothed resulting maps with a noise reduction step.	In vivo (quantitative comparison between cup and disc areas determined by an automatic method on the 3D reconstructions and that obtained by an experienced ophthalmologist).
(Nakagawa et al., 2008)	Kowa WX-1. [*Research Prototype*] (**Almost Simultaneous**)	Performed patched based matching for every 4th pixel using a correlation-based similarity metric which used data from all three RGB colour channels and the edge images of those colour channels. Smoothed resulting maps with a noise reduction step.	T1: Eye model (physically measured ONH model diameter and depth and made the comparison against the reconstruction obtained). T2: In vivo (qualitative validation performed against HRT data, and quantitative validation of all results by way of cross-sectional slice comparison, which included the RMS difference between such slices).
(Juan Xu et al., 2008)	Nidek 3Dx Stereo Fundus Camera. [*Commercial Device*] (**Simultaneous**)	Created two sparse disparity maps from greyscale images. One using cross-correlation as the similarity metric and the other using a feature-based metric. Points in agreement between the two maps were kept and interpolation was used to generate dense maps.	In vivo (ONH parameters, including volumetric measures, were determined from the 3D reconstructions using automatically detected cup and disc margins, and margins subjectively delineated by experienced ophthalmologists, and a comparison was made).
(Nakagawa, Hayashi, Hatanaka, Aoyama, et al., 2007)	Kowa WX-1. [*Research Prototype*] (**Almost Simultaneous**)	Performed patched based matching for every 4th pixel using a correlation-based similarity metric which used data from all three RGB colour channels and the edge images of those colour channels.	In vivo (qualitative validation performed against HRT data, and quantitative validation of one result by way of cross-sectional slice comparison, which included the RMS difference between such slices).
(Nakagawa, Hayashi, Hatanaka, M.d, et al., 2007)	Kowa WX-1. [*Research Prototype*] (**Almost Simultaneous**)	Performed patched based matching for every 4th pixel using a correlation-based similarity metric which used data from all three RGB colour channels and the edge images of those colour channels.	T1: Eye model (physically measured ONH model diameter and depth and made the comparison against the 3D reconstruction). T2: In vivo (qualitative validation performed against HRT data).

(J Xu et al., 2006)	Not detailed.	Created two sparse disparity maps from greyscale images. One using cross-correlation as the similarity metric and the other using a feature-based metric. Points in agreement between the two maps were kept and interpolation was used to generate dense maps.	In vivo (cup-to-disc ratios determined by an automatic method on the 3D reconstructions were compared against ratios determined from the HRT device and by an experienced ophthalmologist).
(J. Xu & Chutatape, 2005)	Not detailed.	Created two sparse disparity maps from greyscale images. One using cross-correlation as the similarity metric and the other using a feature-based metric. Points in agreement between the two maps were kept and interpolation was used to generate dense maps.	In vivo (qualitative validation performed against HRT data).
(J. Xu & Chutatape, 2004)	Not detailed.	Created two sparse disparity maps from greyscale images. One using cross-correlation as the similarity metric and the other using a feature-based metric. Points in agreement between the two maps were kept and interpolation was used to generate dense maps.	In vivo (no validation performed).
(Wünstel & Schumann, 2002, p. 3)	Not detailed.	Correlation based technique.	In vivo (no validation performed).
(Berestov, 2000)	Canon Digital MVC 5000 Retinal Imaging System. [*Commercial Device*] (Asynchronous)	Correlation based technique.	In vivo (qualitative validation against scan from laser scanning system).

APPENDIX 4A – PREFERRED REPORTING ITEMS FOR SYSTEMATIC REVIEWS AND META-ANALYSES EXTENSION FOR SCOPING REVIEWS (PRISMA-SCR) CHECKLIST (USED IN THE EYE PHANTOM REVIEW)

SECTION	ITEM	PRISMA-ScR CHECKLIST ITEM	REPORTED ON PAGE #
TITLE			
Title	1	Identify the report as a scoping review.	52
ABSTRACT			
Structured summary	2	Provide a structured summary that includes (as applicable): background, objectives, eligibility criteria, sources of evidence, charting methods, results, and conclusions that relate to the review questions and objectives.	53
INTRODUCTION			
Rationale	3	Describe the rationale for the review in the context of what is already known. Explain why the review questions/objectives lend themselves to a scoping review approach.	54
Objectives	4	Provide an explicit statement of the questions and objectives being addressed with reference to their key elements (e.g., population or participants, concepts, and context) or other relevant key elements used to conceptualize the review questions and/or objectives.	54
METHODS			
Protocol and registration	5	Indicate whether a review protocol exists; state if and where it can be accessed (e.g., a Web address); and if available, provide registration information, including the registration number.	54

SECTION	ITEM	PRISMA-ScR CHECKLIST ITEM	REPORTED ON PAGE #
Eligibility criteria	6	Specify characteristics of the sources of evidence used as eligibility criteria (e.g., years considered, language, and publication status), and provide a rationale.	54
Information sources*	7	Describe all information sources in the search (e.g., databases with dates of coverage and contact with authors to identify additional sources), as well as the date the most recent search was executed.	55
Search	8	Present the full electronic search strategy for at least 1 database, including any limits used, such that it could be repeated.	55
Selection of sources of evidence†	9	State the process for selecting sources of evidence (i.e., screening and eligibility) included in the scoping review.	56
Data charting process‡	10	Describe the methods of charting data from the included sources of evidence (e.g., calibrated forms or forms that have been tested by the team before their use, and whether data charting was done independently or in duplicate) and any processes for obtaining and confirming data from investigators.	56
Data items	11	List and define all variables for which data were sought and any assumptions and simplifications made.	56
Critical appraisal of individual sources of evidence§	12	If done, provide a rationale for conducting a critical appraisal of included sources of evidence; describe the methods used and how this information was used in any data synthesis (if appropriate).	56
Synthesis of results	13	Describe the methods of handling and summarizing the data that were charted.	56
RESULTS			
Selection of sources of evidence	14	Give numbers of sources of evidence screened, assessed for eligibility, and included in the review, with reasons for exclusions at each stage, ideally using a flow diagram.	57
Characteristics of sources of evidence	15	For each source of evidence, present characteristics for which data were charted and provide the citations.	57
Critical appraisal within sources of evidence	16	If done, present data on critical appraisal of included sources of evidence (see item 12).	57
Results of individual sources of evidence	17	For each included source of evidence, present the relevant data that were charted that relate to the review questions and objectives.	57
Synthesis of results	18	Summarize and/or present the charting results as they relate to the review questions and objectives.	57
DISCUSSION			
Summary of evidence	19	Summarize the main results (including an overview of concepts, themes, and types of evidence available), link to the review questions and objectives, and consider the relevance to key groups.	60
Limitations	20	Discuss the limitations of the scoping review process.	60
Conclusions	21	Provide a general interpretation of the results with respect to the review questions and objectives, as well as potential implications and/or next steps.	61
FUNDING			
Funding	22	Describe sources of funding for the included sources of evidence, as well as sources of funding for the scoping review. Describe the role of the funders of the scoping review.	61

JBI = Joanna Briggs Institute; PRISMA-ScR = Preferred Reporting Items for Systematic reviews and Meta-Analyses extension for Scoping Reviews.

* Where sources of evidence (see second footnote) are compiled from, such as bibliographic databases, social media platforms, and Web sites.

† A more inclusive/heterogeneous term used to account for the different types of evidence or data sources (e.g., quantitative and/or qualitative research, expert opinion, and policy

documents) that may be eligible in a scoping review as opposed to only studies. This is not to be confused with information sources (see first footnote).

‡ The frameworks by Arksey and O'Malley (6) and Levac and colleagues (7) and the JBI guidance (4, 5) refer to the process of data extraction in a scoping review as data charting.

§ The process of systematically examining research evidence to assess its validity, results, and relevance before using it to inform a decision. This term is used for items 12 and 19 instead of "risk of bias" (which is more applicable to systematic reviews of interventions) to include and acknowledge the various sources of evidence that may be used in a scoping review (e.g., quantitative and/or qualitative research, expert opinion, and policy document).

From: (Tricco et al., 2018)

APPENDIX 4B - SEARCH STRINGS USED IN EYE PHANTOM SCOPING REVIEW

Scopus

TITLE(eye) **AND** TITLE(phantom OR model) **AND** ABS(fabricated OR constructed OR built OR construct OR fabricate OR build OR assemble OR assembled) **AND NOT** TITLE(rabbit OR mouse OR chick OR rat OR mathematical OR gaze OR tracking OR movement)

Web of Science

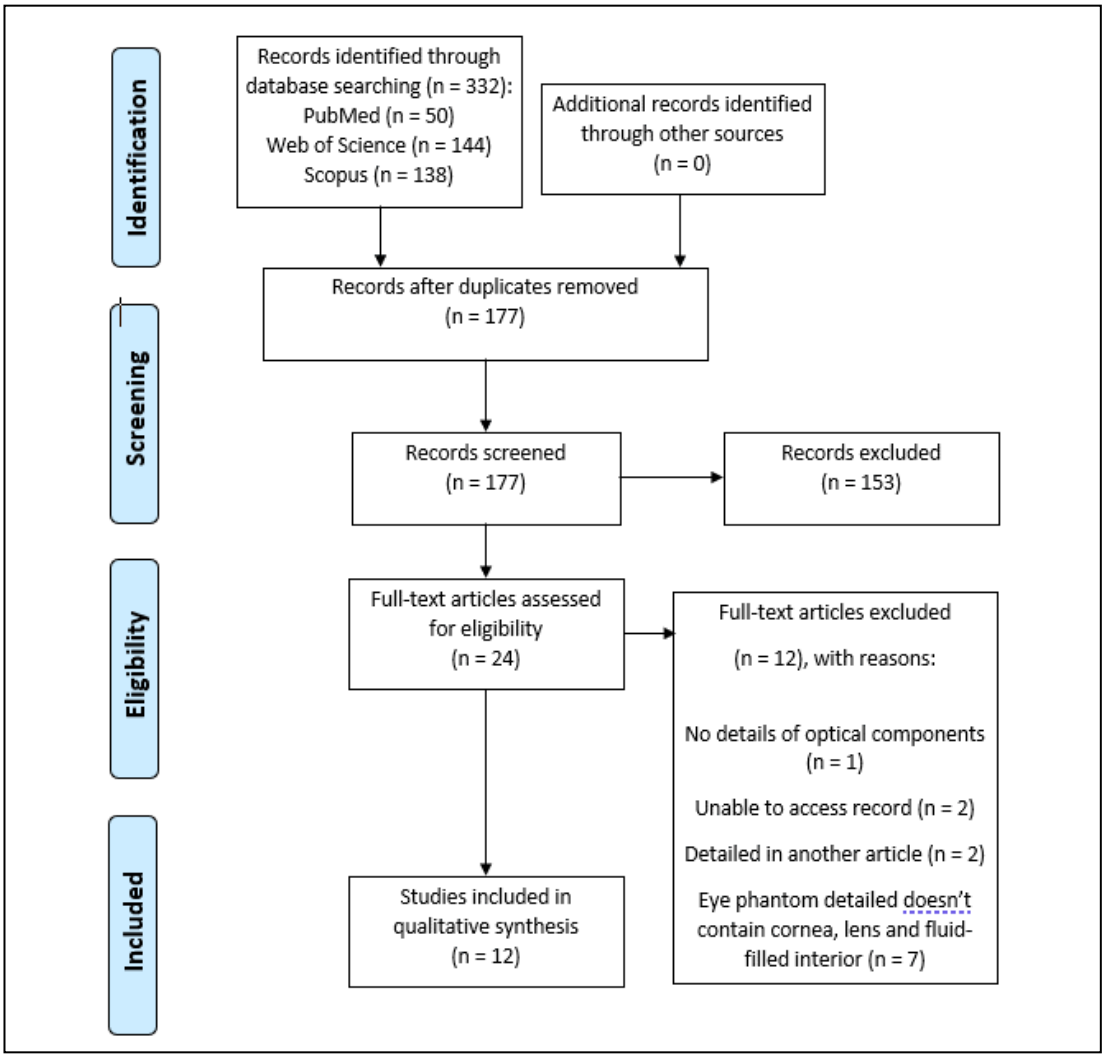
((TI=(eye))) **AND** ((TI=(phantom)) OR (TI=(model))) **AND** ((AB=(fabricated)) OR (AB=(constructed)) OR (AB=(built)) OR (AB=(construct)) OR (AB=(fabricate)) OR (AB=(build)) OR (AB=(assemble)) OR (AB=(assembled))) **NOT** ((TI=(rabbit)) OR (TI=(mouse)) OR (TI=(chick)) OR (TI=(rat)) OR (TI=(mathematical)) OR (TI=(gaze)) OR (TI=(tracking)) OR (TI=(movement)))

PubMed

(eye[Title]) **AND** (phantom[Title] OR model [Title]) **AND** (fabricated[Title/Abstract] OR constructed[Title/Abstract] OR built[Title/Abstract] OR construct[Title/Abstract] OR fabricate[Title/Abstract] OR build[Title/Abstract] OR assemble[Title/Abstract] OR assembled[Title/Abstract]) **NOT** (rabbit[Title] OR mouse[Title] OR chick[Title] OR rat[Title] OR mathematical[Title] OR gaze[Title] OR tracking[Title] OR movement[Title])

APPENDIX 4C – SEARCH FLOW DIAGRAM OF EYE PHANTOM SCOPING REVIEW

According to the PRISMA Statement (Tricco et al., 2018). n: number of articles.



APPENDIX 5A – MATLAB CODE FOR THE STEREO VISION PIPELINE USED IN CHAPTER 5

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%              Uncalibrated Stereo Reconstruction Code              %%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear DM
clear fMatrix
close all

%% Import and Preprocess Images
%Load Images
I1 = imread('L7.tif');
I2 = imread('R7.tif');

%Correct Images for Distortion
% I1 = undistortImage(I1,stereoParams.CameraParameters1);
% I2 = undistortImage(I2,stereoParams.CameraParameters2);

%Apply Adaptive Histogram Equalization
LAB = rgb2lab(I1); %Left Image
L = LAB(:,:,1)/100;
L = adapthisteq(L,'NumTiles',[8 8],'ClipLimit',0.005);
LAB(:,:,1) = L*100;
I1 = lab2rgb(LAB);
LAB = rgb2lab(I2); %Right Image
L = LAB(:,:,1)/100;
L = adapthisteq(L,'NumTiles',[8 8],'ClipLimit',0.005);
LAB(:,:,1) = L*100;
I2 = lab2rgb(LAB);

%Convert Images to Grayscale
I1gray = rgb2gray(I1);
I2gray = rgb2gray(I2);

%Display Colour Images Side-by-Side
figure(1);
imshowpair(I1, I2,'montage');
title('I1 (left); I2 (right)');
figure(8);
imshow(stereoAnaglyph(I1,I2));
title('Composite Image (Red - Left Image, Cyan - Right Image)');

%% Perform Rectification
%Detect Features in Both Images
blobs1 = detectSURFFeatures(I1gray, 'MetricThreshold', 100);
blobs2 = detectSURFFeatures(I2gray, 'MetricThreshold', 100);

%Visualise the 30 Strongest Features
figure(2);
imshow(I1);
hold on;
plot(selectStrongest(blobs1, 30));
title('Thirty strongest SURF features in I1');
figure(3);
imshow(I2);
hold on;
plot(selectStrongest(blobs2, 30));
title('Thirty strongest SURF features in I2');

%Extract Features and Perform Matching to Find Potential Correspondences
[features1, validBlobs1] = extractFeatures(I1gray, blobs1);
[features2, validBlobs2] = extractFeatures(I2gray, blobs2);
indexPairs = matchFeatures(features1, features2, 'Metric', 'SAD','MatchThreshold', 5);

%Retrieve the Locations of the Matched Points for Each Image
matchedPoints1 = validBlobs1(indexPairs(:,1),:);
matchedPoints2 = validBlobs2(indexPairs(:,2),:);
```

```
%Display the Matched Features
figure(4);
showMatchedFeatures(I1, I2, matchedPoints1, matchedPoints2);
legend('Putatively matched points in I1', 'Putatively matched points in I2');

%Filter out Points Which do not Agree with the Epipolar Geometry
[fMatrix, epipolarInliers, status] = estimateFundamentalMatrix(...
  matchedPoints1, matchedPoints2, 'Method', 'RANSAC', ...
  'NumTrials', 10000, 'DistanceThreshold', 0.1, 'Confidence', 99.99);
if status ~= 0 || isEpipoleInImage(fMatrix, size(I1)) ...
  || isEpipoleInImage(fMatrix', size(I2))
  error(['Either not enough matching points were found or '...
         'the epipoles are inside the images. You may need to '...
         'inspect and improve the quality of detected features ',...
         'and/or improve the quality of your images.']);
end
inlierPoints1 = matchedPoints1(epipolarInliers, :);
inlierPoints2 = matchedPoints2(epipolarInliers, :);

figure(5);
showMatchedFeatures(I1, I2, inlierPoints1, inlierPoints2);
legend('Inlier points in I1', 'Inlier points in I2');

%Estimate the Rectification Transformations
[t1, t2] = estimateUncalibratedRectification(fMatrix, inlierPoints1.Location,
inlierPoints2.Location, size(I2));
tform1 = projective2d(t1);
tform2 = projective2d(t2);

%Rectify the Stereo Images, and Display them as a Stereo Anaglyph
[I1Rect, I2Rect] = rectifyStereoImages(I1, I2, tform1, tform2);
figure();
imshow(stereoAnaglyph(I1Rect, I2Rect));
title('Rectified Stereo Images (Red - Left Image, Cyan - Right Image)');
figure ();
imshow(I1Rect);
title('I1Rect');

%% Perform Stereo Matching
%Stereo Match
SAD = 17;
disparityRange = [-16 16];
disparityMap =
disparity(rgb2gray(I1Rect),rgb2gray(I2Rect),'BlockSize',SAD,'DisparityRange',disparityRange);
disparityMap = medfilt2(disparityMap);

%Display Disparity Map
figure ();
imshow(disparityMap);
title('Disparity Map');

%% Display as a 3D Scatter Plot, with RGB data Overlayed, and Colour Disparity Map
%Initialisations
s = size(disparityMap);
w = s(1);
h = s(2);
z=1;
clear DM
clear Col
XC = 300; %X coordinate of central point of circle crop
YC = 500; %Y coordinate of central point of circle crop
CropRad = 250; %Circle Crop Radius (pixels)

%Build Up x,y,z and r,g,b Matrices
for i = 1:w-1
    for j = 1:h-1
        rad = sqrt(((i-XC)*(i-XC))+((j-YC)*(j-YC)));

        if rad < CropRad

        DM(z,1)=i;
```

```
        DM(z,2)=j;

        Col(z,1) = I1Rect(i, j, 1);
        Col(z,2) = I1Rect(i, j, 2);
        Col(z,3) = I1Rect(i, j, 3);
        r = disparityMap(i,j);
        DM(z,3)= r;
        S(i,j) = r;

            if r > -500 && r<500

                z = z + 1;

            end
        end

    end
end

%Show the Colour Disparity Map
figure()
hold on
C=DM(:,3);
scatter3(DM(:,1),DM(:,2),DM(:,3),50,C(:),'.');
colormap(jet);
caxis([-2.5 0.5])
xlabel('X (pixels)');
ylabel('Y (pixels');
zlabel('Disparity (pixels)');
zlim([-3 4.8])
xlim([10 410])
ylim([315 715])

%Show the Textured Disparity Map
figure()
hold on
scatter3(DM(:,1),DM(:,2),DM(:,3),50,Col,'.');
xlabel('X (pixels)');
ylabel('Y (pixels)');
zlabel('Disparity (pixels)');
zlim([-3 4.8])
xlim([100 500])
ylim([200 600])
```

APPENDIX 6A – INTERPOLATION CODE WRITTEN IN MATLAB FOR THE WORK OF CHAPTER 6

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%    Cubic B-Spline Interpolation of Sparse Disparity Map         %%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear all
close all

%% Import Data
S = readtable('sparseCloudData.csv');
S = table2array(S(:, 1:3));

%% Calculate Horizontal Splines
%Initialisations
c = 1;
d = 1;
p = 1;
figure(1);
xlabel('X');
ylabel('Disparity');

%Loop to calculate horizontal splines
for i=1:40

    for j=1:40
        TempMat(j,:) = S(c,:);
        c = c + 1;
    end
    TempMat_no_nans = TempMat(~any(isnan(TempMat),2),:); %keep rows which have no nans
    [curve, goodness, output] =
fit(TempMat_no_nans(:,1),TempMat_no_nans(:,3),'smoothingspline', 'SmoothingParam',0.000004);
    plot(curve,TempMat_no_nans(:,1),TempMat_no_nans(:,3));

    for r = TempMat_no_nans(1,1):TempMat_no_nans((length(TempMat_no_nans)),1)
        SS(d,1) = r;
        SS(d,2) = TempMat_no_nans(1,2);
        SS(d,3) = curve(r);
        d = d + 1;
    end
    clear TempMat TempMat_no_nans
end

%Sort rows
SSS = sortrows(SS,1);

%% Calculate Vertical Splines
%Initialisations
figure(3);
xlabel('Y');
ylabel('Disparity');
SSSTemp = SSS(1,1);
j = 1;
p = 1;
d = 1;

%Loop to calculate vertical splines
for i=1:length(SSS)

    if SSS(i,1) > SSSTemp
        TempMat_no_nans = TempMat(~any(isnan(TempMat),2),:); %keep rows which have no nans
        [curve, goodness, output] =
fit(TempMat_no_nans(:,2),TempMat_no_nans(:,3),'smoothingspline', 'SmoothingParam',0.000004);
        plot(curve,TempMat_no_nans(:,2),TempMat_no_nans(:,3));
        j = 1;
        size = -TempMat_no_nans(1,2)+TempMat_no_nans((length(TempMat_no_nans)),2)+1;
        SSF(d:d+size-1,2) =
TempMat_no_nans(1,2):TempMat_no_nans((length(TempMat_no_nans)),2);
        SSF(d:d+size-1,1) = TempMat_no_nans(1,1);
```

```
        SSF(d:d+size-1,3) =
curve(TempMat_no_nans(1,2):TempMat_no_nans((length(TempMat_no_nans)),2));
        d = d + size;
        clear TempMat TempMat_no_nans
    end
    SSSTemp = SSS(i,1);
    TempMat(j,:) = SSS(i,:);
    j = j + 1;
    SSSTemp
    SSS(i,1)


end

%Write the data (dense disparity map) to file
writematrix(SSF,'interpolatedDisparityMap.csv');
```

