Low cost 3D Laser Scanning Unit with application to Face Recognition

Prepared by:
Aaron Wetzler

Prepared for:
Faculty of Engineering
Department of Electrical Engineering
University of Cape Town

Supervisor:
Dr Fred Nicolls

October 2005
PLAGIARISM DECLARATION

This is to certify that this dissertation is my own, unaided work. All assistance given has been declared and all information drawn from other resources has been referenced and acknowledged.

Submission of this document is in fulfilment of the requirements of the degree of Bachelor of Science in Mechatronic Engineering at the University of Cape Town.

I also declare that this work has never been submitted before for any degree or examination at any institution.

Signature of Author       Date

--------------------------      -------------------
ACKNOWLEDGEMENTS

This thesis is the product of many hours of work and a number of people have either helped or supported me throughout the time that I have been working on it.

I would specially like to thank:

- My thesis supervisor, Dr Fred Nicolls, who kept me grounded and made sure I didn’t stray so far as to waste my time and energy. He tolerated a host of poorly thought out questions and outbursts of bad inspiration.
- Keith Forbes and Mathew Price who answered all my questions and provided invaluable assistance in terms of visualization techniques.
- Jean-Yves Bouguet, Intel Corp. Without his Matlab Camera Calibration toolbox and stereo triangulation code this project would have been a whole lot more difficult if not impossible.
- My electrical mentor, Samuel Ginsberg, who has provided me endless hours of useful advice, entertaining anecdotes, and a wealth of wisdom that will stay with me for the rest of my career.
- Professor John Greene, who’s course “Fuzzy Logic, Neural Networks and Optimization” last year gave me the crucial skills I needed to use Matlab effectively. The course also provided me with a solid introduction to SVD.
- My fellow students who constantly kept me entertained.
- My mom and dad, coz I doubt I’d be here without them!
- The developers of Matlab. This is an unusual thank you but I think they need to be recognized for producing one of the most easy to use, powerful prototyping languages in the world. Without Matlab and its extensive demonstration and documentation packages this project would have not been possible.

This project is dedicated to my grandparents
ABSTRACT

3D laser scanning using stereoscopic photography is a well understood and widely used technology. This thesis presents the design and construction of such a scanner with particular emphasis on simplicity, off the shelf cheap components and speed. The scanner is designed from the ground up and presented as being easily portable and highly configurable. This approach lends itself to a multitude of optimization techniques and allows for easy extension into a wide variety of applications. The design and construction both in hardware and software comprise the majority of this papers content. However no scanner would be of much use without something to scan within a given context. In light of this fact the application chosen for the scanner is that of a facial recognition system. 2D facial recognition from single static photographs or movie frames is a fascinating topic towards which a wealth of research time has been devoted. However in the last ten years the discovery of various human cognitive properties along with the vast computational complexity of deciphering lighting conditions have forced researchers to look towards 3D recognition. As with any new technology there are a variety of problems that need to be dealt with. Noise elimination, object differentiation, visualization and registration are all dealt with in a computationally feasible manner. Furthermore the modular stages necessary after initial processing are presented and a promising metric for comparing human faces is described.
Contents

CONTENTS

PLAGIARISM DECLARATION II

ACKNOWLEDGEMENTS III

ABSTRACT IV

CONTENTS V

1 INTRODUCTION 1

1.1 BACKGROUND TO THESIS 1
1.2 OBJECTIVES OF THE PROJECT 2
  1.2.1 HIGH LEVEL OBJECTIVES 2
  1.2.2 PROJECT SPECIFIC OBJECTIVES 2
  1.2.3 LOW PRIORITY OBJECTIVES 3
1.3 LIMITATIONS 3
1.4 PLAN OF DEVELOPMENT 4

2 OVERVIEW OF 3D SCANNING 5

2.1 LASER RANGE FINDERS 5
  2.1.1 TIME OF FLIGHT 6
  2.1.2 AMPLITUDE MODULATION 7
  2.1.3 FREQUENCY MODULATION 8
2.2 STRUCTURED LIGHTING 9
  2.2.1 STRUCTURED STRIPES, GRIDS, SHADOWS 9
  2.2.2 ACCORDION FRINGE INTERFEROMETRY 10
  2.2.3 DENSE STEREO 11
  2.2.4 SINGLE STRIPE STRUCTURED LIGHTING 12

3 LASER SCANNING UNIT DESIGN, CONSTRUCTION AND OPERATION 14

3.1 FEATURES OF THE SCANNER 14
3.2 DESIGN AND CONSTRUCTION 14
  3.2.1 PHASE 1: CAMERA PLACEMENT 15
  3.2.2 PHASE 2: SCANNING UNIT 16
  3.2.3 PHASE 3: SENSORS, CIRCUITRY AND OUTPUT 17
  3.2.4 PHASE 4: PROGRAMMING THE CONTROLLER AND INTERFACING WITH THE COMPUTER 19
3.3 OPERATION OF THE LSU 22
  3.3.1 SCANNING PROCEDURE 22
  3.3.2 UNDESIRABLE FEATURES OF CONTINUOUS SCANNING 22

4 CAMERA GEOMETRY 25
## 4.1 Camera Parameters

4.1.1 Intrinsic Camera Model 25
4.1.2 Extrinsic Camera Parameters 28
4.1.3 Distortion Model 29

## 4.2 Epipolar Geometry

4.2.1 Stereo Setup 30
4.2.2 Essential Matrix 31
4.2.3 Fundamental Matrix 32

## 5 Camera Calibration

5.1