APPENDIX 6B – STEREO VISION PIPELINE WRITTEN IN C++ FOR THE WORK OF CHAPTER 6

```
////////////////////////////////////////////////////////////////////////////////
//////////////                3D Reconstruction Code                //////////////
////////////////////////////////////////////////////////////////////////////////


// Reconstruction Mode
int ReconMode = 1;                                                    // 1 = Perform
Stereo Matching, 0 = Reproject Interpolated Disparity Map

/*
Author: Ian Coghill(1) (EngD Student)
Supervised by Dr Mario E Giardini(1), Dr Richard Black(1) & Dr Iain AT Livingstone(2)
1 - Department of Biomedical Engineering, University of Strathclyde
2 - NHS Forth Valley
*/



////////////////////////////////   INCLUDE LIBRARIES   ////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////
#include <iostream>
                                              // Misc
#include <fstream>
#include <string>
#include <conio.h>
#include <Eigen/Core>
#include <Eigen/QR>
#include <cmath>
#include <direct.h>
#include <stdlib.h>
#include <string.h>
#include "opencv2/core/core.hpp"
                        // OpenCV
#include "opencv2/core/eigen.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/calib3d/calib3d.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/imgcodecs.hpp"
#include "pcl/common/common.h"
// Point Cloud Library
#include "pcl/point_types.h"
#include "pcl/point_cloud.h"
#include "pcl/visualization/pcl_visualizer.h"
#include "pcl/common/centroid.h"
#include <pcl/common/transforms.h>
#include <pcl/surface/poisson.h>
#include <pcl/io/vtk_io.h>
#include <pcl/io/ply_io.h>
#include <pcl/features/normal_3d.h>
#include <pcl/surface/mls.h>
#include <boost/thread/thread.hpp>
#include <pcl/filters/statistical_outlier_removal.h>



////////////////////////////////   RECONSTRUCTION SETTINGS   ////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////
int             noOfSquares = 40;                                     // Number of points
in grid
double  correlationThreshold = 0.80;            // Correlation threshold
int             pxspc = 1;                                                    // Dense
reconstruction resolution paramter (spacing between pixels reprojected)
int             showProcess = 0;                                  // 1 = Show stereo
matching, 0 = don't show (for debugging)
int             minlvl = 3;                                                   // Lower
limit  of coarseness level
int             numlvls = 5;                                          // Number of window
size (coarseness) levels
```

```
double  TEXTTHRESH = 0.15;                                          // Texture Treshold
double  ConsistTHRESH = 1.5;                                    // Left-right consistency threshold


//////////////////////////  INITIALISE POINT CLOUD OBJECTS   //////////////////////////
////////////////////////////////////////////////////////////////////////////////////////
pcl::PointCloud<pcl::PointXYZRGB>::Ptr denseTexturedCloud(new
pcl::PointCloud<pcl::PointXYZRGB>);
Eigen::Affine3f tMatrix;


//////////////////////   INITIALISE VISUALISATION FUNCTIONS   //////////////////////
////////////////////////////////////////////////////////////////////////////////////////
void CloudVisualisation();



//////////////////////////   OTHER FUNCTIONS INITIALISATION   //////////////////////////
////////////////////////////////////////////////////////////////////////////////////////
int disparitySearchCentreEstimation(cv::Rect, cv::Mat, cv::Mat);
void computeSparsePointCloud(cv::Mat, cv::Mat, int, cv::Rect2d, int, double, cv::Mat);
cv::Rect2d ROISelection(cv::Mat);



////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////
/////////////////////////////////////   MAIN   /////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////
int main()
{
        std::cout << R"(

+------+.      +------+       +------+       +------+      .+------+
|`.    | `.    |\     |\      |      |      /|     /|    .' |    .'|
|  `+--+---+   | +----+-+     +------+     +-+----+ |   +---+--+'  |
|   |  |   |   | |    | |     |      |     | |    | |   |   |  |   |
+---+--+.  |   +-+----+ |     +------+     | +----+-+   |  .+--+---+
 `. |    `.|    \|     \|     |      |     |/     |/    |.'    | .'
   `+------+     +------+     +------+     +------+     +------+'

                        3D RECONSTRUCTION CODE
        )" << '\n';



        ////////////////////////////////   VARIABLE INITIALISATION
//////////////////////////////
        cv::Mat         RGBimageL, RGBimageR;
                                                // Images
        cv::Mat         RGBimageL_Rect, RGBimageR_Rect;
        cv::Mat         inputStereoImages, rectifiedImages, newImage3;

        cv::Mat K1, D1, K2, D2;
                                // Rectification
        cv::Mat Q, R, R1, P1, R2, P2;
        cv::Vec3d       T;
        cv::Mat         map11, map12, map21, map22, img1r, img2r;
        cv::Rect        roi1, roi2;
        cv::Point       Point1, Point2;

        int     dispCentre;
                            // Stereo Matching Variables



        ////////////////////////   LOAD STEREO CALIBRATION PARAMETERS
////////////////////////
        cv::FileStorage fs("Data/stereoParams.yml", cv::FileStorage::READ);
```

```
        fs["K1"] >> K1; fs["D1"] >> D1;
        fs["K2"] >> K2; fs["D2"] >> D2;
        fs["R"] >> R;
        fs["T"] >> T;



        ////////////////////////////////////   LOAD IMAGES
////////////////////////////////
        RGBimageL = cv::imread("Data/L.png");
        RGBimageR = cv::imread("Data/R.png");
        std::cout << "Size of Left Image: " << RGBimageL.size() << std::endl;
        std::cout << "Size of Right Image: " << RGBimageR.size() << std::endl;



        ////////////////////////////////   DISPLAY INPUT IMAGES
//////////////////////////////
        cv::namedWindow("Input Stereo Image Pair", cv::WINDOW_NORMAL);
        cv::hconcat(RGBimageL, RGBimageR, inputStereoImages);
        imshow("Input Stereo Image Pair", inputStereoImages);
        cv::waitKey();



        ////////////////////////////////   RECTIFY INPUT IMAGES
//////////////////////////////
        cv::Size img_size = (RGBimageL.size());
        stereoRectify(K1, D1, K2, D2, img_size, R, T, R1, R2, P1, P2, Q,
cv::CALIB_ZERO_DISPARITY, -1, img_size, &roi1, &roi2);
        cv::initUndistortRectifyMap(K1, D1, R1, P1, img_size, CV_16SC2, map11, map12);
        cv::initUndistortRectifyMap(K2, D2, R2, P2, img_size, CV_16SC2, map21, map22);
        Q.convertTo(Q, CV_32F);
        cv::remap(RGBimageL, RGBimageL_Rect, map11, map12, cv::INTER_LINEAR);
        cv::remap(RGBimageR, RGBimageR_Rect, map21, map22, cv::INTER_LINEAR);



        ////////////////////////   DISPLAY AND SAVE RECTIFIED IMAGES
////////////////////////
        cv::namedWindow("Rectified Images", cv::WINDOW_NORMAL);
        cv::hconcat(RGBimageL_Rect, RGBimageR_Rect, rectifiedImages);
        imshow("Rectified Images", rectifiedImages);
        cv::waitKey();
        cv::imwrite("Data/RectifiedLeftImage.png", RGBimageL_Rect);
        cv::imwrite("Data/RectifiedRightImage.png", RGBimageR_Rect);



        ///////////////////////   SELECT REGION IN IMAGE TO BE ANALYSED
//////////////////////
        cv::Rect2d ROI = ROISelection(RGBimageL_Rect);



        ////////////////////////////   GENERATE 3D RECONSTRUCTIONS
////////////////////////////
        // Disparity Search Centre Estimation
        dispCentre = disparitySearchCentreEstimation(ROI, RGBimageR_Rect, RGBimageL_Rect);

        // Estimate Sparse Point Cloud
        computeSparsePointCloud(RGBimageL_Rect, RGBimageR_Rect, noOfSquares, ROI, dispCentre,
correlationThreshold, Q);
        boost::thread workerThread(CloudVisualisation);
        cv::waitKey();

}
////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////
/////////////////////////////////////   MAIN END  ////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////////////////////


int disparitySearchCentreEstimation(cv::Rect ROI, cv::Mat RightImageRect, cv::Mat
LeftImageRect)
{
	////////////////////////////////////////// DISPARITY SEARCH CENTRE ESTIMATION
	std::cout << "Now estimating the disparity search centre" << std::endl;
	int boxWidthDS = (ROI.width);
	int boxHeightDS = (ROI.height);
	cv::Point TopLeftPoint = cv::Point(ROI.x, ROI.y);
	cv::Mat resultDS, LeftImageRectGrey, RightImageRectGrey, LeftImageRectGreyDisp,
LeftImageBlock;
	cv::Rect2d rL, rR;
	cv::Size img_size;
	img_size = RightImageRect.size();

	// Create the result matrix
	int result_colsDS = RightImageRect.cols - boxWidthDS + 1;
	int result_rowsDS = RightImageRect.rows - boxHeightDS + 1;
	resultDS.create(result_rowsDS, result_colsDS, CV_32FC1);
	cv::cvtColor(LeftImageRect, LeftImageRectGrey, cv::COLOR_BGR2GRAY);
	cv::cvtColor(RightImageRect, RightImageRectGrey, cv::COLOR_BGR2GRAY);
	LeftImageRectGrey.copyTo(LeftImageRectGreyDisp);
	cv::Mat RightImageRectIDS;
	cv::Mat disps;
	double          minValDS, maxValDS;
	cv::Point       minLocDS, maxLocDS, matchLocDS;

	// crop the block from the left image
	rL = cv::Rect2d((TopLeftPoint.x), (TopLeftPoint.y), boxWidthDS, boxHeightDS);
	LeftImageBlock = LeftImageRect(rL);


	// crop the right image so that search is one dimensional
	rR = cv::Rect2d(0, (TopLeftPoint.y), img_size.width, boxHeightDS);
	RightImageRectIDS = RightImageRect(rR);


	/// Do the Matching and Normalize
	cv::matchTemplate(RightImageRectIDS, LeftImageBlock, resultDS, cv::TM_CCORR_NORMED);
	std::cout << 5 << std::endl;
	cv::normalize(resultDS, resultDS, 0, 1, cv::NORM_MINMAX, -1, cv::Mat());
	std::cout << 6 << std::endl;
	minMaxLoc(resultDS, &minValDS, &maxValDS, &minLocDS, &maxLocDS, cv::Mat());
	matchLocDS = maxLocDS;
	std::cout << 7 << std::endl;
	// Store the value
	disps.push_back((matchLocDS.x) - (TopLeftPoint.x));

	std::cout << "3" << std::endl;
	minMaxLoc(disps, &minValDS, &maxValDS, &minLocDS, &maxLocDS, cv::Mat());
	int dispCentre = ((maxValDS + minValDS) / 2);

	std::cout << "Done, min is: " << minValDS << ", max is: " << maxValDS << ", disparity
search centre is: " << dispCentre << std::endl;

	rR.x = rR.x + matchLocDS.x;
	rR.y = rR.y + matchLocDS.y;
	rR.width = boxWidthDS;
	/*
	cv::Mat newImage3;
	cv::rectangle(LeftImageRectGrey, rL, cv::Scalar(255, 255, 255), 3);
	cv::rectangle(RightImageRectGrey, rR, cv::Scalar(255, 255, 255), 3);
	cv::hconcat(LeftImageRectGrey, RightImageRectGrey, newImage3);
	cv::namedWindow("Search Visualisation2", cv::WINDOW_NORMAL);
	imshow("Search Visualisation2", newImage3);
	cv::waitKey();
	*/
```

```
	return dispCentre;
}


void computeSparsePointCloud(cv::Mat LeftImageRect, cv::Mat RightImageRect, int noOfSquares,
cv::Rect2d ROI, int dispCentre, double correlationThreshold, cv::Mat Q)
{
	// Initialise Variables
	cv::Mat dispS2, LeftImageBlock, RightImageBlock;
	int TOPLEFTX, TOPLEFTY, TOPLEFTDX, TOPLEFTDY;
	int TLX, TLY, BW, BH;
	int centX, centY;
	cv::Mat		resultDS, resultD;
	double		minValDS, maxValDS, minValD, maxValD;
	cv::Point	minLocDS, maxLocDS, maxLocD, minLocD;
	double		minValdisp, maxValdisp;
	cv::Point	minLocdisp, maxLocdisp, matchLocDS;
	cv::Mat CorrS2;
	cv::Mat dispS2M;
	cv::Mat A(7, 3, CV_32FC1);
	Eigen::MatrixXd dispA, dispY, pinv;
	double aS, bS, cS, sampleDisp, SPdispX, SPdispY;
	cv::Mat SM, sampleSx, sampleSy, sampleS;
	std::vector<double> X;
	std::vector<double> Y;
	std::vector<double> Z;
	std::vector<std::vector<double> > P;
	cv::Mat LeftImageRectGrey, RightImageRectGrey;
	cv::Rect2d rL, rR;
	cv::Mat RightImageRectGreyDispCopy, LeftImageRectGreyDispCopy, newImage3;
	cv::Rect2d rRvisualisation;
	std::vector<cv::Point> CentrePoints;
	std::vector<double> DisparityVector, DisparityVectorTemp;
	cv::Mat newImage5;

	// Convert to grayscale
	cv::cvtColor(LeftImageRect, LeftImageRectGrey, cv::COLOR_BGR2GRAY);
	cv::cvtColor(RightImageRect, RightImageRectGrey, cv::COLOR_BGR2GRAY);

	// Display Images
	cv::hconcat(LeftImageRectGrey, RightImageRectGrey, newImage5);
	cv::namedWindow("Search Visualisation2", cv::WINDOW_NORMAL);
	imshow("Search Visualisation2", newImage5);
	cv::waitKey();

	// Get Top Left Point and Spacing Between Points in Grid
	cv::Point TopLeftPoint = cv::Point(ROI.x, ROI.y);
	cv::Point BottomRightPoint = cv::Point(((ROI.x + ROI.width)), ((ROI.y +
ROI.height)));
	int boxWidth = (ROI.width) / noOfSquares;
	int boxHeight = (ROI.height) / noOfSquares;

	// Create Window to Show Search Process
	if (showProcess == 1)
	{
		cv::namedWindow("Search VisualisationL", cv::WINDOW_NORMAL);
		cv::namedWindow("Search VisualisationR", cv::WINDOW_NORMAL);
		LeftImageRectGrey.copyTo(LeftImageRectGreyDispCopy);

		cv::namedWindow("Search Visualisation", cv::WINDOW_NORMAL);
	}

	// Create List of Points (in the grid)
	for (int k = 0; k < (noOfSquares); k++) { // shifts in the vertical direction

		for (int j = 0; j < (noOfSquares); j++) { // shifts in the horizontal
direction

			TOPLEFTX = (TopLeftPoint.x + (j * boxWidth));
			TOPLEFTY = (TopLeftPoint.y + (k * boxHeight));
```

```
                        centX = TOPLEFTX + (boxWidth / 2);
                        centY = TOPLEFTY + (boxHeight / 2);

                        CentrePoints.push_back(cv::Point(centX, centY));
                        DisparityVector.push_back(dispCentre);
                        DisparityVectorTemp.push_back(dispCentre);


                        //cv::circle(LeftImageRectGreyDispCopy, cv::Point(centX, centY), 2,
cv::Scalar(255, 255, 255), 5);  // DEBUG

                }
        }
        //RightImageRectGrey.copyTo(RightImageRectGreyDispCopy);
        //cv::hconcat(LeftImageRectGreyDispCopy, RightImageRectGreyDispCopy, newImage3);
        // DEBUG
        //imshow("Search Visualisation", newImage3);
        //cv::waitKey();


        // Initilise more parameters
        for (int disps2 = 0; disps2 < 7; disps2++) {
                dispS2.push_back(disps2);
        }
        int BWR, BHR;
        double ConsistencyDiff, Diff;
        int s;
        cv::Mat TextureTest;
        cv::Scalar mean, stddev;
        cv::Size PatchSize, SearchBlockSize;
        cv::Mat RightBlockSubPix;
        cv::Mat Patch, SearchBlock;


        // Open CSV file for saving sparse disparity map data
        std::ofstream myfile;
        if (ReconMode == 1)
        {
                myfile.open("Data/sparseCloudData.csv");
                int check = 1;
        }

        // STEREO MATCHING LOOP
        std::cout << "Computing topography using " << CentrePoints.size() << " points" <<
std::endl;
        for (int i = 0; i < (CentrePoints.size()); i++) { // Loop Over Points

                // First Patch Size
                BW = boxWidth * 6;
                BH = boxHeight * 6;

                for (s = 1; s < (numlvls + 1); s++) { // Loop Over Support Window Size

                        if (ReconMode == 1)
                        {
                                ////////////////////////////// FORWARD PASS
                                // Get Left Patch
                                rL = cv::Rect2d((CentrePoints[i].x - (BW / 2)),
(CentrePoints[i].y) - (BH / 2), BW, BH);
                                LeftImageBlock = LeftImageRectGrey(rL);

                                // Apply Texture Threshold
                                LeftImageBlock.copyTo(TextureTest);
                                cv::normalize(TextureTest, TextureTest, 0, 1,
cv::NORM_MINMAX);
                                meanStdDev(TextureTest, mean, stddev);
                                if (stddev(0) < TEXTTHRESH)
                                {
                                        break;
                                }
```

```
                                // Get Search Patch in Right Image (Sub-pixel)
                                SearchBlockSize.height = BH;
                                SearchBlockSize.width = 1.5 * BW;
                                cv::getRectSubPix(RightImageRectGrey, SearchBlockSize,
cv::Point2f((CentrePoints[i].x + DisparityVector[i]), (CentrePoints[i].y)), SearchBlock);

                                if (showProcess == 1)
                                {
                                        LeftImageRectGrey.copyTo(LeftImageRectGreyDispCopy);
                                        cv::rectangle(LeftImageRectGreyDispCopy, rL,
cv::Scalar(255, 255, 255), 5);

        RightImageRectGrey.copyTo(RightImageRectGreyDispCopy);
                                        cv::rectangle(RightImageRectGreyDispCopy, rR,
cv::Scalar(255, 0, 0), 5);
                                        cv::hconcat(LeftImageRectGreyDispCopy,
RightImageRectGreyDispCopy, newImage3);
                                        imshow("Search Visualisation", newImage3);
                                        imshow("Search VisualisationL", LeftImageBlock);
                                        imshow("Search VisualisationR", RightImageBlock);
                                        cv::waitKey();
                                }

                                // Perform Matching for sub pixel disparity
                                cv::matchTemplate(LeftImageBlock, SearchBlock, resultD,
cv::TM_CCORR_NORMED);
                                minMaxLoc(resultD, &minValD, &maxValD, &minLocD, &maxLocD,
cv::Mat());
                                // Get the correlation in the range -3 to 3 over the point
of highest disparity
                                for (int cor = 0; cor < (7); cor++) { // s
                                        CorrS2.push_back(resultD.at<float>(0, maxLocD.x - 3
+ cor));
                                }
                                // Fit quadratic to data and find sub pixel disparity
corresponding to highest correlation
                                A.setTo(1);
                                dispS2M = dispS2.mul(dispS2);
                                dispS2M.copyTo(A.col(0));
                                dispS2.copyTo(A.col(1));
                                cv::cv2eigen(A, dispA);
                                cv::cv2eigen(CorrS2, dispY);
                                pinv =
(dispA.completeOrthogonalDecomposition().pseudoInverse());
                                pinv = pinv * dispY;
                                cv::eigen2cv(pinv, SM);
                                aS = SM.at<double>(0, 0);
                                bS = SM.at<double>(0, 1);
                                cS = SM.at<double>(0, 2);
                                SPdispX = (DisparityVector[i] - (SearchBlockSize.width / 2))
+ (maxLocD.x - 3) - ((bS) / (2 * (aS))) + (BW / 2); // Sub Pixel Disparity
                                SPdispY = (((-((bS) / (2 * (aS)))) * (-((bS) / (2 * (aS))))
* aS) + ((-((bS) / (2 * (aS)))) * bS) + cS); // Corresponding Correlation
                                DisparityVectorTemp[i] = SPdispX;



                                //////////////////////////// BACKWARD PASS
                                CorrS2.release();
                                cv::Mat CorrS2;

                                // Get Right Patch
                                PatchSize.height = BH;
                                PatchSize.width = BW;
                                cv::getRectSubPix(RightImageRectGrey, PatchSize,
cv::Point2f((CentrePoints[i].x + DisparityVectorTemp[i]), (CentrePoints[i].y)),
RightBlockSubPix);

                                // Get Search Block in Left Image
                                SearchBlockSize.height = BH;
```

```
                                SearchBlockSize.width = 1.5 * BW;
                                cv::getRectSubPix(LeftImageRectGrey, SearchBlockSize,
cv::Point2f((CentrePoints[i].x), (CentrePoints[i].y)), SearchBlock);

                                if (showProcess == 1)
                                {
                                        cv::rectangle(RightImageRectGreyDispCopy, rR,
cv::Scalar(255, 0, 0), 5);
                                        cv::hconcat(LeftImageRectGreyDispCopy,
RightImageRectGreyDispCopy, newImage3);
                                        imshow("Search Visualisation", newImage3);
                                        imshow("Search VisualisationL", LeftImageBlock);
                                        imshow("Search VisualisationR", RightImageBlock);
                                }

                                ////////// Perform Matching for sub pixel disparity
                                cv::matchTemplate(RightBlockSubPix, SearchBlock, resultD,
cv::TM_CCORR_NORMED);
                                minMaxLoc(resultD, &minValD, &maxValD, &minLocD, &maxLocD,
cv::Mat());
                                // Get the correlation in the range -3 to 3 over the point
of highest disparity
                                for (int cor = 0; cor < (7); cor++) { // s
                                        CorrS2.push_back(resultD.at<float>(0, maxLocD.x - 3
+ cor));
                                }
                                // Fit quadratic to data and find sub pixel disparity
corresponding to highest correlation
                                A.setTo(1);
                                dispS2M = dispS2.mul(dispS2);
                                dispS2M.copyTo(A.col(0));
                                dispS2.copyTo(A.col(1));
                                cv::cv2eigen(A, dispA);
                                cv::cv2eigen(CorrS2, dispY);
                                pinv =
(dispA.completeOrthogonalDecomposition().pseudoInverse());
                                pinv = pinv * dispY;
                                cv::eigen2cv(pinv, SM);
                                aS = SM.at<double>(0, 0);
                                bS = SM.at<double>(0, 1);
                                cS = SM.at<double>(0, 2);
                                SPdispY = (((-((bS) / (2 * (aS)))) * (-((bS) / (2 * (aS))))
* aS) + ((-((bS) / (2 * (aS)))) * bS) + cS); // Corresponding Correlation
                                SPdispX = ((DisparityVectorTemp[i]) - ((-
SearchBlockSize.width / 2) + (((maxLocD.x - 3) - ((bS) / (2 * (aS))))) + (PatchSize.width /
2)));


                                ////////////////////////// L-R Consistency Check
                                Diff = SPdispX - DisparityVectorTemp[i];
                                ConsistencyDiff = abs(Diff);
                                if (ConsistencyDiff < ConsistTHRESH)
                                {
                                        DisparityVector[i] = (((DisparityVectorTemp[i]) +
(SPdispX)) / 2);
                                }
                                else
                                {
                                        break;
                                }

                                ////////////////////////// Decrease Support Window Size
                                BW = BW / 1.2;
                                BH = BH / 1.2;



                                ////////////////////////// LEVEL THRESHOLD CHECK
                                if (s > minlvl)
                                {
                                        X.push_back(CentrePoints[i].x);
```

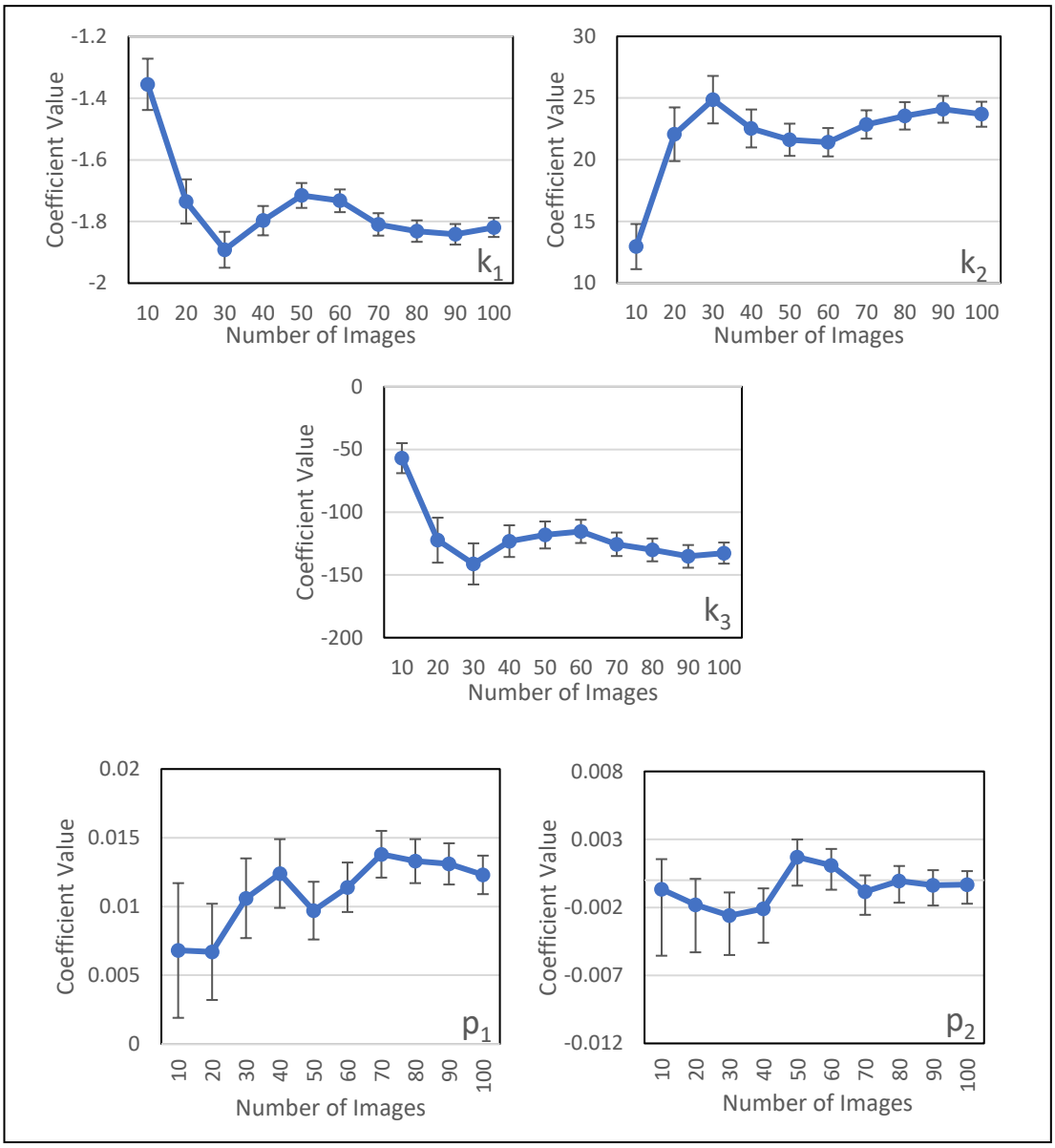
```
                                        Y.push_back(CentrePoints[i].y);
                                        Z.push_back(DisparityVector[i]);
                                        myfile << CentrePoints[i].x << "," <<
CentrePoints[i].y << "," << DisparityVector[i] << ",\n";
                                }
                                else
                                {
                                        myfile << CentrePoints[i].x << "," <<
CentrePoints[i].y << "," << ",\n";
                                }

                                std::cout << " " << i << '\r';
                        }
                }
        }
        // Close csv file
        if (ReconMode == 1)
        {
                myfile.close();
        }

        // Open csv file containing interpolated disparity map
        if (ReconMode == 0)
        {
                std::ifstream myFile("Data/interpolatedDisparityMap.csv");
                std::string line;
                double val;

                while (std::getline(myFile, line))
                {
                        // Create a stringstream of the current line
                        std::stringstream ss(line);

                        // Keep track of the current column index
                        int colIdx = 0;

                        // Extract each integer
                        while (ss >> val) {

                                // Add the current integer to the 'colIdx' column's values
vector

                                if (colIdx == 0)
                                {

                                        X.push_back(val);
                                        //std::cout << "X: " << val << std::endl;
                                }

                                if (colIdx == 1)
                                {


                                        Y.push_back(val);
                                        //std::cout << "Y: " << val << std::endl;

                                }

                                if (colIdx == 2)
                                {

                                        Z.push_back(-val);
                                        //std::cout << "Z: " << val << std::endl;

                                }

                                // If the next token is a comma, ignore it and move on
                                if (ss.peek() == ',') ss.ignore();

                                // Increment the column index
                                colIdx++;
```

```
                        }
                }

                // Close file
                myFile.close();
        }

        // Place XYZ data in vector P
        P.push_back(X);
        P.push_back(Y);
        P.push_back(Z);


        // Reproject points to 3D space
        cv::Vec3b vColTemp;
        cv::Mat_<float> tmpvec(4, 1);
        pcl::PointXYZ pcpoint;
        pcl::PointXYZRGB pcpointCol;
        for (int z = 0; z < (X.size()); z++) {

                tmpvec(0) = P[0].at(z);
                tmpvec(1) = P[1].at(z);
                tmpvec(2) = P[2].at(z);
                tmpvec(3) = 1;

                vColTemp = LeftImageRect.at<cv::Vec3b>(P[1].at(z), P[0].at(z));


                tmpvec = Q * tmpvec;
                tmpvec /= tmpvec(3);


                pcpointCol.x = tmpvec(0);
                pcpointCol.y = tmpvec(1);
                pcpointCol.z = tmpvec(2);

                pcpointCol.r = vColTemp[2];
                pcpointCol.g = vColTemp[1];
                pcpointCol.b = vColTemp[0];


                denseTexturedCloud->push_back(pcpointCol);

        }

        // Save Point Cloud
        pcl::io::savePLYFileBinary("Data/DensePointCloud.ply", *denseTexturedCloud);
        std::cout << "Final point cloud saved                                    " << '\r';
        cv::waitKey();

}


void CloudVisualisation()
{
        // Setup the visualiser
        boost::shared_ptr<pcl::visualization::PCLVisualizer> viewer(new
pcl::visualization::PCLVisualizer("3D Viewer"));
        viewer->setBackgroundColor(0, 0, 0);

        viewer->addPointCloud<pcl::PointXYZRGB>(denseTexturedCloud, "Final_PC_Overlay");


        while (!viewer->wasStopped())
        {

                viewer->spinOnce(100);
                boost::this_thread::sleep(boost::posix_time::microseconds(100000));

        }
```

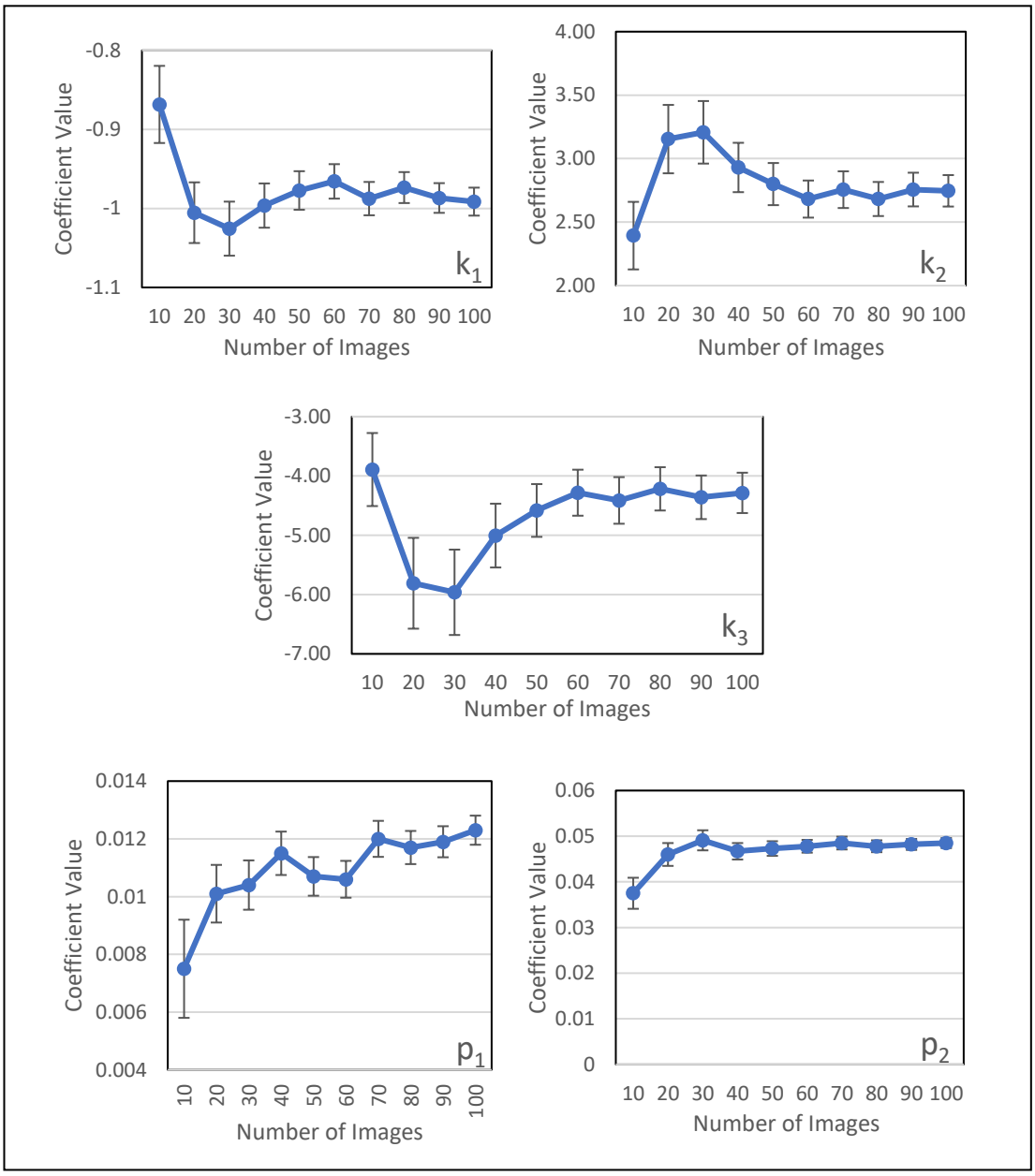
```
}


cv::Rect2d ROISelection(cv::Mat LeftImageRect)
{
        std::cout << "Select the region of interest for analysis" << '\r';
        cv::Rect2d ROI;
        cv::Mat LeftImageRectCopyResized;
        cv::resize(LeftImageRect, LeftImageRectCopyResized, cv::Size((int(LeftImageRect.cols
/ 4)), (int(LeftImageRect.rows / 4))));
        ROI = cv::selectROI(LeftImageRectCopyResized);
        cv::destroyWindow("ROI selector");
        ROI.x = 4 * ROI.x;
        ROI.y = 4 * ROI.y;
        ROI.width = 4 * ROI.width;
        ROI.height = 4 * ROI.height;

        std::cout << "Region of interest selected
" << std::endl;
        return ROI;
}
```

APPENDIX 6C – GRAPHS SHOWING CALIBRATION COEFFICIENTS AND PARAMETERS AGAINST THE NUMBER OF CALIBRATION IMAGES USED

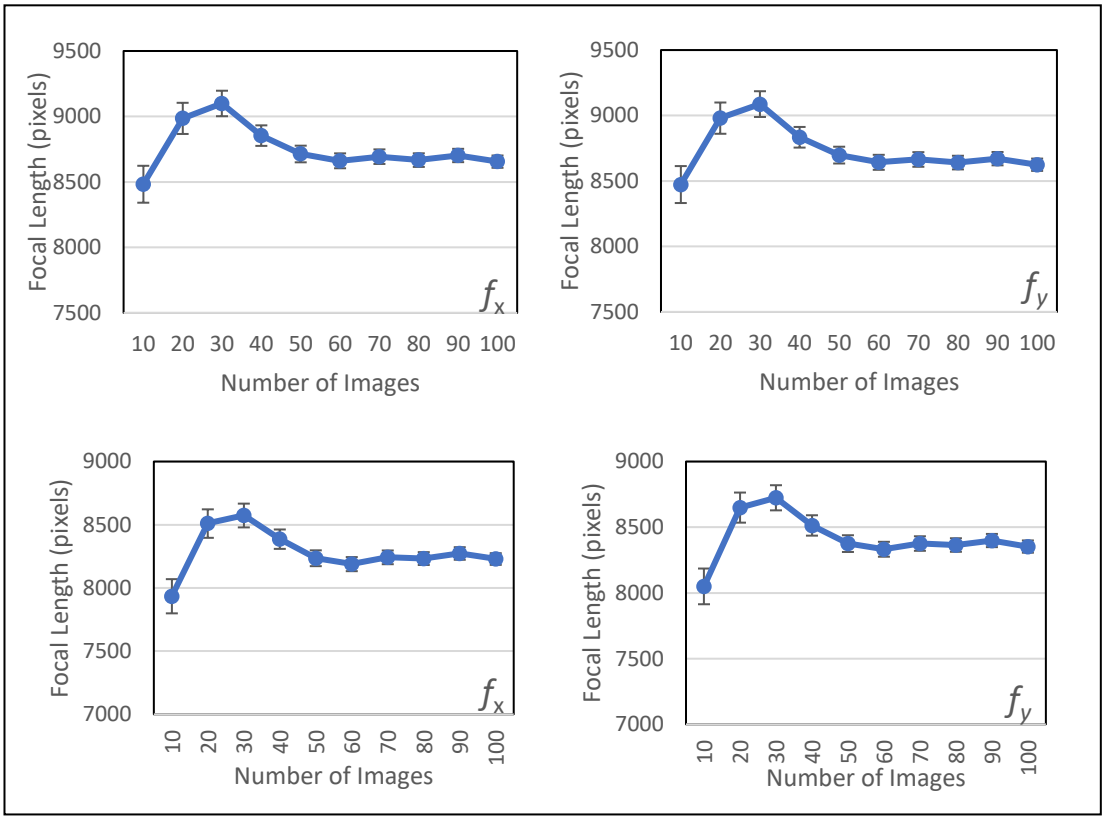
Left Camera Distortion Coefficients

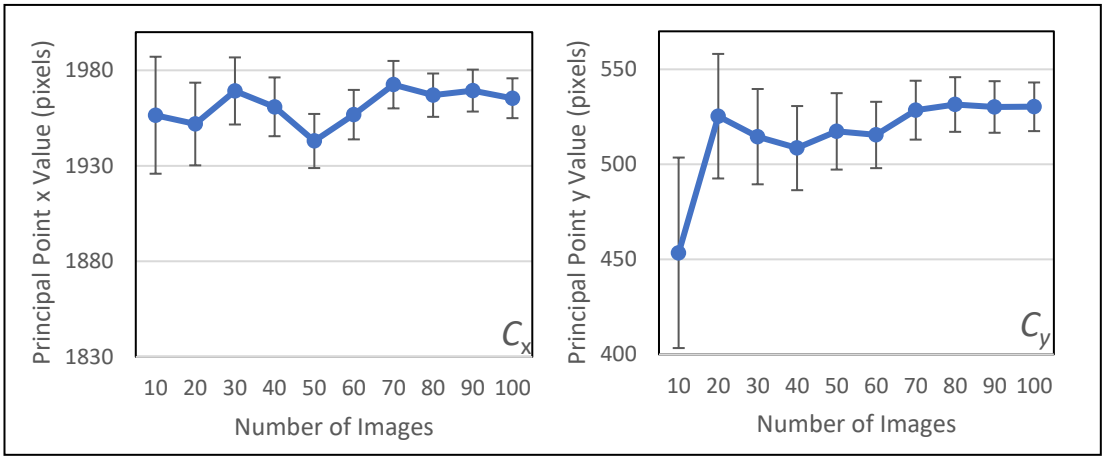


Right Camera Distortion Coefficients

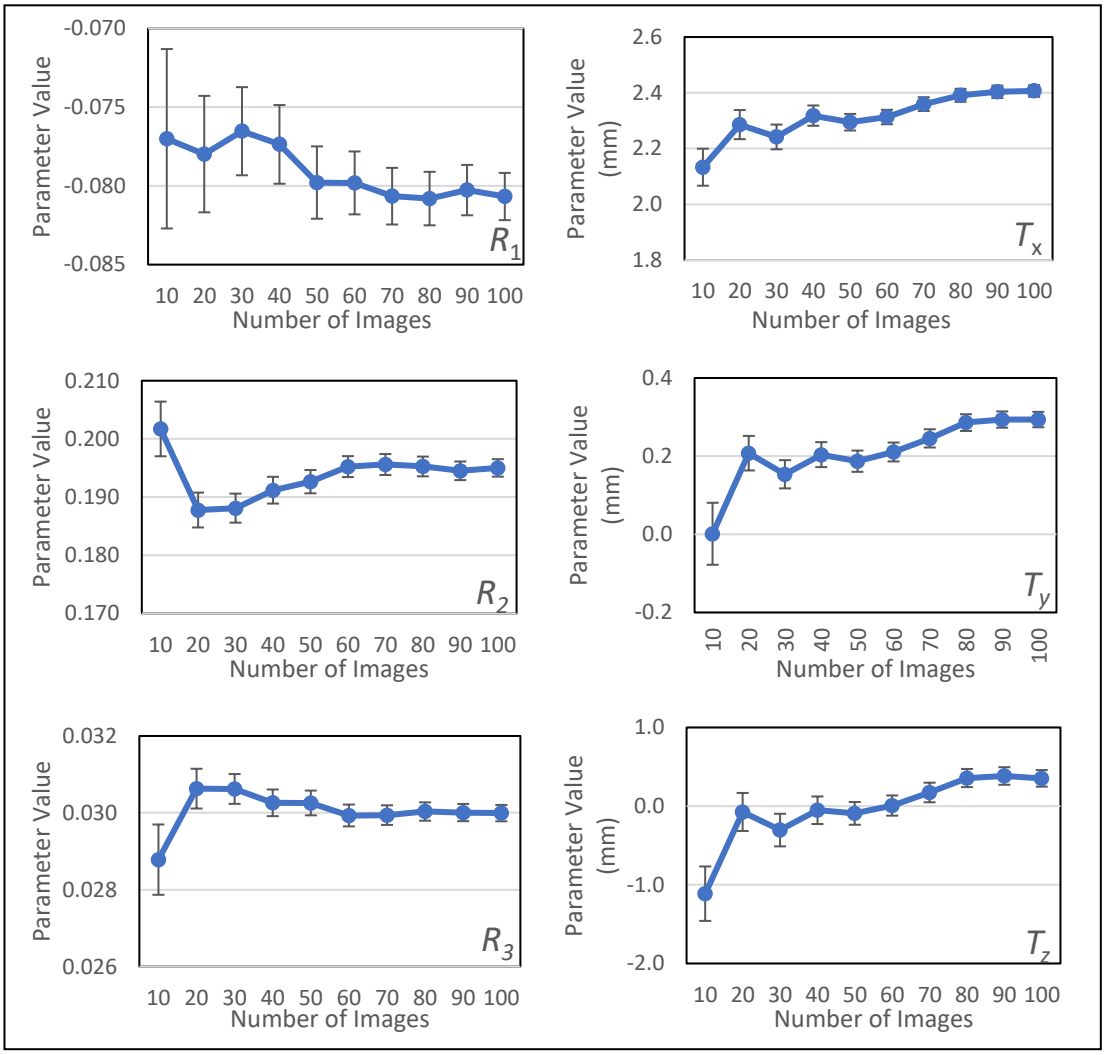


Intrinsic Parameters





Extrinsic Parameters



APPENDIX 7A – STEREO VISION PIPELINE WRITTEN IN C++ FOR THE WORK OF CHAPTER 7

```
////////////////////////////////////////////////////////////////////////////////
//////////////                3D Reconstruction Code                //////////////
////////////////////////////////////////////////////////////////////////////////

/*
Author: Ian Coghill(1) (EngD Student)
Supervised by Dr Mario E Giardini(1), Dr Richard Black(1) & Dr Iain AT Livingstone(2)
1 - Department of Biomedical Engineering, University of Strathclyde
2 - NHS Forth Valley
*/

//////////////////////////////////   INCLUDE LIBRARIES   //////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////
#include <iostream>                                                          // General
#include <fstream>
#include <string>
#include <conio.h>
#include <Eigen/Core>
#include <Eigen/QR>
#include <cmath>
#include <direct.h>
#include <stdlib.h>
#include <string.h>
#include "opencv2/core/core.hpp"                                             // OpenCV
#include "opencv2/core/eigen.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/calib3d/calib3d.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/imgcodecs.hpp"
#include "pcl/common/common.h"                                  // Point Cloud Library
#include "pcl/point_types.h"
#include "pcl/point_cloud.h"
#include "pcl/visualization/pcl_visualizer.h"
#include "pcl/common/centroid.h"
#include <pcl/common/transforms.h>
#include <pcl/surface/poisson.h>
#include <pcl/io/vtk_io.h>
#include <pcl/io/ply_io.h>
#include <pcl/features/normal_3d.h>
#include <pcl/surface/mls.h>
#include <boost/thread/thread.hpp>
#include <pcl/filters/statistical_outlier_removal.h>


//////////////////////////////   RECONSTRUCTION SETTINGS   //////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////
double  correlationThreshold = 0.80;                     // Correlation threshold
int             pxspc = 2;                                                          //
Dense reconstruction resolution paramter
int             showProcess = 0;                                    // 1 = Show stereo
matching, 0 = don't show
int             minlvl = 3;                                                         //
Coarseness Level Where Points Accepted
int             numlvls = 5;                                                        // Number
of window size levels
double  ConsistThresh = 0.9;                                  // Consitency Threshold


//////////////////////////////   POINT CLOUD OBJECTS   //////////////////////////////
////////////////////////////////////////////////////////////////////////////////////
pcl::PointCloud<pcl::PointXYZ>::Ptr cloud(new pcl::PointCloud<pcl::PointXYZ>);
pcl::PointCloud<pcl::PointXYZ>::Ptr centeredCloud(new pcl::PointCloud<pcl::PointXYZ>);
pcl::PointCloud<pcl::PointNormal>::Ptr cloudWithNormals(new
pcl::PointCloud<pcl::PointNormal>());
pcl::PointCloud<pcl::PointXYZRGB>::Ptr denseTexturedCloud(new
pcl::PointCloud<pcl::PointXYZRGB>);
pcl::PointCloud<pcl::PointNormal>::Ptr cloudUpsampledNormals(new
pcl::PointCloud<pcl::PointNormal>());
```

```
pcl::PointCloud<pcl::PointXYZRGB>::Ptr cloudTranslated0(new
pcl::PointCloud<pcl::PointXYZRGB>());
pcl::PointCloud<pcl::PointXYZ>::Ptr cloudTranslated1000(new
pcl::PointCloud<pcl::PointXYZ>());
pcl::PolygonMesh mesh;
Eigen::Affine3f tMatrix;


////////////////////////////   VISUALISATION FUNCTIONS   ////////////////////////////
/////////////////////////////////////////////////////////////////////////////////////
void sparseCloudVisualisation();
void densePointCloudVisualisation();


//////////////////////////   OTHER FUNCTIONS INITIALISATION   //////////////////////////
/////////////////////////////////////////////////////////////////////////////////////
int disparitySearchCentreEstimation(cv::Rect, cv::Mat, cv::Mat);
void computeSparsePointCloud(cv::Mat, cv::Mat, cv::Rect2d, int, double, cv::Mat,
std::vector<cv::Point2f>, int, int);
void centreCloudData();
void estimateMesh();
void reprojectImagetoMesh(cv::Rect2d, cv::Mat, int, int, cv::Mat);
cv::Rect2d ROISelection(cv::Mat);
bool NMultipleTemplateMatching(cv::Mat, cv::Mat, float, float, std::vector<cv::Point2f>&);


/////////////////////////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////   MAIN   ///////////////////////////////////////
/////////////////////////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////////////////////////
int main()
{

        std::cout << R"(

+------+.      +------+       +------+       +------+      .+------+
|`.    | `.    |\     |\      |      |      /|     /|    .' |    .'|
|  `+--+---+   | +----+-+     +------+     +-+----+ |   +---+--+'  |
|   |  |   |   | |    | |     |      |     | |    | |   |   |  |   |
+---+--+.  |   +-+----+ |     +------+     | +----+-+   |  .+--+---+
 `. |    `.|    \|     \|     |      |     |/     |/    |.'    | .'
   `+------+     +------+     +------+     +------+     +------+'

                      3D RECONSTRUCTION CODE
                                  (Feature Matching Version)

        )" << '\n';


        ////////////////////////////   VARIABLE INITIALISATION
////////////////////////////
        // Images
        cv::Mat         RGBimageL, RGBimageR;
        cv::Mat         RGBimageL_Rect, RGBimageR_Rect;
        cv::Mat         inputStereoImages, rectifiedImages, newImage3;
        // Rectification
        cv::Mat K1, D1, K2, D2;
        cv::Mat Q, R, R1, P1, R2, P2;
        cv::Vec3d       T;
        cv::Mat         map11, map12, map21, map22, img1r, img2r;
        cv::Rect        roi1, roi2;
        cv::Point       Point1, Point2;
        // Stereo Matching Variables
        int     dispCentre;


        ////////////////////////   LOAD STEREO CALIBRATION PARAMETERS
////////////////////////
        cv::FileStorage fs("Data/stereoParams.yml", cv::FileStorage::READ);
        fs["K1"] >> K1; fs["D1"] >> D1;
```

```
        fs["K2"] >> K2; fs["D2"] >> D2;
        fs["R"] >> R;
        fs["T"] >> T;


        ////////////////////////////////////   LOAD IMAGES
////////////////////////////////////
        RGBimageL = cv::imread("Data/L.png");
        RGBimageR = cv::imread("Data/R.png");
        std::cout << "Size of Left Image: " << RGBimageL.size() << std::endl;
        std::cout << "Size of Right Image: " << RGBimageR.size() << std::endl;


        ////////////////////////////////   DISPLAY INPUT IMAGES
////////////////////////////////
        cv::namedWindow("Input Stereo Image Pair", cv::WINDOW_NORMAL);
        cv::hconcat(RGBimageL, RGBimageR, inputStereoImages);
        imshow("Input Stereo Image Pair", inputStereoImages);
        cv::waitKey();


        ////////////////////////////////   RECTIFY INPUT IMAGES
////////////////////////////////
        cv::Size img_size = (RGBimageL.size());
        stereoRectify(K1, D1, K2, D2, img_size, R, T, R1, R2, P1, P2, Q,
cv::CALIB_ZERO_DISPARITY, -1, img_size, &roi1, &roi2);
        cv::initUndistortRectifyMap(K1, D1, R1, P1, img_size, CV_16SC2, map11, map12);
        cv::initUndistortRectifyMap(K2, D2, R2, P2, img_size, CV_16SC2, map21, map22);
        Q.convertTo(Q, CV_32F);
        cv::remap(RGBimageL, RGBimageL_Rect, map11, map12, cv::INTER_LINEAR);
        cv::remap(RGBimageR, RGBimageR_Rect, map21, map22, cv::INTER_LINEAR);


        ////////////////////////   DISPLAY AND SAVE RECTIFIED IMAGES
////////////////////////
        cv::namedWindow("Rectified Images", cv::WINDOW_NORMAL);
        cv::hconcat(RGBimageL_Rect, RGBimageR_Rect, rectifiedImages);
        imshow("Rectified Images", rectifiedImages);
        cv::imwrite("Data/RectifiedLeftImage.png", RGBimageL_Rect);
        cv::imwrite("Data/RectifiedRightImage.png", RGBimageR_Rect);
        cv::waitKey();


        ///////////////////////   SELECT REGION IN IMAGE TO BE ANALYSED
///////////////////////
        cv::Rect2d ROI = ROISelection(RGBimageL_Rect);


        ///////////////////////          ESTIMATE SEARCH CENTRE
///////////////////////
        dispCentre = disparitySearchCentreEstimation(ROI, RGBimageR_Rect, RGBimageL_Rect);
        std::cout << "Estimated disparity centre" << std::endl;
        cv::waitKey();


        ///////////////////////        DETERMINE POINTS TO MATCH
///////////////////////
        // Create template
        cv::Mat LeftImageTemplateSelectionRGB, LeftImageTemplateSelection,
LeftImageTemplateSelectionCropped;
        RGBimageL_Rect.copyTo(LeftImageTemplateSelectionRGB);
        std::cout << "Copied left image" << std::endl;
        cvtColor(LeftImageTemplateSelectionRGB, LeftImageTemplateSelection,
cv::COLOR_RGB2GRAY);
        std::cout << "Converted it to greyscale" << std::endl;
        cv::Rect2d ROITemplate = ROI;
        ROITemplate.height = ROI.height;
        ROITemplate.width = ROI.width;
        std::cout << "Set ROI for selecting template" << std::endl;
        LeftImageTemplateSelectionCropped = LeftImageTemplateSelection(ROITemplate);
        std::cout << "Cropped image copy with template" << std::endl;
```

```
        cv::Rect2d TemplateROIFinal = ROISelection(LeftImageTemplateSelectionCropped);
        std::cout << "Selected ROI for template" << std::endl;
        cv::Mat Img_Template_Gray = LeftImageTemplateSelectionCropped(TemplateROIFinal);
        imshow("Img_Template_Gray", Img_Template_Gray);
        std::cout << "Done getting template" << std::endl;
        cv::Mat Img_Source_Gray = LeftImageTemplateSelection (ROI);
        cv::Mat Img_Result_Bgr = LeftImageTemplateSelectionRGB (ROI);

        // Set Thresholds for Patch Similarity and Closeness to Other Patches
        float Threshold = 0.3;// How much percentage the Template should Present in the
Range: 0.0 to 1.0
        float Closeness = 0.9;// How much close Two Templates can be Range: 0-100

        // Find the Matches
        std::vector<cv::Point2f> List_Matches;
        std::vector<cv::Point2f> List_Matches_Corrected;
        std::vector<cv::Point2f> List_Matches_Centred;
        NMultipleTemplateMatching(Img_Source_Gray, Img_Template_Gray, Threshold, Closeness,
List_Matches);


        // Sort Matches Such That Points Correspond to the Patch Centres, and Display Them
        for (int i = 0; i < List_Matches.size(); i++)
        {
                rectangle(Img_Result_Bgr, List_Matches[i], cv::Point(List_Matches[i].x +
Img_Template_Gray.cols, List_Matches[i].y + Img_Template_Gray.rows), cv::Scalar(0, 0, 255),
2);
                //cv::circle(Img_Result_Bgr, cv::Point(List_Matches[i].x +
(Img_Template_Gray.cols/2), List_Matches[i].y + (Img_Template_Gray.rows/2)), 2,
cv::Scalar(255, 255, 255), 5);
                List_Matches_Corrected.push_back(cv::Point(List_Matches[i].x +
(Img_Template_Gray.cols / 2)+ROI.x, List_Matches[i].y + (Img_Template_Gray.rows / 2)+ROI.y));
        }
        cv::namedWindow("Img_Result_Bgr", cv::WINDOW_NORMAL);
        imshow("Img_Result_Bgr", Img_Result_Bgr);
        cv::waitKey();


        ////////////////////////        ESTIMATE SPARSE POINT CLOUD
////////////////////////
        int PHeight = Img_Template_Gray.rows;
        int PWidth = Img_Template_Gray.cols;
        computeSparsePointCloud(RGBimageL_Rect, RGBimageR_Rect, ROI, dispCentre,
correlationThreshold, Q, List_Matches_Corrected, PHeight, PWidth);
        centreCloudData();
        boost::thread workerThread(sparseCloudVisualisation); //Display it
        std::cout << "Sparse Cloud Estimated" << std::endl;
        pcl::io::savePLYFileBinary("Data/SparsePointCloud.ply", *centeredCloud); //Save it
        std::cout << "Sparse Point Cloud Saved" << std::endl;
        cv::waitKey();


        ////////////////////////             Fit Mesh to Cloud
////////////////////////
        // Estimate Normals
        pcl::MovingLeastSquares<pcl::PointXYZ, pcl::PointNormal> mls;
        pcl::search::KdTree<pcl::PointXYZ>::Ptr tree(new
pcl::search::KdTree<pcl::PointXYZ>());
        mls.setInputCloud(centeredCloud);
        mls.setComputeNormals(true);
        mls.setSearchMethod(tree);
        mls.setSearchRadius(2.5);
        mls.setUpsamplingMethod(pcl::MovingLeastSquares<pcl::PointXYZ,
pcl::PointNormal>::RANDOM_UNIFORM_DENSITY);
        mls.setPolynomialFit(true);
        mls.setPolynomialOrder(2);
        mls.process(*cloudUpsampledNormals);
        pcl::PointXYZ PointT;
        // Flip any abnormally oriented normals
        for (int z = 0; z < (int(centeredCloud->width)); z++)
        {
```

```
                PointT = centeredCloud->points[z];

                pcl::flipNormalTowardsViewpoint(PointT, 0, 0, -5000, cloudUpsampledNormals-
>points[z].normal_x, cloudUpsampledNormals->points[z].normal_y, cloudUpsampledNormals-
>points[z].normal_z);

        }
        pcl::concatenateFields(*centeredCloud, *cloudUpsampledNormals, *cloudWithNormals);
        // Flit the mesh
        pcl::Poisson<pcl::PointNormal> poisson;
        poisson.setInputCloud(cloudWithNormals);
        poisson.setDepth(6);
        poisson.setSolverDivide(6);
        poisson.setIsoDivide(6);
        poisson.setSamplesPerNode(4);
        poisson.reconstruct(mesh);
        pcl::io::saveVTKFile("Data/Mesh.vtk", mesh);


        ////////////////////////     Obtain Dense Textured Point Cloud
//////////////////////////
        cv::Mat TextureImage = cv::imread("Data/LT.png");
        cv::remap(TextureImage, TextureImage, map11, map12, cv::INTER_LINEAR);
        reprojectImagetoMesh(ROI, TextureImage, pxspc, -dispCentre, Q);
        boost::thread workerThread2(densePointCloudVisualisation);
        cv::waitKey();
}

int disparitySearchCentreEstimation(cv::Rect ROI, cv::Mat RightImageRect, cv::Mat
LeftImageRect)
{
        ////////////////////////////////////////// DISPARITY SEARCH CENTRE ESTIMATION
        std::cout << "Now estimating the disparity search centre" << std::endl;
        int boxWidthDS = (ROI.width);
        int boxHeightDS = (ROI.height);
        cv::Point TopLeftPoint = cv::Point(ROI.x, ROI.y);
        cv::Mat resultDS, LeftImageRectGrey, RightImageRectGrey, LeftImageRectGreyDisp,
LeftImageBlock;
        cv::Rect2d rL, rR;
        cv::Size img_size;
        img_size = RightImageRect.size();

        // Create the result matrix
        int result_colsDS = RightImageRect.cols - boxWidthDS + 1;
        int result_rowsDS = RightImageRect.rows - boxHeightDS + 1;
        resultDS.create(result_rowsDS, result_colsDS, CV_32FC1);
        cv::cvtColor(LeftImageRect, LeftImageRectGrey, cv::COLOR_BGR2GRAY);
        cv::cvtColor(RightImageRect, RightImageRectGrey, cv::COLOR_BGR2GRAY);
        LeftImageRectGrey.copyTo(LeftImageRectGreyDisp);
        cv::Mat RightImageRectIDS;
        cv::Mat disps;
        double          minValDS, maxValDS;
        cv::Point       minLocDS, maxLocDS, matchLocDS;

        // crop the block from the left image
        rL = cv::Rect2d((TopLeftPoint.x), (TopLeftPoint.y), boxWidthDS, boxHeightDS);
        LeftImageBlock = LeftImageRect(rL);


        // crop the right image so that search is one dimensional
        rR = cv::Rect2d(0, (TopLeftPoint.y), img_size.width, boxHeightDS);
        RightImageRectIDS = RightImageRect(rR);


        /// Do the Matching and Normalize
        cv::matchTemplate(RightImageRectIDS, LeftImageBlock, resultDS, cv::TM_CCORR_NORMED);
        std::cout << 5 << std::endl;
        cv::normalize(resultDS, resultDS, 0, 1, cv::NORM_MINMAX, -1, cv::Mat());
        std::cout << 6 << std::endl;
        minMaxLoc(resultDS, &minValDS, &maxValDS, &minLocDS, &maxLocDS, cv::Mat());
        matchLocDS = maxLocDS;
```

```
        std::cout << 7 << std::endl;
        // Store the value
        disps.push_back((matchLocDS.x) - (TopLeftPoint.x));

        std::cout << "3" << std::endl;
        minMaxLoc(disps, &minValDS, &maxValDS, &minLocDS, &maxLocDS, cv::Mat());
        int dispCentre = ((maxValDS + minValDS) / 2);

        std::cout << "Done, min is: " << minValDS << ", max is: " << maxValDS << ", disparity
search centre is: " << dispCentre << std::endl;

        rR.x = rR.x + matchLocDS.x;
        rR.y = rR.y + matchLocDS.y;
        rR.width = boxWidthDS;
        /*
        cv::Mat newImage3;
        cv::rectangle(LeftImageRectGrey, rL, cv::Scalar(255, 255, 255), 3);
        cv::rectangle(RightImageRectGrey, rR, cv::Scalar(255, 255, 255), 3);
        cv::hconcat(LeftImageRectGrey, RightImageRectGrey, newImage3);
        cv::namedWindow("Search Visualisation2", cv::WINDOW_NORMAL);
        imshow("Search Visualisation2", newImage3);
        cv::waitKey();
        */


        return dispCentre;
}

bool NMultipleTemplateMatching(cv::Mat Img_Input, cv::Mat Img_Template, float Threshold,
float Closeness, std::vector<cv::Point2f>& List_Matches)
{
        cv::Mat Img_Result;
        cv::Size Size_Template = Img_Template.size();
        cv::Size Size_TemplateCloseRadius((Size_Template.width / 2) * Closeness,
(Size_Template.height / 2) * Closeness);

        matchTemplate(Img_Input, Img_Template, Img_Result, cv::TM_CCOEFF_NORMED);
        threshold(Img_Result, Img_Result, Threshold, 1.0, cv::THRESH_TOZERO);
        while (true)
        {
                double minval, maxval;
                cv::Point minloc, maxloc;
                minMaxLoc(Img_Result, &minval, &maxval, &minloc, &maxloc);

                if (maxval >= Threshold)
                {
                        List_Matches.push_back(maxloc);
                        rectangle(Img_Result, cv::Point2f(maxloc.x -
Size_TemplateCloseRadius.width, maxloc.y - Size_TemplateCloseRadius.height),
cv::Point2f(maxloc.x + Size_TemplateCloseRadius.width, maxloc.y +
Size_TemplateCloseRadius.height), cv::Scalar(0), -1);
                }
                else
                        break;
        }
        //imshow("reference", Img_Debug_Bgr);
        return true;
}

void computeSparsePointCloud(cv::Mat LeftImageRect, cv::Mat RightImageRect, cv::Rect2d ROI,
int dispCentre, double correlationThreshold, cv::Mat Q, std::vector<cv::Point2f>
List_Matches_Corrected, int PHeight, int PWidth)
{
        // Initialise Variables
        cv::Mat dispS2, LeftImageBlock, RightImageBlock;
        int TOPLEFTX, TOPLEFTY, TOPLEFTDX, TOPLEFTDY;
        int TLX, TLY, BW, BH;
        int centX, centY;
        cv::Mat         resultDS, resultD;
        double          minValDS, maxValDS, minValD, maxValD;
        cv::Point       minLocDS, maxLocDS, maxLocD, minLocD;
```

```
	double		minValdisp, maxValdisp;
	cv::Point	minLocdisp, maxLocdisp, matchLocDS;
	cv::Mat CorrS2;
	cv::Mat dispS2M;
	cv::Mat A(7, 3, CV_32FC1);
	Eigen::MatrixXd dispA, dispY, pinv;
	double aS, bS, cS, sampleDisp, SPdispX, SPdispY;
	cv::Mat SM, sampleSx, sampleSy, sampleS;
	std::vector<double> X;
	std::vector<double> Y;
	std::vector<double> Z;
	std::vector<std::vector<double> > P;
	cv::Mat LeftImageRectGrey, RightImageRectGrey, MatchVisualisation;
	cv::Rect2d rL, rR;
	std::vector<double> DisparityVector, DisparityVectorTemp;

	// Convert Images to Grayscale
	cv::cvtColor(LeftImageRect, LeftImageRectGrey, cv::COLOR_BGR2GRAY);
	cv::cvtColor(RightImageRect, RightImageRectGrey, cv::COLOR_BGR2GRAY);

	// Further Initialisations
	for (int disps2 = 0; disps2 < 7; disps2++) {
		dispS2.push_back(disps2);
	}
	int BWR, BHR;
	double ConsistencyDiff, Diff;
	int s;
	cv::Mat LeftImageRectGreyDispCopy, RightImageRectGreyDispCopy;
	cv::Mat TextureTest;
	cv::Scalar mean, stddev;
	int check = 1;
	cv::Size PatchSize, SearchBlockSize;
	cv::Mat RightBlockSubPix;
	cv::Mat Patch, SearchBlock;
	BH = PHeight;
	BW = PWidth;

	// Create Vector for Storing Disparity Values
	for (int i = 0; i < (List_Matches_Corrected.size()); i++) { // Loop Over Points
	DisparityVector.push_back(dispCentre);
	DisparityVectorTemp.push_back(dispCentre);
	}

	// For if the process is to be showed
	if (showProcess == 1)
	{
		cv::namedWindow("Search Visualisation", cv::WINDOW_NORMAL);
	}



	//////////////////////		ESTIMATE SPARSE POINT CLOUD
//////////////////////
	for (int i = 0; i < (List_Matches_Corrected.size()); i++) { // Loop Over Points

		////////// FORWARD PASS //
		// Get Left Patch
		rL = cv::Rect2d((List_Matches_Corrected[i].x - (BW/2)),
(List_Matches_Corrected[i].y) - (BH/2), BW, BH);
		LeftImageBlock = LeftImageRectGrey(rL);

		// Texture Threshold
		LeftImageBlock.copyTo(TextureTest);
		cv::normalize(TextureTest, TextureTest, 0, 1, cv::NORM_MINMAX);
		meanStdDev(TextureTest, mean, stddev);
		if (stddev(0) < 0.35)
		{
			break;
		}

		// Get Search Patch in Right Image (Sub-pixel)
```

```
                SearchBlockSize.height = BH;
                SearchBlockSize.width = 1.5 * BW;
                cv::getRectSubPix(RightImageRectGrey, SearchBlockSize,
cv::Point2f((List_Matches_Corrected[i].x + DisparityVector[i]),
(List_Matches_Corrected[i].y)), SearchBlock);

                // Show Patch to be Matched
                if (showProcess == 1)
                {
                        LeftImageRectGrey.copyTo(LeftImageRectGreyDispCopy);
                        cv::rectangle(LeftImageRectGreyDispCopy, rL, cv::Scalar(255, 255,
255), 5);
                        RightImageRectGrey.copyTo(RightImageRectGreyDispCopy);
                        cv::hconcat(LeftImageRectGreyDispCopy, RightImageRectGreyDispCopy,
MatchVisualisation);
                        imshow("Search Visualisation", MatchVisualisation);
                        cv::waitKey();
                }

                //Perform Matching for sub pixel disparity
                cv::matchTemplate(LeftImageBlock, SearchBlock, resultD, cv::TM_CCORR_NORMED);
                minMaxLoc(resultD, &minValD, &maxValD, &minLocD, &maxLocD, cv::Mat());
                // Get the correlation in the range -3 to 3 over the point of highest
disparity
                for (int cor = 0; cor < (7); cor++) { // s
                        CorrS2.push_back(resultD.at<float>(0, maxLocD.x - 3 + cor));
                }
                // Fit quadratic to data and find sub pixel disparity corresponding to
highest correlation
                A.setTo(1);
                dispS2M = dispS2.mul(dispS2);
                dispS2M.copyTo(A.col(0));
                dispS2.copyTo(A.col(1));
                cv::cv2eigen(A, dispA);
                cv::cv2eigen(CorrS2, dispY);
                pinv = (dispA.completeOrthogonalDecomposition().pseudoInverse());
                pinv = pinv * dispY;
                cv::eigen2cv(pinv, SM);
                aS = SM.at<double>(0, 0);
                bS = SM.at<double>(0, 1);
                cS = SM.at<double>(0, 2);
                SPdispX = (DisparityVector[i] - (SearchBlockSize.width / 2)) + (maxLocD.x -
3) - ((bS) / (2 * (aS)))+(BW/2); // Sub Pixel Disparity
                SPdispY = (((-((bS) / (2 * (aS)))) * (-((bS) / (2 * (aS)))) * aS) + ((-((bS)
/ (2 * (aS)))) * bS) + cS); // Corresponding Correlation
                //std::cout << "Subpixadd: " << ((bS) / (2 * (aS))) << std::endl;
                DisparityVectorTemp[i] = SPdispX;


                ////////// BACKWARD PASS //
                CorrS2.release();
                cv::Mat CorrS2;

                // Get Right Patch
                PatchSize.height = BH;
                PatchSize.width = BW;
                cv::getRectSubPix(RightImageRectGrey, PatchSize,
cv::Point2f((List_Matches_Corrected[i].x + DisparityVectorTemp[i]),
(List_Matches_Corrected[i].y)), RightBlockSubPix);

                // Get Search Block in Left Image
                SearchBlockSize.height = BH;
                SearchBlockSize.width = 1.5 * BW;
                cv::getRectSubPix(LeftImageRectGrey, SearchBlockSize,
cv::Point2f((List_Matches_Corrected[i].x), (List_Matches_Corrected[i].y)), SearchBlock);
                rR = cv::Rect2d(((List_Matches_Corrected[i].x + DisparityVectorTemp[i]) - (BW
/ 2)), (List_Matches_Corrected[i].y - (BH / 2)), (BW), (BH));


                // Show Patch to be Matched
                if (showProcess == 1)
```

```
                {

                        cv::rectangle(RightImageRectGreyDispCopy, rR, cv::Scalar(255, 0, 0),
5);
                        cv::hconcat(LeftImageRectGreyDispCopy, RightImageRectGreyDispCopy,
MatchVisualisation);
                        imshow("Search Visualisation", MatchVisualisation);
                        cv::waitKey();
                }


                // Perform Matching
                cv::matchTemplate(RightBlockSubPix, SearchBlock, resultD,
cv::TM_CCORR_NORMED);
                minMaxLoc(resultD, &minValD, &maxValD, &minLocD, &maxLocD, cv::Mat());

                // Get the correlation in the range -3 to 3 over the point of highest
disparity
                for (int cor = 0; cor < (7); cor++) { // s
                        CorrS2.push_back(resultD.at<float>(0, maxLocD.x - 3 + cor));
                }

                // Fit quadratic to data and find sub pixel disparity corresponding to
highest correlation
                A.setTo(1);
                dispS2M = dispS2.mul(dispS2);
                dispS2M.copyTo(A.col(0));
                dispS2.copyTo(A.col(1));
                cv::cv2eigen(A, dispA);
                cv::cv2eigen(CorrS2, dispY);
                pinv = (dispA.completeOrthogonalDecomposition().pseudoInverse());
                pinv = pinv * dispY;
                cv::eigen2cv(pinv, SM);
                aS = SM.at<double>(0, 0);
                bS = SM.at<double>(0, 1);
                cS = SM.at<double>(0, 2);
                SPdispY = (((-((bS) / (2 * (aS)))) * (-((bS) / (2 * (aS)))) * aS) + ((-((bS)
/ (2 * (aS)))) * bS) + cS); // Corresponding Correlation
                SPdispX = ( (DisparityVectorTemp[i]) - ( (-SearchBlockSize.width / 2) +
(((maxLocD.x - 3) - ((bS) / (2 * (aS))))) + (PatchSize.width / 2) ) );

                // Apply Consistency Threshold
                Diff = SPdispX - DisparityVectorTemp[i];
                ConsistencyDiff = abs(Diff);
                if (ConsistencyDiff < ConsistThresh)
                {
                        DisparityVector[i] = (((DisparityVectorTemp[i]) + (SPdispX))/2);
                        X.push_back(List_Matches_Corrected[i].x);
                        Y.push_back(List_Matches_Corrected[i].y);
                        Z.push_back(-DisparityVector[i]);
                        std::cout << "Matched" << std::endl;
                }
}

        // Place XYZ data in vector P
        P.push_back(X);
        P.push_back(Y);
        P.push_back(Z);

        // Reproject points to 3D space
        cv::Vec3b vColTemp;
        cv::Mat_<float> tmpvec(4, 1);
        pcl::PointXYZ pcpoint;
        for (int z = 0; z < (X.size()); z++) {

                tmpvec(0) = P[0].at(z);
                tmpvec(1) = P[1].at(z);
                tmpvec(2) = P[2].at(z);
                tmpvec(3) = 1;

                vColTemp = LeftImageRect.at<cv::Vec3b>(P[1].at(z), P[0].at(z));
```

```
                tmpvec = Q * tmpvec;
                tmpvec /= tmpvec(3);


                pcpoint.x = tmpvec(0);
                pcpoint.y = tmpvec(1);
                pcpoint.z = tmpvec(2);


                cloud->push_back(pcpoint);

        }

        pcl::io::savePLYFileBinary("Data/SparsePointCloud.ply", *cloud);
        std::cout << "Final point cloud saved                                    " << '\r';
        cv::waitKey();

}

void centreCloudData()
{
        Eigen::Vector4f centroid;
        pcl::compute3DCentroid(*cloud, centroid);

        pcl::getTransformation((-centroid[0]), (-centroid[1]), (-centroid[2]), 0, 0, 0,
tMatrix);

        pcl::transformPointCloud(*cloud, *centeredCloud, tMatrix);
}

void sparseCloudVisualisation()
{
        // Setup the visualiser
        boost::shared_ptr<pcl::visualization::PCLVisualizer> viewer(new
pcl::visualization::PCLVisualizer("3D Viewer"));
        viewer->setBackgroundColor(0, 0, 0);

        viewer->addPointCloud<pcl::PointXYZ>(centeredCloud, "sample cloud");
        //viewer->addPointCloud<pcl::PointXYZRGB>(cloudTranslated0, "rays base points");
        //viewer->addPointCloud<pcl::PointXYZ>(cloudTranslated1000, "ray end points");

        while (!viewer->wasStopped())
        {

                viewer->spinOnce(100);
                boost::this_thread::sleep(boost::posix_time::microseconds(100000));

        }

}

void densePointCloudVisualisation()
{
        // Setup the visualiser
        boost::shared_ptr<pcl::visualization::PCLVisualizer> viewer(new
pcl::visualization::PCLVisualizer("3D Viewer"));
        viewer->setBackgroundColor(0, 0, 0);

        viewer->addPointCloud<pcl::PointXYZRGB>(denseTexturedCloud, "Final_PC_Overlay");

        while (!viewer->wasStopped())
        {

                viewer->spinOnce(100);
                boost::this_thread::sleep(boost::posix_time::microseconds(100000));

        }

}
```

```
void reprojectImagetoMesh(cv::Rect2d ROI, cv::Mat LeftImageRect, int pxspc, int dispCentre,
cv::Mat Q)
{

	// mesh intersection algorithm
	std::vector<int> cX0;
	std::vector<int> cY0;
	std::vector<int> cZ0;
	std::vector<int> cR0;
	std::vector<int> cG0;
	std::vector<int> cB0;
	int cCPx0;
	int cCPy0;
	std::vector<int> cX1000;
	std::vector<int> cY1000;
	std::vector<int> cZ1000;
	std::vector<std::vector<int> > cP0;
	std::vector<std::vector<int> > cP1000;
	cv::Vec3b vColTemp;

	Eigen::Vector3f vP1;
	Eigen::Vector3f vP2;
	Eigen::Vector3f vP3;
	Eigen::Vector3f vR1;
	Eigen::Vector3f vR2;
	Eigen::Vector3f vPIP;
	float vXtemp, vYtemp, vZtemp, vRtemp, vGtemp, vBtemp;
	float vR1x, vR1y, vR1z, vR2x, vR2y, vR2z;
	float vP1x, vP1y, vP1z, vP2x, vP2y, vP2z, vP3x, vP3y, vP3z;
	Eigen::Vector3f vNormal, vIntersectPos;

	cv::Point TopLeftPoint = cv::Point(ROI.x, ROI.y);
	//Create two sets of points above and below mesh but with only different disparity
	for (int Ypixel = 1; Ypixel < ((ROI.height) / pxspc); Ypixel++) {
		for (int Xpixel = 1; Xpixel < ((ROI.width) / pxspc); Xpixel++) {

			cCPx0 = (TopLeftPoint.x + (Xpixel * pxspc)); // get x and y of each
pixel in selected area
			cCPy0 = (TopLeftPoint.y + (Ypixel * pxspc));

			cX0.push_back(cCPx0); // create the components of the dataset with
disparity 40 below mesh and RGB values
			cY0.push_back(cCPy0);
			if (dispCentre < 0) {

				cZ0.push_back((dispCentre)-200);
			}
			if (dispCentre > 0) {

				cZ0.push_back((-dispCentre)-200);
			}
			vColTemp = LeftImageRect.at<cv::Vec3b>(cCPy0, cCPx0);
			cR0.push_back(vColTemp[2]);
			cG0.push_back(vColTemp[1]);
			cB0.push_back(vColTemp[0]);

			cX1000.push_back(cCPx0); // create the components of the dataset
with disparity 40 above mesh
			cY1000.push_back(cCPy0);
			if (dispCentre < 0) {

				cZ1000.push_back((dispCentre)+200);
			}

			if (dispCentre > 0) {

				cZ1000.push_back((-dispCentre)+200);
			}
```

```
                }
        }

        cP0.push_back(cX0); // create the dataset with disparity 50 below mesh and RGB values
        cP0.push_back(cY0);
        cP0.push_back(cZ0);
        cP0.push_back(cR0);
        cP0.push_back(cG0);
        cP0.push_back(cB0);

        cP1000.push_back(cX1000); // create the dataset with disparity 50 above mesh
        cP1000.push_back(cY1000);
        cP1000.push_back(cZ1000);

        // Convert these datasets from pixels to mm
        pcl::PointCloud<pcl::PointXYZRGB>::Ptr point_cloud_ptr0(new
pcl::PointCloud<pcl::PointXYZRGB>);
        pcl::PointXYZRGB pcpoint0;

        cv::Mat_<float> tmpvec(4, 1);

        pcl::PointXYZ pcpoint;
        std::vector<double> above;
        std::vector<double> below;

        // 0 disparity dataset
        for (int z = 0; z < (cX0.size()); z++) {

                tmpvec(0) = cP0[0].at(z);
                tmpvec(1) = cP0[1].at(z);
                tmpvec(2) = cP0[2].at(z);
                tmpvec(3) = 1;

                tmpvec = Q * tmpvec;
                tmpvec /= tmpvec(3);

                pcpoint0.x = tmpvec(0);
                pcpoint0.y = tmpvec(1);
                pcpoint0.z = tmpvec(2);
                pcpoint0.r = cP0[3].at(z);
                pcpoint0.g = cP0[4].at(z);
                pcpoint0.b = cP0[5].at(z);
                below.push_back(tmpvec(0));
                below.push_back(tmpvec(1));
                below.push_back(tmpvec(2));
                below.push_back(cP0[3].at(z));
                below.push_back(cP0[4].at(z));
                below.push_back(cP0[5].at(z));


                point_cloud_ptr0->push_back(pcpoint0);
        }

        pcl::PointCloud<pcl::PointXYZ>::Ptr point_cloud_ptr1000(new
pcl::PointCloud<pcl::PointXYZ>);
        pcl::PointXYZ pcpoint1000;

        // 1000 disparity dataset
        for (int z = 0; z < (cX1000.size()); z++) {

                tmpvec(0) = cP1000[0].at(z);
                tmpvec(1) = cP1000[1].at(z);
                tmpvec(2) = cP1000[2].at(z);
                tmpvec(3) = 1;

                tmpvec = Q * tmpvec;
                tmpvec /= tmpvec(3);

                pcpoint1000.x = tmpvec(0);
                pcpoint1000.y = tmpvec(1);
                pcpoint1000.z = tmpvec(2);
```

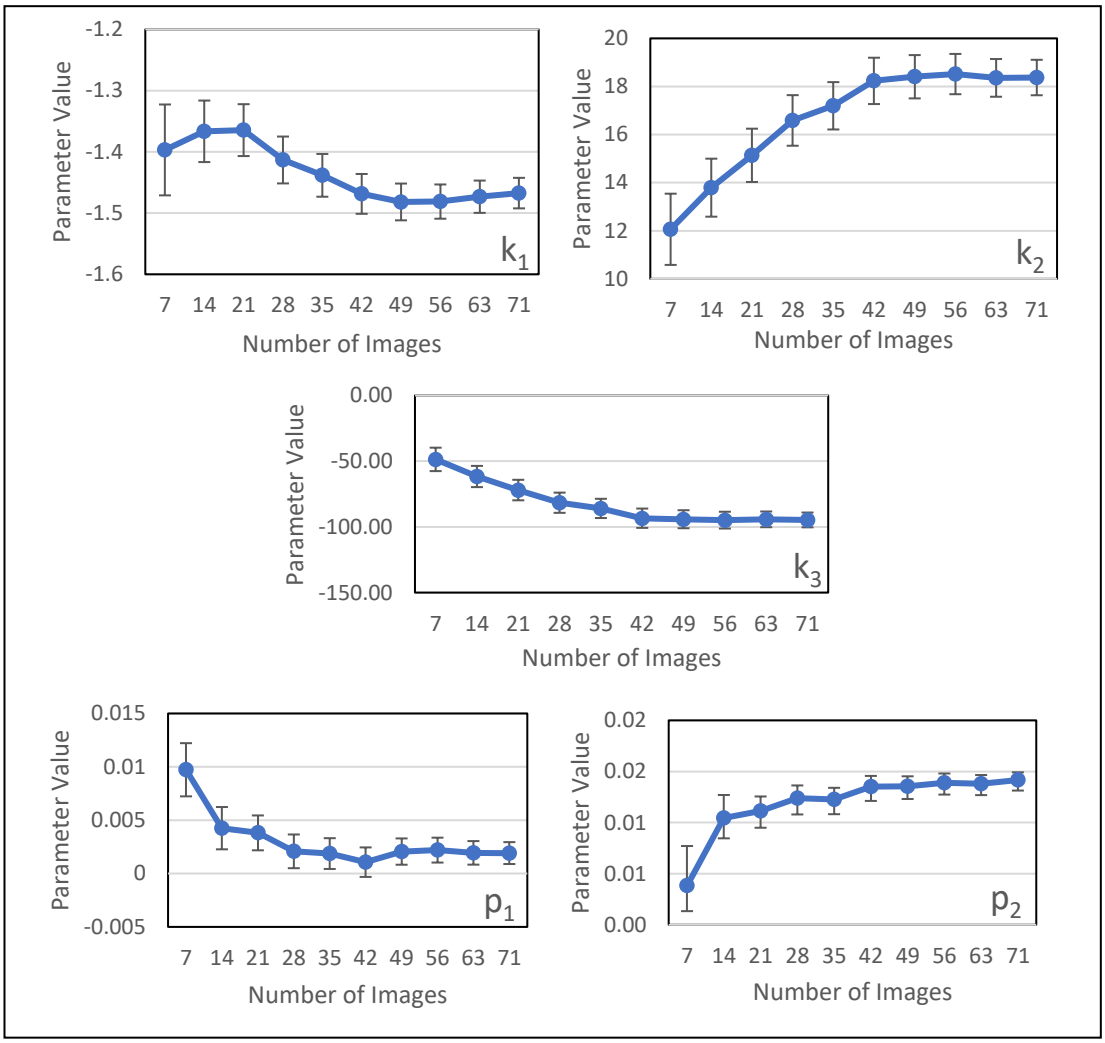
```
        above.push_back(tmpvec(0));
        above.push_back(tmpvec(1));
        above.push_back(tmpvec(2));

        point_cloud_ptr1000->push_back(pcpoint1000);
}

// translate the base and end points the same as the actual sample points
pcl::transformPointCloud(*point_cloud_ptr0, *cloudTranslated0, tMatrix);
pcl::transformPointCloud(*point_cloud_ptr1000, *cloudTranslated1000, tMatrix);

std::cout << "Created points above and below mesh" << std::endl;

// Display them in the visualizer
//std::cout << "Press any key to display the ray base points" << std::endl;
//cv::waitKey();
//viewer->addPointCloud<pcl::PointXYZRGB>(cloudTranslated0, "rays base points");
//std::cout << "Press any key to display the ray end points     " << std::endl;
//cv::waitKey();
//viewer->addPointCloud<pcl::PointXYZ>(cloudTranslated1000, "ray end points");

std::cout << "Rendering in progess...                    " << std::endl;


// try every line with every triangle to check for an intersection
pcl::PointCloud<pcl::PointXYZ>::Ptr vertices(new pcl::PointCloud<pcl::PointXYZ>);
pcl::PointXYZRGB pcpointCol;
pcl::fromPCLPointCloud2(mesh.cloud, *vertices);



for (size_t PCC = 0; PCC < (cloudTranslated0->size()); PCC++) { // try every line

        std::cout << " " << ((PCC * 100) / (cloudTranslated0->size())) << '\r';
        for (int MC = 0; MC < mesh.polygons.size(); MC++) { // try every triangle


                        // Get point cloud

                vR1x = (cloudTranslated0->at(PCC)).x;
                vR1y = (cloudTranslated0->at(PCC)).y;
                vR1z = (cloudTranslated0->at(PCC)).z;

                vR2x = (cloudTranslated1000->at(PCC)).x;
                vR2y = (cloudTranslated1000->at(PCC)).y;
                vR2z = (cloudTranslated1000->at(PCC)).z;

                vP1x = vertices->points[mesh.polygons[MC].vertices[0]].x;
                vP1y = vertices->points[mesh.polygons[MC].vertices[0]].y;
                vP1z = vertices->points[mesh.polygons[MC].vertices[0]].z;
                vP2x = vertices->points[mesh.polygons[MC].vertices[1]].x;
                vP2y = vertices->points[mesh.polygons[MC].vertices[1]].y;
                vP2z = vertices->points[mesh.polygons[MC].vertices[1]].z;
                vP3x = vertices->points[mesh.polygons[MC].vertices[2]].x;
                vP3y = vertices->points[mesh.polygons[MC].vertices[2]].y;
                vP3z = vertices->points[mesh.polygons[MC].vertices[2]].z;

                pcpointCol.r = (cloudTranslated0->at(PCC)).r;
                pcpointCol.g = (cloudTranslated0->at(PCC)).g;
                pcpointCol.b = (cloudTranslated0->at(PCC)).b;


                vR1 = Eigen::Vector3f(vR1x, vR1y, vR1z);
                vR2 = Eigen::Vector3f(vR2x, vR2y, vR2z);

                /*
                std::cout << "vR1: " << vR1 << std::endl;
                std::cout << " " << std::endl;
                std::cout << "vR2: " << vR2 << std::endl;
```

```
                        std::cout << " " << std::endl;
                        std::cout << " " << std::endl;
                        */



                        vP1 = Eigen::Vector3f(vP1x, vP1y, vP1z);
                        vP2 = Eigen::Vector3f(vP2x, vP2y, vP2z);
                        vP3 = Eigen::Vector3f(vP3x, vP3y, vP3z);
                        /*
                        std::cout << "vP1: " << vP1 << std::endl;
                        std::cout << " " << std::endl;
                        std::cout << "vP2: " << vP2 << std::endl;
                        std::cout << " " << std::endl;
                        std::cout << "vP3: " << vP3 << std::endl;
                        */

                        // Get triangle normal
                        Eigen::Vector3f u = vP2 - vP1; // u=P2-P1
                        Eigen::Vector3f v = vP3 - vP1; // v=P3-P1

                                                                                                // n=u×v
                        vNormal = u.cross(v);
                        vNormal.normalize();

                        // Find distance from LP1 and LP2 to the plane defined by the
triangle
                        float Dist1 = (vR1 - vP1).dot(vNormal);
                        float Dist2 = (vR2 - vP1).dot(vNormal);

                        //std::cout << Dist1 << std::endl;
                        //std::cout << Dist2 << std::endl;

                        if ((Dist1 * Dist2) >= 0.0f) { // Does the ray intersect the
triangle

//std::cout << "Doesn't intersect" << std::endl;
                                continue;
                        }

                        if (Dist1 == Dist2) { // Is the ray and the triangle parallel
                                                                //std::cout << "Parallel"
<< std::endl;
                                continue;
                        }

                        // Find point on the line that intersects with the plane
                        vIntersectPos = vR1 + (vR2 - vR1) * (-Dist1 / (Dist2 - Dist1));
                        //std::cout << vIntersectPos << std::endl;

                        // Find if the interesection point lies inside the triangle by
testing it against all edges
                        Eigen::Vector3f vTest;

                        vTest = vNormal.cross(vP2 - vP1);
                        if (vTest.dot(vIntersectPos - vP1) < 0.0f) {
                                continue;
                        }
                        //std::cout << "Test 1 " << vIntersectPos << std::endl;
                        //cv::waitKey();
                        vTest = vNormal.cross(vP3 - vP2);
                        if (vTest.dot(vIntersectPos - vP2) < 0.0f) {
                                continue;
                        }
                        //std::cout << "Test 2 " << vIntersectPos << std::endl;
                        //cv::waitKey();
                        vTest = vNormal.cross(vP1 - vP3);
                        if (vTest.dot(vIntersectPos - vP1) < 0.0f) {
                                continue;
                        }
```

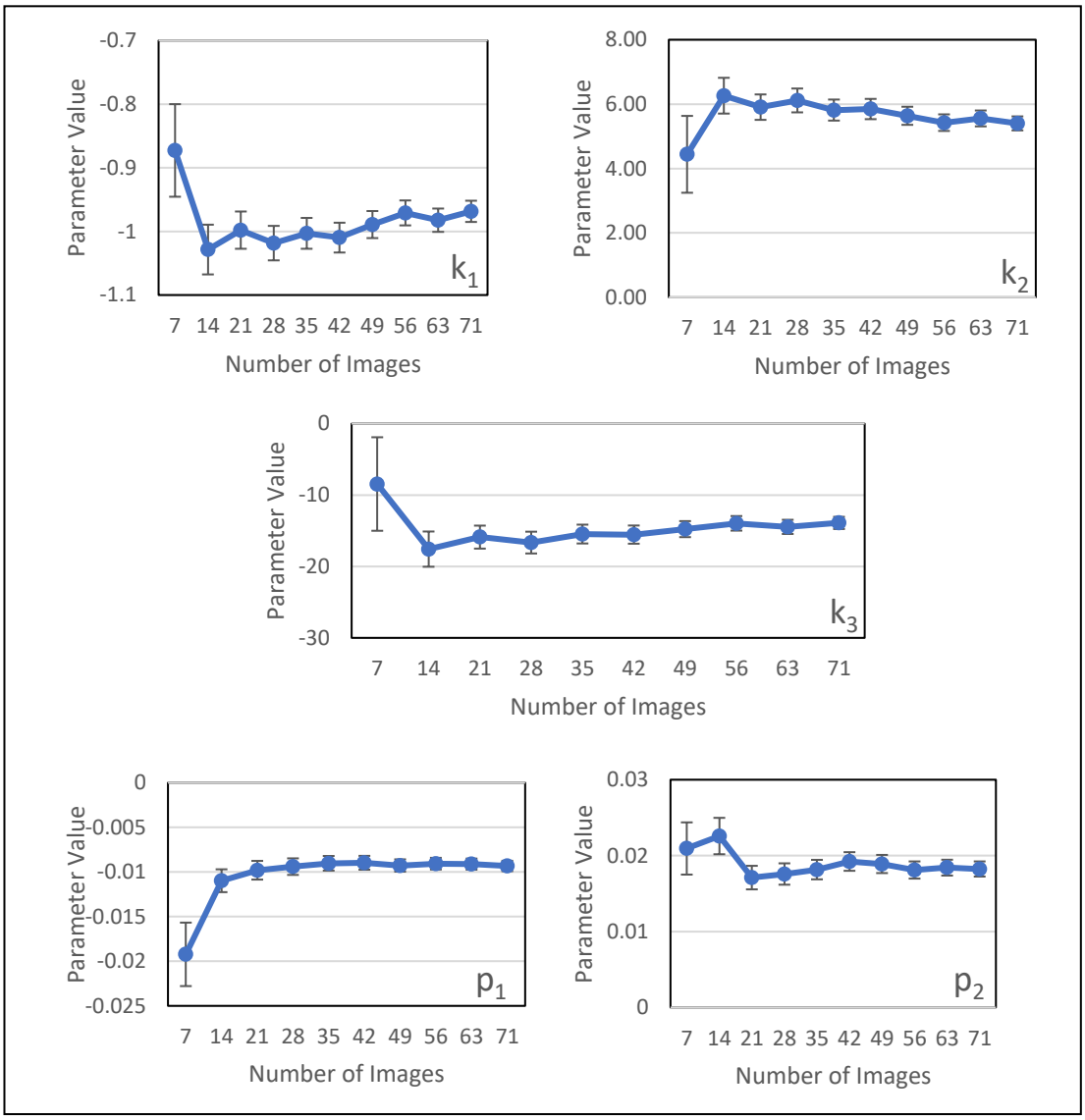
```
                        vXtemp = vIntersectPos(0);
                        vYtemp = vIntersectPos(1);
                        vZtemp = vIntersectPos(2);

                        pcpointCol.x = vXtemp;
                        pcpointCol.y = vYtemp;
                        pcpointCol.z = vZtemp;
                        pcpointCol.r = (cloudTranslated0->at(PCC)).r;
                        pcpointCol.g = (cloudTranslated0->at(PCC)).g;
                        pcpointCol.b = (cloudTranslated0->at(PCC)).b;

                        denseTexturedCloud->push_back(pcpointCol);

                }


        }

        std::cout << "Rendering complete                                    " << std::endl;



        pcl::io::savePLYFileBinary("Data/DensePointCloud.ply", *denseTexturedCloud);
        std::cout << "Final point cloud saved                                    " << '\r';
        cv::waitKey();

}

cv::Rect2d ROISelection(cv::Mat LeftImageRect)
{
        std::cout << "Select the region of interest for analysis" << '\r';
        cv::namedWindow("ROI selector", cv::WINDOW_NORMAL);
        cv::Rect2d ROI;
        cv::Mat LeftImageRectCopyResized;
        cv::resize(LeftImageRect, LeftImageRectCopyResized, cv::Size((int(LeftImageRect.cols
/ 4)), (int(LeftImageRect.rows / 4))));
        ROI = cv::selectROI(LeftImageRectCopyResized);
        cv::destroyWindow("ROI selector");
        ROI.x = 4 * ROI.x;
        ROI.y = 4 * ROI.y;
        ROI.width = 4 * ROI.width;
        ROI.height = 4 * ROI.height;

        std::cout << "Region of interest selected
" << std::endl;
        return ROI;
}
```

APPENDIX 7B – GRAPHS SHOWING CALIBRATION COEFFICIENTS AND PARAMETERS AGAINST THE NUMBER OF CALIBRATION IMAGES USED

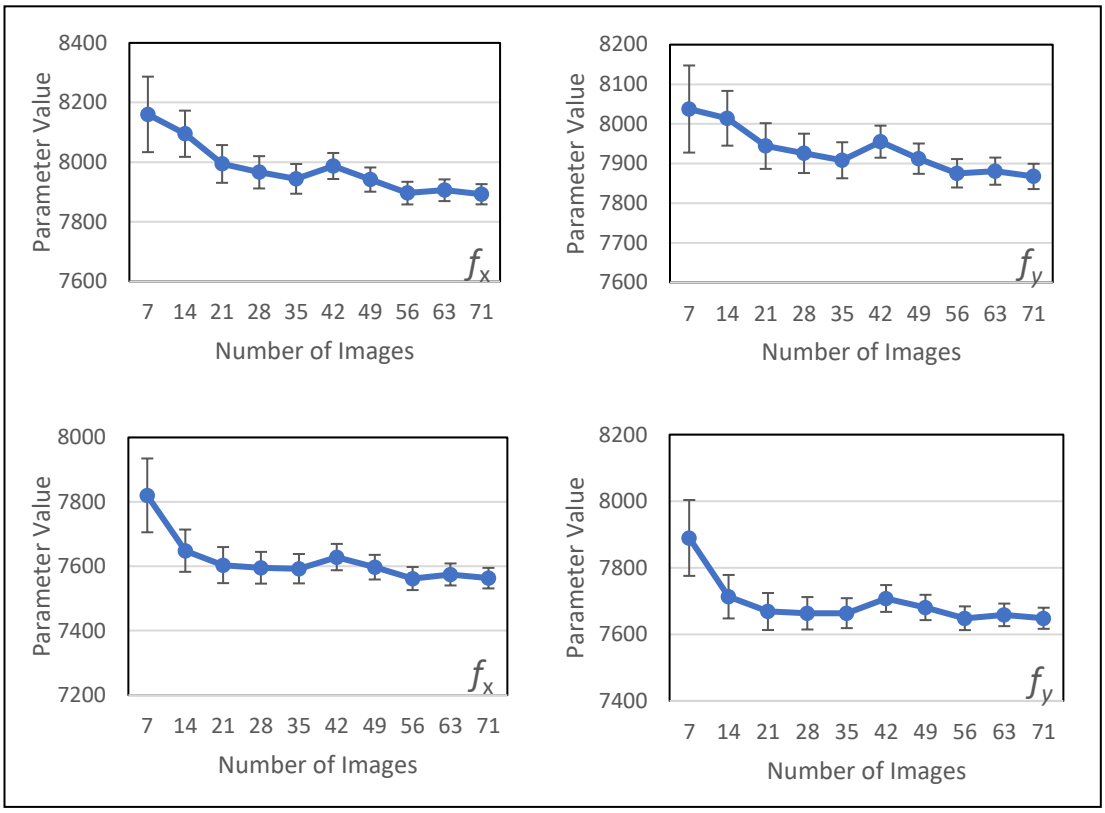
Left Camera Distortion Coefficients

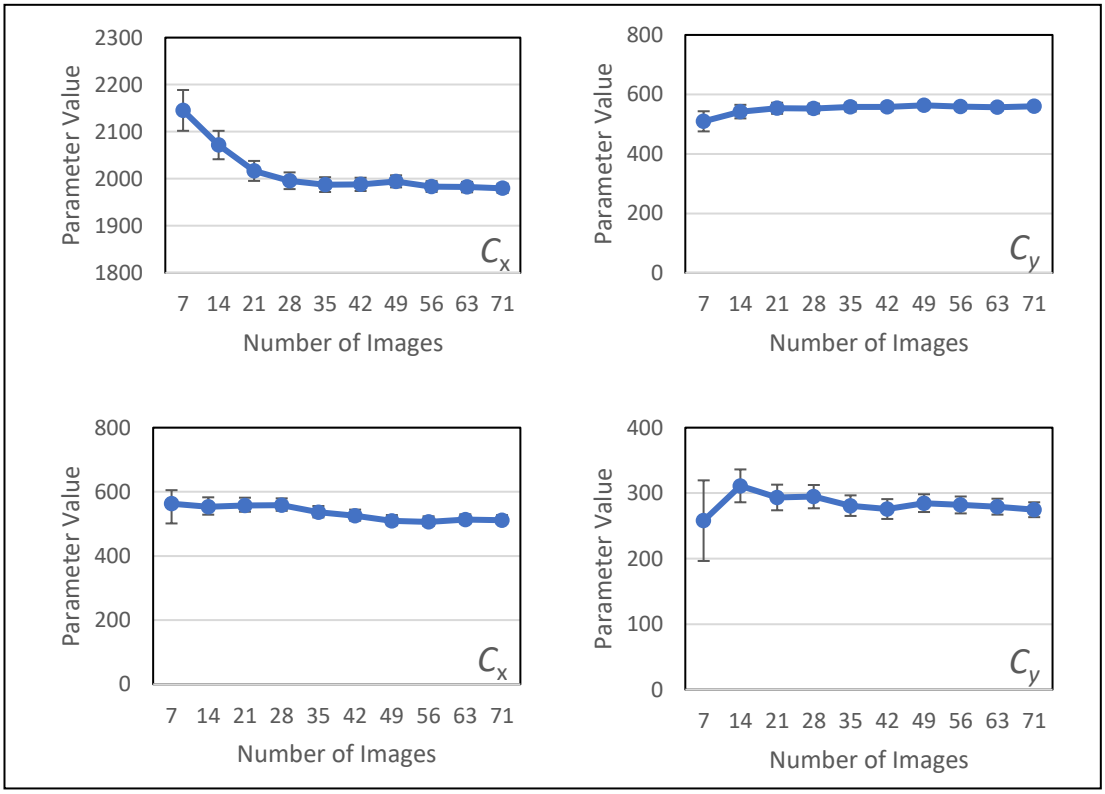


Right Camera Distortion Coefficients

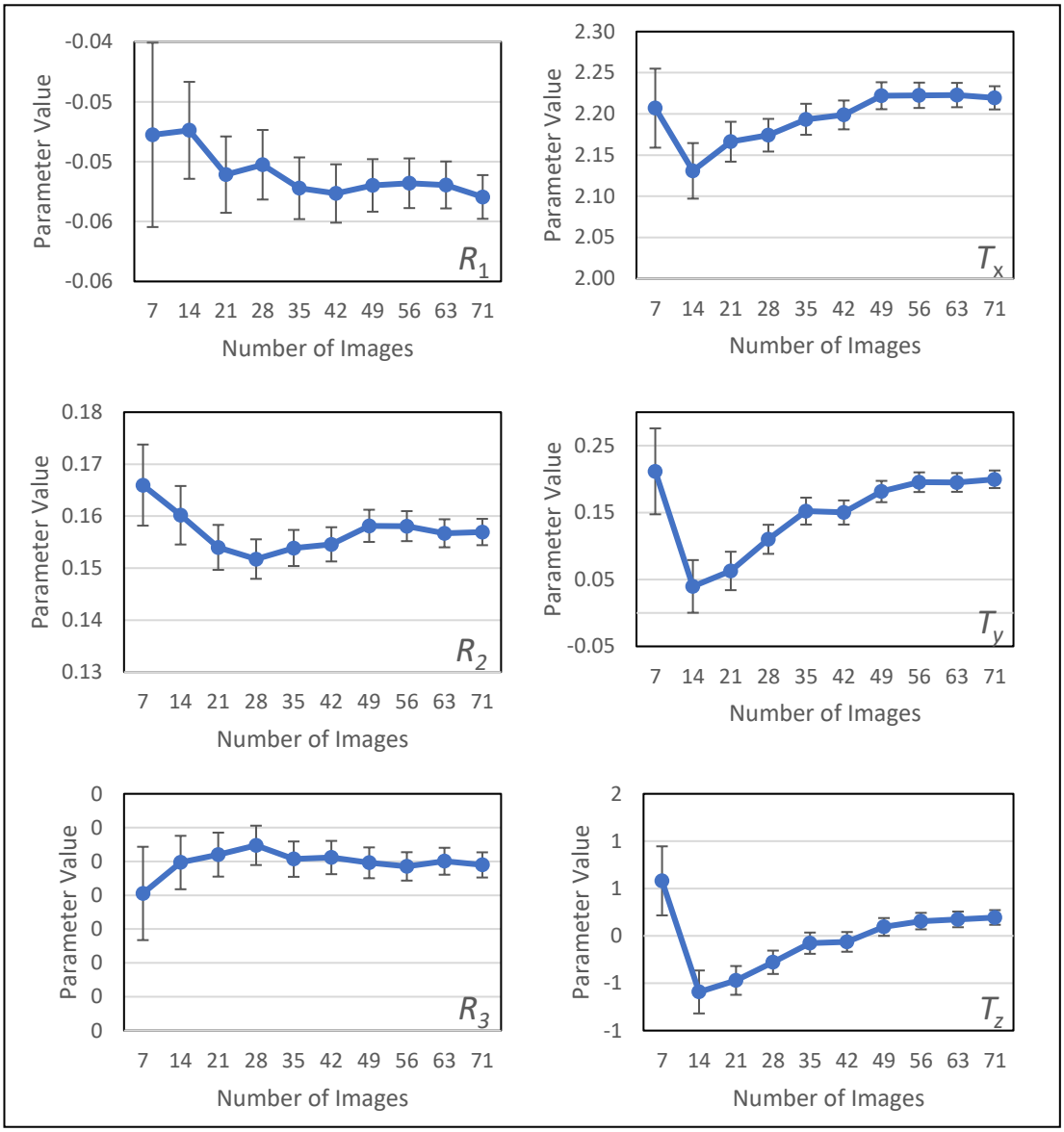


Intrinsic Parameters



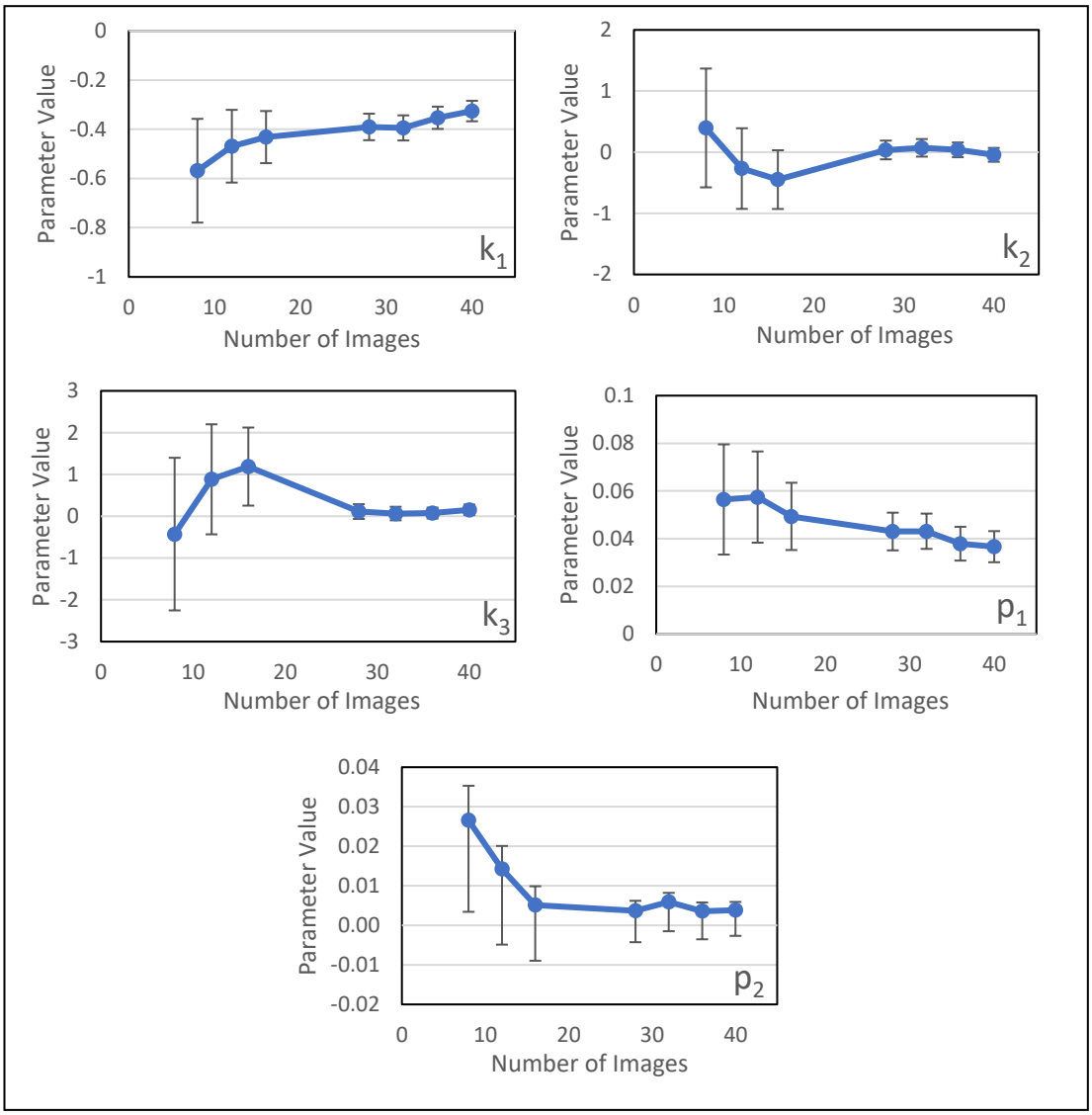


Extrinsic Parameters

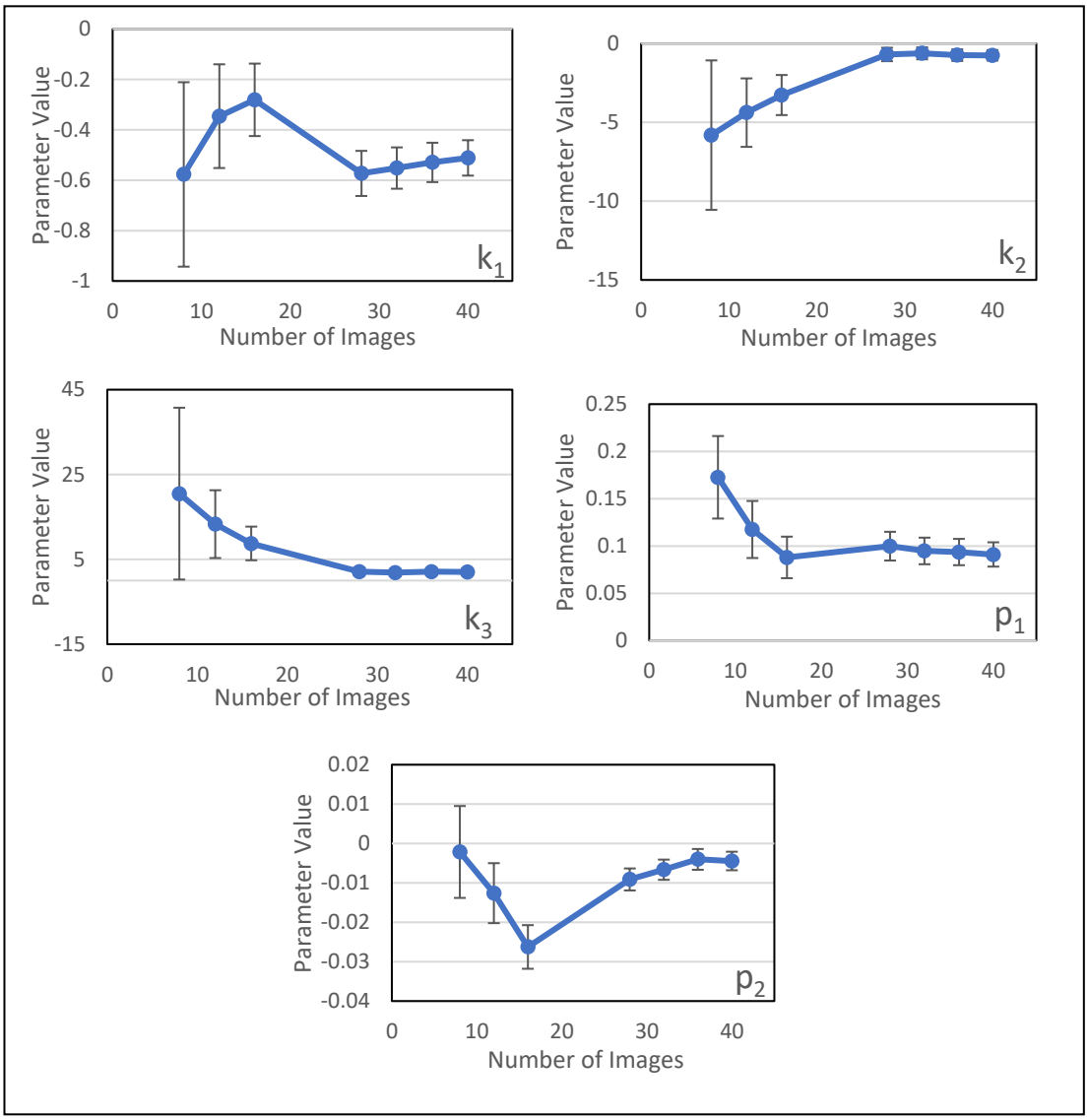


APPENDIX 8A – GRAPHS SHOWING CALIBRATION COEFFICIENTS AND PARAMETERS AGAINST THE NUMBER OF CALIBRATION IMAGES USED

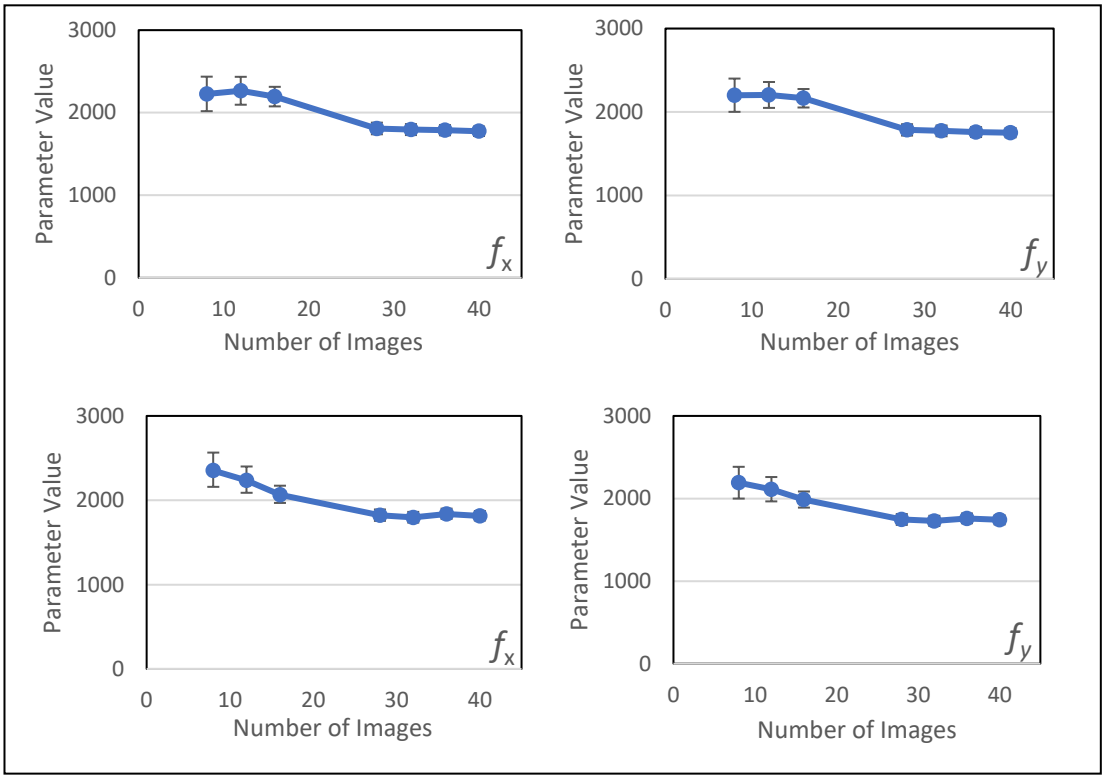
Left Camera Distortion Coefficients



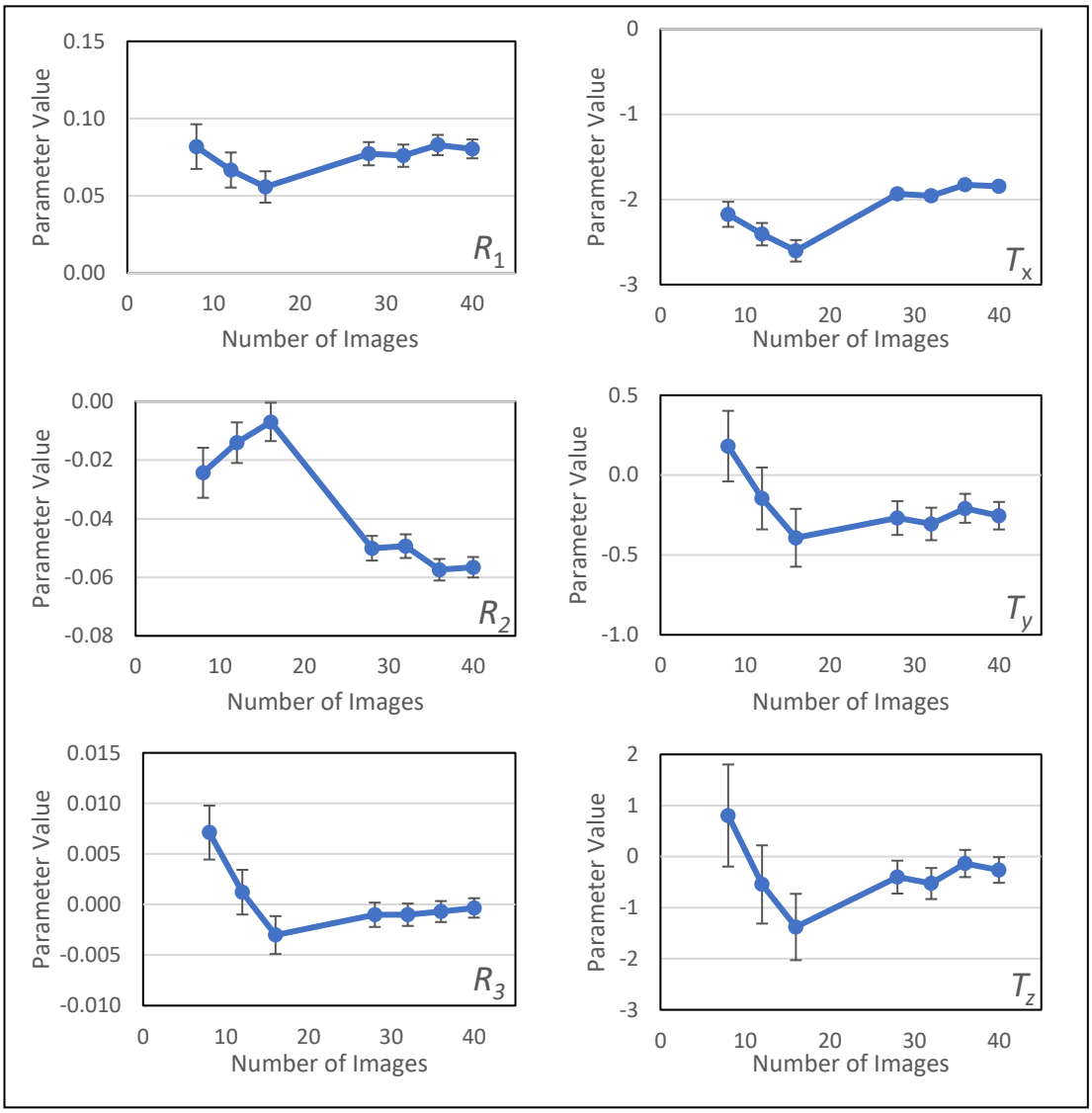
Right Camera Distortion Coefficients



Intrinsic Parameters



Extrinsic Parameters



APPENDIX 8B: THEORETICAL DEPTH RESOLUTION OF THE SLIT LAMP-BASED INSTRUMENT CALCULATION

Derivation of depth resolution equation:

Equation 2.1 can be differentiated as follows:

$$Z = \frac{fb}{x_L - x_R}$$

$$\frac{dZ}{d(x_L - x_R)} = \frac{fb}{(x_L - x_R)^2} \qquad \text{Eq. A8}$$

Where all terms are provided in section 2.3.

The best-case scenario, where the highest resolution cameras were used (i.e., the raspberry pi cameras), is used to calculate the theoretical depth resolution. The ideal stereo camera model relating to this setup, determined through calibration and rectification, is described by the following parameters:

$$f = 7758\ mm * \frac{pixel}{mm}$$

$$b = 2.24\ mm$$

The disparity maps obtained had a disparity $(x_L - x_R)$ range of 234 to 252 pixels. Therefore, the depth resolution (per matching accuracy in pixels), after also accounting for the scale factor of the eye phantom used in the calibration (division by 4.3), ranges from:

$$\frac{dZ}{d(x_L - x_R)} = \mathbf{73.8}\ \frac{\mu \boldsymbol{m}}{\boldsymbol{pixel}}$$

To:

$$\frac{dZ}{d(x_L - x_R)} = \mathbf{63.6}\ \frac{\mu \boldsymbol{m}}{\boldsymbol{pixel}}$$

APPENDIX 9A – STEREO VISION PIPELINE WRITTEN IN C++ FOR THE WORK OF CHAPTER 9

```
////////////////////////////////////////////////////////////////////////////////
//////////////                3D Reconstruction Code                //////////////
////////////////////////////////////////////////////////////////////////////////

/*
Author: Ian Coghill(1) (EngD Student)
Supervised by Dr Mario E Giardini(1), Dr Richard Black(1) & Dr Iain AT Livingstone(2)
1 - Department of Biomedical Engineering, University of Strathclyde
2 - NHS Forth Valley
*/

//////////////////////////////////   INCLUDE LIBRARIES   ////////////////////////////
////////////////////////////////////////////////////////////////////////////////
#include <iostream>
                                    // General
#include <fstream>
#include <string>
#include <conio.h>
#include <Eigen/Core>
#include <Eigen/QR>
#include <cmath>
#include <direct.h>
#include <stdlib.h>
#include <string.h>
#include "opencv2/core/core.hpp"
                // OpenCV
#include "opencv2/core/eigen.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/calib3d/calib3d.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/imgcodecs.hpp"
#include "pcl/common/common.h"                                                  //
Point Cloud Library
#include "pcl/point_types.h"
#include "pcl/point_cloud.h"
#include "pcl/visualization/pcl_visualizer.h"
#include "pcl/common/centroid.h"
#include <pcl/common/transforms.h>
#include <pcl/surface/poisson.h>
#include <pcl/io/vtk_io.h>
#include <pcl/io/ply_io.h>
#include <pcl/features/normal_3d.h>
#include <pcl/surface/mls.h>
#include <boost/thread/thread.hpp>
#include <pcl/filters/statistical_outlier_removal.h>



//////////////////////////////////   MATCHING SETTINGS   ////////////////////////////
////////////////////////////////////////////////////////////////////////////////
int             noOfSquares = 10;                               // No of boxes to match (x and y
direction)
int             dispRange = 60;                                     // Disparity search range
double  correlationThreshold = 0.80;     // Correlation threshold
int             pxspc = 5;                                                  // Dense
reconstruction resolution paramter
int             showProcess = 0;                        // 1 = Show stereo matching, 0 =
don't show


//////////////////////////////////   POINT CLOUD OBJECTS   ////////////////////////////
////////////////////////////////////////////////////////////////////////////////
pcl::PointCloud<pcl::PointXYZ>::Ptr cloud(new pcl::PointCloud<pcl::PointXYZ>);
pcl::PointCloud<pcl::PointXYZ>::Ptr centeredCloud(new pcl::PointCloud<pcl::PointXYZ>);
pcl::PointCloud<pcl::PointNormal>::Ptr cloudWithNormals(new
pcl::PointCloud<pcl::PointNormal>());
```

```
pcl::PointCloud<pcl::PointXYZRGB>::Ptr denseTexturedCloud(new
pcl::PointCloud<pcl::PointXYZRGB>);
pcl::PointCloud<pcl::PointNormal>::Ptr cloudUpsampledNormals(new
pcl::PointCloud<pcl::PointNormal>());
pcl::PointCloud<pcl::PointXYZRGB>::Ptr cloudTranslated0(new
pcl::PointCloud<pcl::PointXYZRGB>());
pcl::PointCloud<pcl::PointXYZ>::Ptr cloudTranslated1000(new
pcl::PointCloud<pcl::PointXYZ>());
pcl::PolygonMesh mesh;
Eigen::Affine3f tMatrix;


//////////////////////////////   VISUALISATION FUNCTIONS   ////////////////////////
////////////////////////////////////////////////////////////////////////////////////
void sparseCloudVisualisation();
void densePointCloudVisualisation();


//////////////////////////   OTHER FUNCTIONS INITIALISATION   ///////////////////
////////////////////////////////////////////////////////////////////////////////////
int disparitySearchCentreEstimation(cv::Rect, cv::Mat, cv::Mat);
void computeSparsePointCloud(cv::Mat, cv::Mat, int, cv::Rect2d, int, int, double, cv::Mat);
void centreCloudData();
void reprojectImagetoMesh(cv::Rect2d, cv::Mat, int, int, cv::Mat);
cv::Rect2d ROISelection(cv::Mat);


////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////
/////////////////////////////////////////   MAIN   /////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////
int main()
{

        std::cout << R"(

+------+.      +------+       +------+       +------+      .+------+
|`.    | `.    |\     |\      |      |      /|     /|    .' |    .'|
|  `+--+---+   | +----+-+     +------+     +-+----+ |   +---+--+'  |
|   |  |   |   | |    | |     |      |     | |    | |   |   |  |   |
+---+--+.  |   +-+----+ |     +------+     | +----+-+   |  .+--+---+
 `. |    `.|    \|     \|     |      |     |/     |/    |.'    | .'
   `+------+     +------+     +------+     +------+     +------+'

                     3D RECONSTRUCTION CODE
                                              (Version 1)

        )" << '\n';


        ////////////////////////////   VARIABLE INITIALISATION   ////////////////////
        ////////////////////////////////////////////////////////////////////////////////
        // Images
        cv::Mat         RGBimageL, RGBimageR;
        cv::Mat         RGBimageL_Rect, RGBimageR_Rect;
        cv::Mat         inputStereoImages, rectifiedImages, newImage3;

        // Rectification
        cv::Mat K1, D1, K2, D2;
        cv::Mat Q, R, R1, P1, R2, P2;
        cv::Vec3d       T;
        cv::Mat         map11, map12, map21, map22, img1r, img2r;
        cv::Rect        roi1, roi2;
        cv::Point       Point1, Point2;

        // Stereo Matching Variables
        int     dispCentre;


        ////////////////////////   LOAD STEREO CALIBRATION PARAMETERS   /////////////////
```

```
        ////////////////////////////////////////////////////////////////////////////
        cv::FileStorage fs("Data/stereoParams.yml", cv::FileStorage::READ);
        fs["K1"] >> K1; fs["D1"] >> D1;
        fs["K2"] >> K2; fs["D2"] >> D2;
        fs["R"] >> R;
        fs["T"] >> T;

        ////////////////////////////////   LOAD IMAGES   ///////////////////////////
        ////////////////////////////////////////////////////////////////////////////
        RGBimageL = cv::imread("Data/L.png");
        RGBimageR = cv::imread("Data/R.png");
        std::cout << "Size of Left Image: " << RGBimageL.size() << std::endl;
        std::cout << "Size of Right Image: " << RGBimageR.size() << std::endl;

        ////////////////////////////////   DISPLAY INPUT IMAGES   ///////////////////////
        ////////////////////////////////////////////////////////////////////////////
        cv::namedWindow("Input Stereo Image Pair", cv::WINDOW_NORMAL);
        cv::hconcat(RGBimageL, RGBimageR, inputStereoImages);
        imshow("Input Stereo Image Pair", inputStereoImages);
        cv::waitKey();


        ////////////////////////////////   RECTIFY INPUT IMAGES   ///////////////////////
        ////////////////////////////////////////////////////////////////////////////
        cv::Size img_size = (RGBimageL.size());
        stereoRectify(K1, D1, K2, D2, img_size, R, T, R1, R2, P1, P2, Q,
cv::CALIB_ZERO_DISPARITY, -1, img_size, &roi1, &roi2);
        cv::initUndistortRectifyMap(K1, D1, R1, P1, img_size, CV_16SC2, map11, map12);
        cv::initUndistortRectifyMap(K2, D2, R2, P2, img_size, CV_16SC2, map21, map22);
        Q.convertTo(Q, CV_32F);
        cv::remap(RGBimageL, RGBimageL_Rect, map11, map12, cv::INTER_LINEAR);
        cv::remap(RGBimageR, RGBimageR_Rect, map21, map22, cv::INTER_LINEAR);


        ////////////////////////   DISPLAY AND SAVE RECTIFIED IMAGES   /////////////////
        ////////////////////////////////////////////////////////////////////////////
        cv::namedWindow("Rectified Images", cv::WINDOW_NORMAL);
        cv::hconcat(RGBimageL_Rect, RGBimageR_Rect, rectifiedImages);
        imshow("Rectified Images", rectifiedImages);
        cv::waitKey();
        cv::imwrite("Data/RectifiedLeftImage.png", RGBimageL_Rect);
        cv::imwrite("Data/RectifiedRightImage.png", RGBimageR_Rect);


        ////////////////////////   SELECT REGION IN IMAGE TO BE ANALYSED   ///////////////
        ////////////////////////////////////////////////////////////////////////////
        cv::Rect2d ROI = ROISelection(RGBimageL_Rect);


        ////////////////////////   Estimate Sparse Disparity Map   ///////////////////////
        ////////////////////////////////////////////////////////////////////////////
        // Disparity Search Centre Estimation
        dispCentre = disparitySearchCentreEstimation(ROI, RGBimageR_Rect, RGBimageL_Rect);
        // Estimate Sparse Point Cloud
        computeSparsePointCloud(RGBimageL_Rect, RGBimageR_Rect, noOfSquares, ROI, dispCentre,
dispRange, correlationThreshold, Q);
        centreCloudData();
        boost::thread workerThread(sparseCloudVisualisation);
        cv::waitKey();

        ////////////////////////              Fit Mesh to Cloud            ///////////////
        ////////////////////////////////////////////////////////////////////////////
        // Estimate Normals
        pcl::MovingLeastSquares<pcl::PointXYZ, pcl::PointNormal> mls;
        pcl::search::KdTree<pcl::PointXYZ>::Ptr tree(new
pcl::search::KdTree<pcl::PointXYZ>());
        mls.setInputCloud(centeredCloud);
        mls.setComputeNormals(true);
        mls.setSearchMethod(tree);
        mls.setSearchRadius(3.5);
```

```
        mls.setUpsamplingMethod(pcl::MovingLeastSquares<pcl::PointXYZ,
pcl::PointNormal>::RANDOM_UNIFORM_DENSITY);
        mls.setPolynomialFit(true);
        mls.setPolynomialOrder(2);
        mls.process(*cloudUpsampledNormals);
        pcl::PointXYZ PointT;
        // Flip any abnormally oriented normals
        for (int z = 0; z < (int(centeredCloud->width)); z++)
        {
                PointT = centeredCloud->points[z];

                pcl::flipNormalTowardsViewpoint(PointT, 0, 0, -5000, cloudUpsampledNormals-
>points[z].normal_x, cloudUpsampledNormals->points[z].normal_y, cloudUpsampledNormals-
>points[z].normal_z);

        }
        pcl::concatenateFields(*centeredCloud, *cloudUpsampledNormals, *cloudWithNormals);
        // Flit the mesh
        pcl::Poisson<pcl::PointNormal> poisson;
        poisson.setInputCloud(cloudWithNormals);
        poisson.setDepth(6);
        poisson.setSolverDivide(6);
        poisson.setIsoDivide(6);
        poisson.setSamplesPerNode(4);
        poisson.reconstruct(mesh);
        pcl::io::saveVTKFile("Data/Mesh.vtk", mesh);

        ////////////////////////     Obtain Dense Textured Point Cloud
////////////////////////
        reprojectImagetoMesh(ROI, RGBimageL_Rect, pxspc, dispCentre, Q);
        boost::thread workerThread2(densePointCloudVisualisation);
        cv::waitKey();

}
////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////
/////////////////////////////////////   MAIN END  //////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////


int disparitySearchCentreEstimation(cv::Rect ROI, cv::Mat RightImageRect, cv::Mat
LeftImageRect)
{
        ////////////////////////////////////////// DISPARITY SEARCH CENTRE ESTIMATION
        std::cout << "Now estimating the disparity search centre" << std::endl;
        int boxWidthDS = (ROI.width);
        int boxHeightDS = (ROI.height);
        cv::Point TopLeftPoint = cv::Point(ROI.x, ROI.y);
        cv::Mat resultDS, LeftImageRectGrey, RightImageRectGrey, LeftImageRectGreyDisp,
LeftImageBlock;
        cv::Rect2d rL, rR;
        cv::Size img_size;
        img_size = RightImageRect.size();

        // Create the result matrix
        int result_colsDS = RightImageRect.cols - boxWidthDS + 1;
        int result_rowsDS = RightImageRect.rows - boxHeightDS + 1;
        resultDS.create(result_rowsDS, result_colsDS, CV_32FC1);
        cv::cvtColor(LeftImageRect, LeftImageRectGrey, cv::COLOR_BGR2GRAY);
        cv::cvtColor(RightImageRect, RightImageRectGrey, cv::COLOR_BGR2GRAY);
        LeftImageRectGrey.copyTo(LeftImageRectGreyDisp);
        cv::Mat RightImageRectIDS;
        cv::Mat disps;
        double          minValDS, maxValDS;
        cv::Point       minLocDS, maxLocDS, matchLocDS;

        // crop the block from the left image
        rL = cv::Rect2d((TopLeftPoint.x), (TopLeftPoint.y), boxWidthDS, boxHeightDS);
        LeftImageBlock = LeftImageRect(rL);
```

```
        // crop the right image so that search is one dimensional
        rR = cv::Rect2d(0, (TopLeftPoint.y), img_size.width, boxHeightDS);
        RightImageRectIDS = RightImageRect(rR);


        /// Do the Matching and Normalize
        cv::matchTemplate(RightImageRectIDS, LeftImageBlock, resultDS, cv::TM_CCORR_NORMED);
        std::cout << 5 << std::endl;
        cv::normalize(resultDS, resultDS, 0, 1, cv::NORM_MINMAX, -1, cv::Mat());
        std::cout << 6 << std::endl;
        minMaxLoc(resultDS, &minValDS, &maxValDS, &minLocDS, &maxLocDS, cv::Mat());
        matchLocDS = maxLocDS;
        std::cout << 7 << std::endl;
        // Store the value
        disps.push_back((matchLocDS.x) - (TopLeftPoint.x));

        std::cout << "3" << std::endl;
        minMaxLoc(disps, &minValDS, &maxValDS, &minLocDS, &maxLocDS, cv::Mat());
        int dispCentre = ((maxValDS + minValDS) / 2);

        std::cout << "Done, min is: " << minValDS << ", max is: " << maxValDS << ", disparity
search centre is: " << dispCentre << std::endl;

        rR.x = rR.x + matchLocDS.x;
        rR.y = rR.y + matchLocDS.y;
        rR.width = boxWidthDS;
        /*
        cv::Mat newImage3;
        cv::rectangle(LeftImageRectGrey, rL, cv::Scalar(255, 255, 255), 3);
        cv::rectangle(RightImageRectGrey, rR, cv::Scalar(255, 255, 255), 3);
        cv::hconcat(LeftImageRectGrey, RightImageRectGrey, newImage3);
        cv::namedWindow("Search Visualisation2", cv::WINDOW_NORMAL);
        imshow("Search Visualisation2", newImage3);
        cv::waitKey();
        */


        return dispCentre;
}

void computeSparsePointCloud(cv::Mat LeftImageRect, cv::Mat RightImageRect, int noOfSquares,
cv::Rect2d ROI, int dispCentre, int dispRange, double correlationThreshold, cv::Mat Q)
{
        // Initialise Variables
        cv::Mat dispS2, LeftImageBlock, RightImageBlock;
        int TOPLEFTX, TOPLEFTY, TOPLEFTDX, TOPLEFTDY;
        int TLX, TLY, BW, BH;
        int centX, centY;
        cv::Mat         resultDS, resultD;
        double          minValDS, maxValDS, minValD, maxValD;
        cv::Point       minLocDS, maxLocDS, maxLocD, minLocD;
        double          minValdisp, maxValdisp;
        cv::Point       minLocdisp, maxLocdisp, matchLocDS;
        cv::Mat CorrS2;
        cv::Mat dispS2M;
        cv::Mat A(7, 3, CV_32FC1);
        Eigen::MatrixXd dispA, dispY, pinv;
        double aS, bS, cS, sampleDisp, SPdispX, SPdispY;
        cv::Mat SM, sampleSx, sampleSy, sampleS;
        std::vector<double> X;
        std::vector<double> Y;
        std::vector<double> Z;
        std::vector<std::vector<double> > P;
        cv::Mat LeftImageRectGrey, RightImageRectGrey;
        cv::Rect2d rL, rR;


        // Convert to grayscale
        cv::cvtColor(LeftImageRect, LeftImageRectGrey, cv::COLOR_BGR2GRAY);
        cv::cvtColor(RightImageRect, RightImageRectGrey, cv::COLOR_BGR2GRAY);
```

```
        cv::Mat newImage5;
        cv::hconcat(LeftImageRectGrey, RightImageRectGrey, newImage5);
        cv::namedWindow("Search Visualisation2", cv::WINDOW_NORMAL);
        imshow("Search Visualisation2", newImage5);
        cv::waitKey();


        // Get Start Point and Box Dimensions
        cv::Point TopLeftPoint = cv::Point(ROI.x, ROI.y);
        cv::Point BottomRightPoint = cv::Point(((ROI.x + ROI.width)), ((ROI.y +
ROI.height)));
        int boxWidth = (ROI.width) / noOfSquares;
        int boxHeight = (ROI.height) / noOfSquares;

        // Print Q
        std::cout << Q << std::endl;


        // Create Window to Show Search Process (if showProcess = 1)
        cv::Mat RightImageRectGreyDispCopy, LeftImageRectGreyDispCopy, newImage3;
        cv::Rect2d rRvisualisation;
        if (showProcess == 1)
        {
                cv::namedWindow("Search Visualisation", cv::WINDOW_NORMAL);
        }
        std::cout << "Computing topography using " << noOfSquares * noOfSquares << " blocks"
<< std::endl;


        for (int disps2 = 0; disps2 < 7; disps2++) {
                dispS2.push_back(disps2);
        }

        for (int k = 0; k < (noOfSquares); k++) { // shifts in the vertical direction

                for (int j = 0; j < (noOfSquares); j++) { // shifts in the horizontal
direction

                        //LEFT
                        // crop out square in left image
                        TOPLEFTX = (TopLeftPoint.x + (j * boxWidth));
                        TOPLEFTY = (TopLeftPoint.y + (k * boxHeight));
                        BW = boxWidth;
                        BH = boxHeight;
                        centX = TOPLEFTX + (BW / 2);
                        centY = TOPLEFTY + (BH / 2);
                        rL = cv::Rect2d(TOPLEFTX, TOPLEFTY, BW, BH);
                        LeftImageBlock = LeftImageRectGrey(rL);


                        // RIGHT
                        // crop out block from the right image
                        TOPLEFTDX = (TopLeftPoint.x + (j * boxWidth) + (dispCentre -
(dispRange / 2)));
                        TOPLEFTDY = (TopLeftPoint.y + (k * boxHeight));
                        BW = boxWidth + dispRange;
                        BH = boxHeight;
                        rR = cv::Rect2d(TOPLEFTDX, TOPLEFTDY, BW, BH);
                        RightImageBlock = RightImageRectGrey(rR);



                        if (showProcess == 1)
                        {

                                LeftImageRectGrey.copyTo(LeftImageRectGreyDispCopy);
                                cv::rectangle(LeftImageRectGreyDispCopy, rL, cv::Scalar(255,
255, 255), 5);

                                RightImageRectGrey.copyTo(RightImageRectGreyDispCopy);
```

```
                                cv::rectangle(RightImageRectGreyDispCopy, rR,
cv::Scalar(255, 0, 0), 5);
                                cv::hconcat(LeftImageRectGreyDispCopy,
RightImageRectGreyDispCopy, newImage3);
                                imshow("Search Visualisation", newImage3);
                                cv::waitKey();

                        }


                        // Determine correlation coefficient
                        cv::matchTemplate(LeftImageBlock, RightImageBlock, resultD,
cv::TM_CCORR_NORMED);
                        minMaxLoc(resultD, &minValD, &maxValD, &minLocD, &maxLocD,
cv::Mat());

                        // Get the correlation in the range -3 to 3 over the point of
highest disparity
                        for (int cor = 0; cor < (7); cor++) { // s
                                CorrS2.push_back(resultD.at<float>(0, maxLocD.x - 3 + cor));
                        }


                        // Fit quadratic to data and find sub pixel disparity corresponding
to highest correlation
                        A.setTo(1);
                        dispS2M = dispS2.mul(dispS2);
                        dispS2M.copyTo(A.col(0));
                        dispS2.copyTo(A.col(1));
                        cv::cv2eigen(A, dispA);
                        cv::cv2eigen(CorrS2, dispY);
                        pinv = (dispA.completeOrthogonalDecomposition().pseudoInverse());
                        pinv = pinv * dispY;
                        cv::eigen2cv(pinv, SM);
                        aS = SM.at<double>(0, 0);
                        bS = SM.at<double>(0, 1);
                        cS = SM.at<double>(0, 2);
                        SPdispX = (dispCentre - (dispRange / 2)) + (maxLocD.x - 3) - ((bS) /
(2 * (aS))); // Sub Pixel Disparity
                        SPdispY = (((-((bS) / (2 * (aS)))) * (-((bS) / (2 * (aS)))) * aS) +
((-((bS) / (2 * (aS)))) * bS) + cS); // Corresponding Correlation
                        std::cout << "Pixel Accurate Disparity: " << (dispCentre -
(dispRange / 2)) + (maxLocD.x) << ", Sub Pixel Disparity: " << SPdispX << ", Difference: " <<
double((dispCentre - (dispRange / 2)) + (maxLocD.x) - SPdispX) << std::endl;

                        // Add to XYZ matrix if it exceeds correlation threshold
                        if (SPdispY > correlationThreshold)
                        {
                                if ((SPdispX < 1000) && (SPdispX > -1000))
                                {

                                        // Create the xyz vector
                                        X.push_back(centX);
                                        Y.push_back(centY);

                                        Z.push_back(-SPdispX);

                                }
                        }

                        CorrS2.release();
                        cv::Mat CorrS2;

                }

        }


        // Place XYZ data in vector P
        P.push_back(X);
        P.push_back(Y);
```

```
		P.push_back(Z);


		// Reproject points to 3D space
		cv::Mat_<float> tmpvec(4, 1);
		pcl::PointXYZ pcpoint;
		for (int z = 0; z < (X.size()); z++) {

			tmpvec(0) = P[0].at(z);
			tmpvec(1) = P[1].at(z);
			tmpvec(2) = P[2].at(z);
			tmpvec(3) = 1;

			tmpvec = Q * tmpvec;
			tmpvec /= tmpvec(3);


			pcpoint.x = tmpvec(0);
			pcpoint.y = tmpvec(1);
			pcpoint.z = tmpvec(2);


			cloud->push_back(pcpoint);
		}


}

void centreCloudData()
{
		Eigen::Vector4f centroid;
		pcl::compute3DCentroid(*cloud, centroid);

		pcl::getTransformation((-centroid[0]), (-centroid[1]), (-centroid[2]), 0, 0, 0,
tMatrix);

		pcl::transformPointCloud(*cloud, *centeredCloud, tMatrix);

		pcl::io::savePLYFileBinary("Data/SparsePointCloud.ply", *centeredCloud);
		std::cout << "Sparse Point Cloud Saved" << std::endl;
}

void sparseCloudVisualisation()
{
		// Setup the visualiser
		boost::shared_ptr<pcl::visualization::PCLVisualizer> viewer(new
pcl::visualization::PCLVisualizer("3D Viewer"));
		viewer->setBackgroundColor(0, 0, 0);

		viewer->addPointCloud<pcl::PointXYZ>(centeredCloud, "sample cloud");

		while (!viewer->wasStopped())
		{

			viewer->spinOnce(100);
			boost::this_thread::sleep(boost::posix_time::microseconds(100000));

		}
}

void densePointCloudVisualisation()
{
		// Setup the visualiser
		boost::shared_ptr<pcl::visualization::PCLVisualizer> viewer(new
pcl::visualization::PCLVisualizer("3D Viewer"));
		viewer->setBackgroundColor(0, 0, 0);

		viewer->addPointCloud<pcl::PointXYZRGB>(denseTexturedCloud, "Final_PC_Overlay");


		while (!viewer->wasStopped())
```

```
        {

                viewer->spinOnce(100);
                boost::this_thread::sleep(boost::posix_time::microseconds(100000));

        }

}

void reprojectImagetoMesh(cv::Rect2d ROI, cv::Mat LeftImageRect, int pxspc, int dispCentre,
cv::Mat Q)
{

        // mesh intersection algorithm
        std::vector<int> cX0;
        std::vector<int> cY0;
        std::vector<int> cZ0;
        std::vector<int> cR0;
        std::vector<int> cG0;
        std::vector<int> cB0;
        int cCPx0;
        int cCPy0;
        std::vector<int> cX1000;
        std::vector<int> cY1000;
        std::vector<int> cZ1000;
        std::vector<std::vector<int> > cP0;
        std::vector<std::vector<int> > cP1000;
        cv::Vec3b vColTemp;

        Eigen::Vector3f vP1;
        Eigen::Vector3f vP2;
        Eigen::Vector3f vP3;
        Eigen::Vector3f vR1;
        Eigen::Vector3f vR2;
        Eigen::Vector3f vPIP;
        float vXtemp, vYtemp, vZtemp, vRtemp, vGtemp, vBtemp;
        float vR1x, vR1y, vR1z, vR2x, vR2y, vR2z;
        float vP1x, vP1y, vP1z, vP2x, vP2y, vP2z, vP3x, vP3y, vP3z;
        Eigen::Vector3f vNormal, vIntersectPos;

        cv::Point TopLeftPoint = cv::Point(ROI.x, ROI.y);
        //Create two sets of points above and below mesh but with only different disparity
        for (int Ypixel = 1; Ypixel < ((ROI.height) / pxspc); Ypixel++) {
                for (int Xpixel = 1; Xpixel < ((ROI.width) / pxspc); Xpixel++) {

                        cCPx0 = (TopLeftPoint.x + (Xpixel * pxspc)); // get x and y of each
pixel in selected area
                        cCPy0 = (TopLeftPoint.y + (Ypixel * pxspc));

                        cX0.push_back(cCPx0); // create the components of the dataset with
disparity 40 below mesh and RGB values
                        cY0.push_back(cCPy0);
                        if (dispCentre < 0) {

                                cZ0.push_back((dispCentre)-200);
                        }
                        if (dispCentre > 0) {

                                cZ0.push_back((-dispCentre) - 200);
                        }
                        vColTemp = LeftImageRect.at<cv::Vec3b>(cCPy0, cCPx0);
                        cR0.push_back(vColTemp[2]);
                        cG0.push_back(vColTemp[1]);
                        cB0.push_back(vColTemp[0]);

                        cX1000.push_back(cCPx0); // create the components of the dataset
with disparity 40 above mesh
                        cY1000.push_back(cCPy0);
                        if (dispCentre < 0) {

                                cZ1000.push_back((dispCentre)+200);
```

```
                        }

                        if (dispCentre > 0) {

                                cZ1000.push_back((-dispCentre) + 200);
                        }


                }
        }

        cP0.push_back(cX0); // create the dataset with disparity 50 below mesh and RGB values
        cP0.push_back(cY0);
        cP0.push_back(cZ0);
        cP0.push_back(cR0);
        cP0.push_back(cG0);
        cP0.push_back(cB0);

        cP1000.push_back(cX1000); // create the dataset with disparity 50 above mesh
        cP1000.push_back(cY1000);
        cP1000.push_back(cZ1000);

        // Convert these datasets from pixels to mm
        pcl::PointCloud<pcl::PointXYZRGB>::Ptr point_cloud_ptr0(new
pcl::PointCloud<pcl::PointXYZRGB>);
        pcl::PointXYZRGB pcpoint0;

        cv::Mat_<float> tmpvec(4, 1);

        pcl::PointXYZ pcpoint;
        std::vector<double> above;
        std::vector<double> below;

        // 0 disparity dataset
        for (int z = 0; z < (cX0.size()); z++) {

                tmpvec(0) = cP0[0].at(z);
                tmpvec(1) = cP0[1].at(z);
                tmpvec(2) = cP0[2].at(z);
                tmpvec(3) = 1;

                tmpvec = Q * tmpvec;
                tmpvec /= tmpvec(3);

                pcpoint0.x = tmpvec(0);
                pcpoint0.y = tmpvec(1);
                pcpoint0.z = tmpvec(2);
                pcpoint0.r = cP0[3].at(z);
                pcpoint0.g = cP0[4].at(z);
                pcpoint0.b = cP0[5].at(z);
                below.push_back(tmpvec(0));
                below.push_back(tmpvec(1));
                below.push_back(tmpvec(2));
                below.push_back(cP0[3].at(z));
                below.push_back(cP0[4].at(z));
                below.push_back(cP0[5].at(z));


                point_cloud_ptr0->push_back(pcpoint0);
        }

        pcl::PointCloud<pcl::PointXYZ>::Ptr point_cloud_ptr1000(new
pcl::PointCloud<pcl::PointXYZ>);
        pcl::PointXYZ pcpoint1000;

        // 1000 disparity dataset
        for (int z = 0; z < (cX1000.size()); z++) {

                tmpvec(0) = cP1000[0].at(z);
                tmpvec(1) = cP1000[1].at(z);
                tmpvec(2) = cP1000[2].at(z);
```

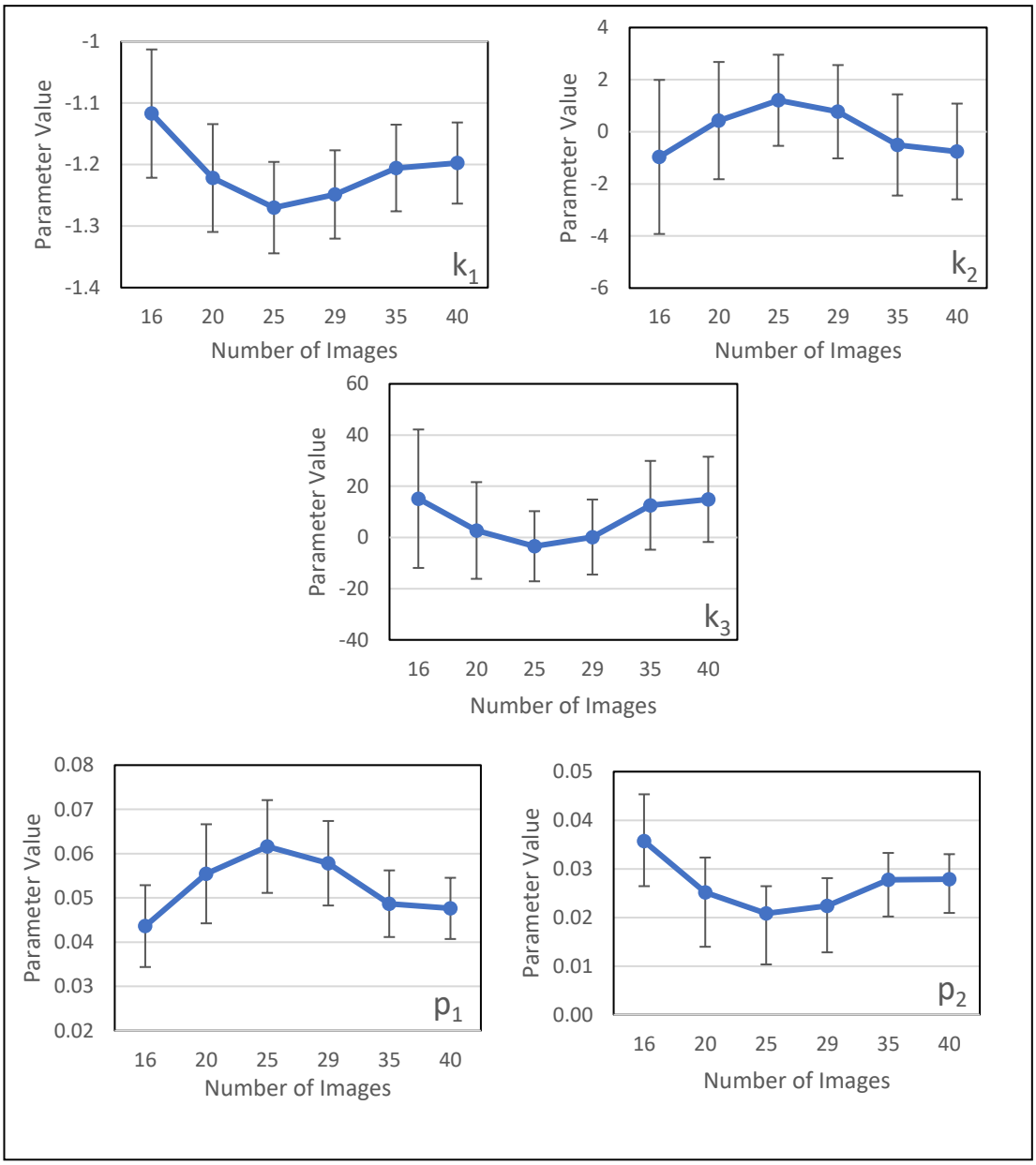
```
        tmpvec(3) = 1;

        tmpvec = Q * tmpvec;
        tmpvec /= tmpvec(3);

        pcpoint1000.x = tmpvec(0);
        pcpoint1000.y = tmpvec(1);
        pcpoint1000.z = tmpvec(2);

        above.push_back(tmpvec(0));
        above.push_back(tmpvec(1));
        above.push_back(tmpvec(2));

        point_cloud_ptr1000->push_back(pcpoint1000);
}

// translate the base and end points the same as the actual sample points
pcl::transformPointCloud(*point_cloud_ptr0, *cloudTranslated0, tMatrix);
pcl::transformPointCloud(*point_cloud_ptr1000, *cloudTranslated1000, tMatrix);

std::cout << "Created points above and below mesh" << std::endl;

// Display them in the visualizer
//std::cout << "Press any key to display the ray base points" << std::endl;
//cv::waitKey();
//viewer->addPointCloud<pcl::PointXYZRGB>(cloudTranslated0, "rays base points");
//std::cout << "Press any key to display the ray end points     " << std::endl;
//cv::waitKey();
//viewer->addPointCloud<pcl::PointXYZ>(cloudTranslated1000, "ray end points");

std::cout << "Rendering in progess...                    " << std::endl;


// try every line with every triangle to check for an intersection
pcl::PointCloud<pcl::PointXYZ>::Ptr vertices(new pcl::PointCloud<pcl::PointXYZ>);
pcl::PointXYZRGB pcpointCol;
pcl::fromPCLPointCloud2(mesh.cloud, *vertices);



for (size_t PCC = 0; PCC < (cloudTranslated0->size()); PCC++) { // try every line

        std::cout << " " << ((PCC * 100) / (cloudTranslated0->size())) << '\r';
        for (int MC = 0; MC < mesh.polygons.size(); MC++) { // try every triangle


                        // Get point cloud

                vR1x = (cloudTranslated0->at(PCC)).x;
                vR1y = (cloudTranslated0->at(PCC)).y;
                vR1z = (cloudTranslated0->at(PCC)).z;

                vR2x = (cloudTranslated1000->at(PCC)).x;
                vR2y = (cloudTranslated1000->at(PCC)).y;
                vR2z = (cloudTranslated1000->at(PCC)).z;

                vP1x = vertices->points[mesh.polygons[MC].vertices[0]].x;
                vP1y = vertices->points[mesh.polygons[MC].vertices[0]].y;
                vP1z = vertices->points[mesh.polygons[MC].vertices[0]].z;
                vP2x = vertices->points[mesh.polygons[MC].vertices[1]].x;
                vP2y = vertices->points[mesh.polygons[MC].vertices[1]].y;
                vP2z = vertices->points[mesh.polygons[MC].vertices[1]].z;
                vP3x = vertices->points[mesh.polygons[MC].vertices[2]].x;
                vP3y = vertices->points[mesh.polygons[MC].vertices[2]].y;
                vP3z = vertices->points[mesh.polygons[MC].vertices[2]].z;

                pcpointCol.r = (cloudTranslated0->at(PCC)).r;
                pcpointCol.g = (cloudTranslated0->at(PCC)).g;
                pcpointCol.b = (cloudTranslated0->at(PCC)).b;
```

```
                        vR1 = Eigen::Vector3f(vR1x, vR1y, vR1z);
                        vR2 = Eigen::Vector3f(vR2x, vR2y, vR2z);

                        /*
                        std::cout << "vR1: " << vR1 << std::endl;
                        std::cout << " " << std::endl;
                        std::cout << "vR2: " << vR2 << std::endl;
                        std::cout << " " << std::endl;
                        std::cout << " " << std::endl;
                        */



                        vP1 = Eigen::Vector3f(vP1x, vP1y, vP1z);
                        vP2 = Eigen::Vector3f(vP2x, vP2y, vP2z);
                        vP3 = Eigen::Vector3f(vP3x, vP3y, vP3z);
                        /*
                        std::cout << "vP1: " << vP1 << std::endl;
                        std::cout << " " << std::endl;
                        std::cout << "vP2: " << vP2 << std::endl;
                        std::cout << " " << std::endl;
                        std::cout << "vP3: " << vP3 << std::endl;
                        */

                        // Get triangle normal
                        Eigen::Vector3f u = vP2 - vP1; // u=P2-P1
                        Eigen::Vector3f v = vP3 - vP1; // v=P3-P1

                                                                                // n=u×v
                        vNormal = u.cross(v);
                        vNormal.normalize();

                        // Find distance from LP1 and LP2 to the plane defined by the
triangle
                        float Dist1 = (vR1 - vP1).dot(vNormal);
                        float Dist2 = (vR2 - vP1).dot(vNormal);

                        //std::cout << Dist1 << std::endl;
                        //std::cout << Dist2 << std::endl;

                        if ((Dist1 * Dist2) >= 0.0f) { // Does the ray intersect the
triangle

//std::cout << "Doesn't intersect" << std::endl;
                                continue;
                        }

                        if (Dist1 == Dist2) { // Is the ray and the triangle parallel
                                                                //std::cout << "Parallel"
<< std::endl;
                                continue;
                        }

                        // Find point on the line that intersects with the plane
                        vIntersectPos = vR1 + (vR2 - vR1) * (-Dist1 / (Dist2 - Dist1));
                        //std::cout << vIntersectPos << std::endl;

                        // Find if the interesection point lies inside the triangle by
testing it against all edges
                        Eigen::Vector3f vTest;

                        vTest = vNormal.cross(vP2 - vP1);
                        if (vTest.dot(vIntersectPos - vP1) < 0.0f) {
                                continue;
                        }
                        //std::cout << "Test 1 " << vIntersectPos << std::endl;
                        //cv::waitKey();
                        vTest = vNormal.cross(vP3 - vP2);
                        if (vTest.dot(vIntersectPos - vP2) < 0.0f) {
                                continue;
```

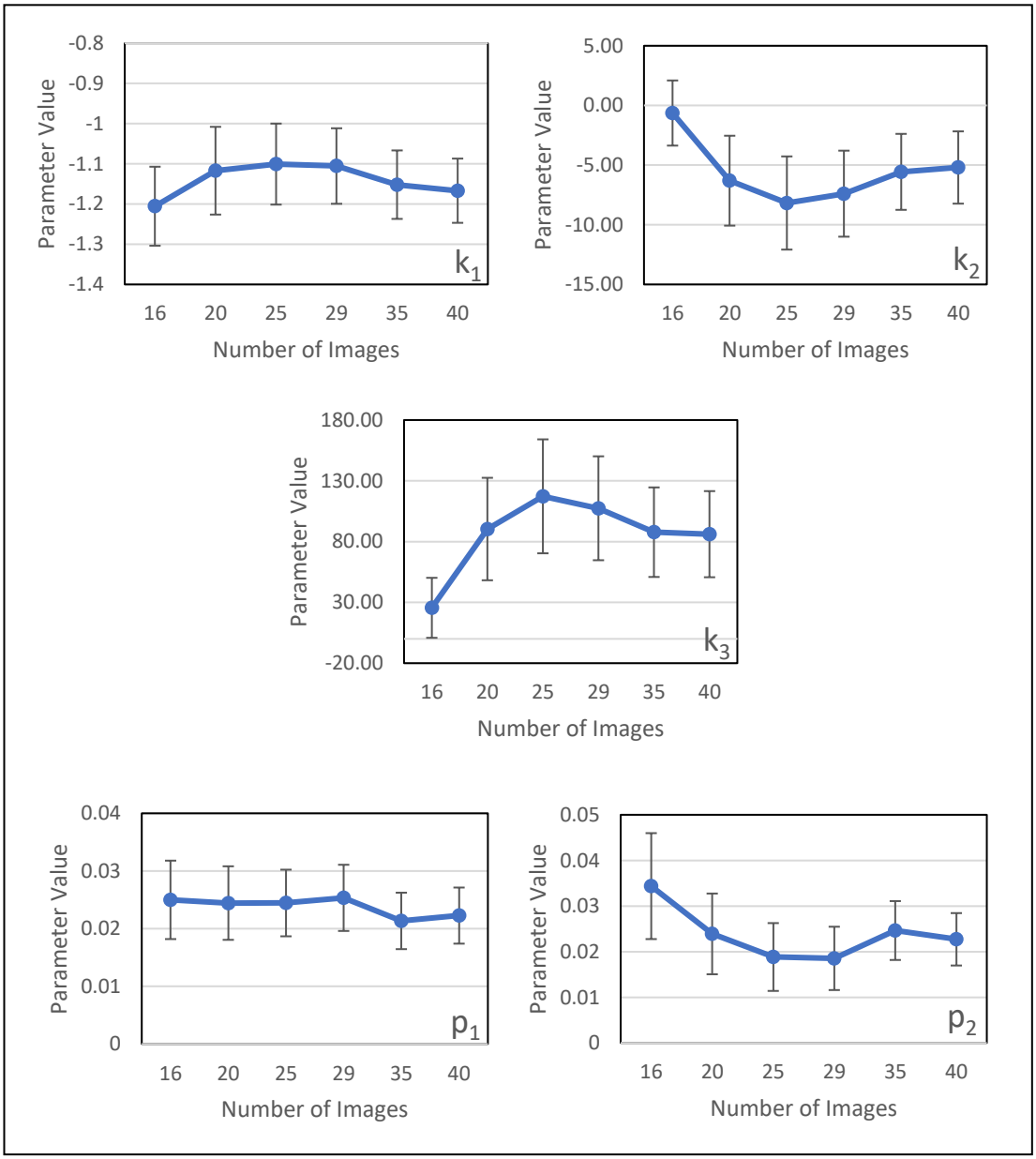
```
                        }
                        //std::cout << "Test 2 " << vIntersectPos << std::endl;
                        //cv::waitKey();
                        vTest = vNormal.cross(vP1 - vP3);
                        if (vTest.dot(vIntersectPos - vP1) < 0.0f) {
                                continue;
                        }

                        vXtemp = vIntersectPos(0);
                        vYtemp = vIntersectPos(1);
                        vZtemp = vIntersectPos(2);

                        pcpointCol.x = vXtemp;
                        pcpointCol.y = vYtemp;
                        pcpointCol.z = vZtemp;
                        pcpointCol.r = (cloudTranslated0->at(PCC)).r;
                        pcpointCol.g = (cloudTranslated0->at(PCC)).g;
                        pcpointCol.b = (cloudTranslated0->at(PCC)).b;

                        denseTexturedCloud->push_back(pcpointCol);

                }


        }

        std::cout << "Rendering complete                                        " << std::endl;



        pcl::io::savePLYFileBinary("Data/DensePointCloud.ply", *denseTexturedCloud);
        std::cout << "Final point cloud saved                                        " << '\r';
        cv::waitKey();

}

cv::Rect2d ROISelection(cv::Mat LeftImageRect)
{
        std::cout << "Select the region of interest for analysis" << '\r';
        cv::Rect2d ROI;
        cv::Mat LeftImageRectCopyResized;
        cv::resize(LeftImageRect, LeftImageRectCopyResized, cv::Size((int(LeftImageRect.cols
/ 4)), (int(LeftImageRect.rows / 4))));
        ROI = cv::selectROI(LeftImageRectCopyResized);
        cv::destroyWindow("ROI selector");
        ROI.x = 4 * ROI.x;
        ROI.y = 4 * ROI.y;
        ROI.width = 4 * ROI.width;
        ROI.height = 4 * ROI.height;

        std::cout << "Region of interest selected
" << std::endl;
        return ROI;
}
```

Appendix 10A – Graphs Showing Calibration Coefficients and Parameters Against the Number of Calibration Images Used

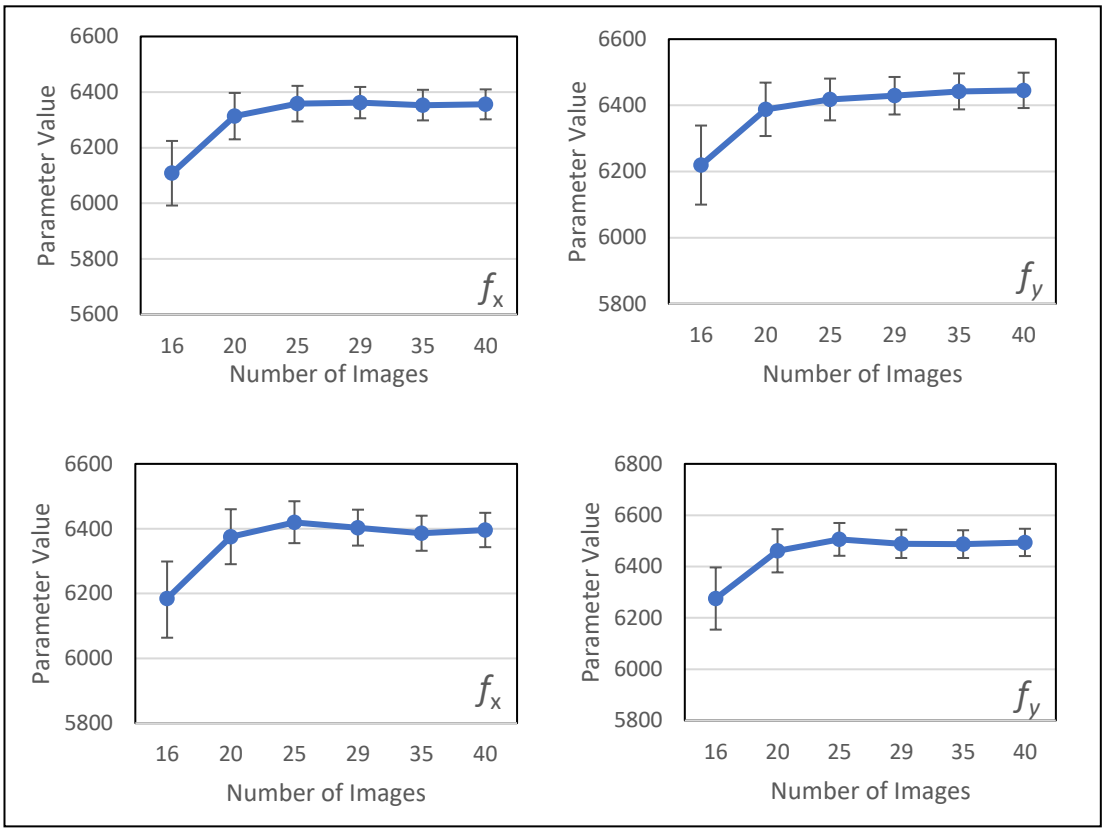
Left Camera Distortion Coefficients

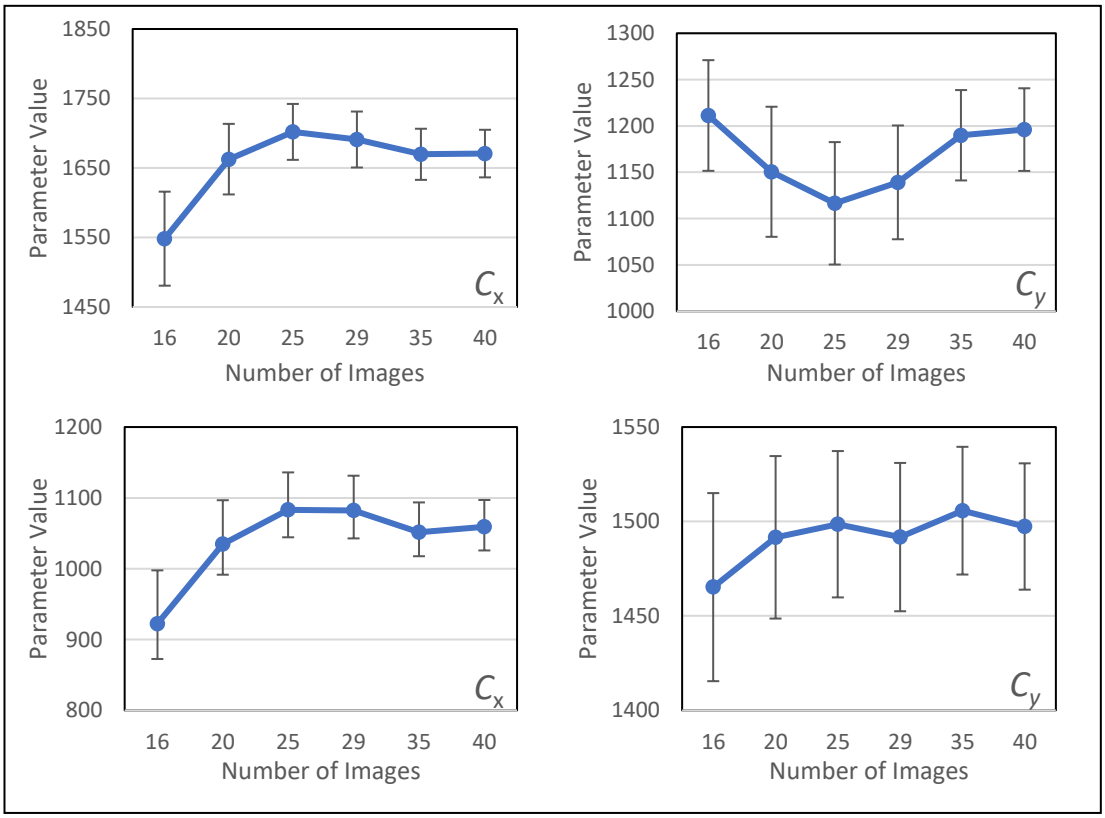


Right Camera Distortion Coefficients

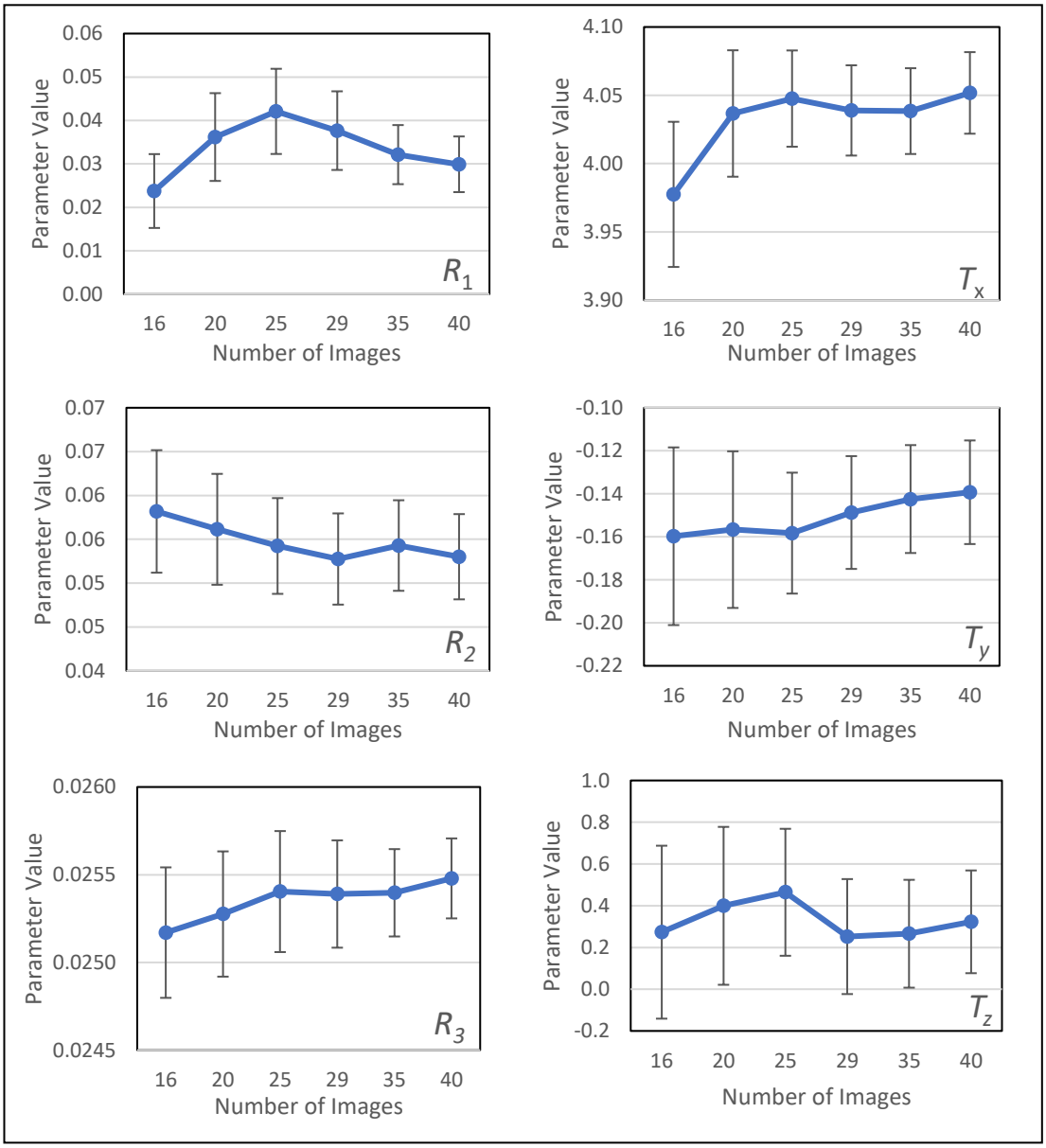


Intrinsic Parameters





Extrinsic Parameters



APPENDIX 10B: THEORETICAL DEPTH RESOLUTION OF THE HANDHELD INSTRUMENT CALCULATION

The ideal stereo camera model relating to the handheld unit, determined through calibration and rectification, is described by the following parameters:

$$f = 6431\ mm * \frac{pixel}{mm}$$

$$b = 4.04\ mm$$

The disparity maps obtained had a disparity $(x_L - x_R)$ range of 402 to 424 pixels. Therefore, the depth resolution (per matching accuracy in pixels), after also accounting for the scale factor of the eye phantom used in the calibration (division by 4.3), ranges from:

$$\frac{dZ}{d(x_L - x_R)} = \mathbf{37.4}\ \frac{\mu\boldsymbol{m}}{\boldsymbol{pixel}}$$

To:

$$\frac{dZ}{d(x_L - x_R)} = \mathbf{33.6}\ \frac{\mu\boldsymbol{m}}{\boldsymbol{pixel}}$$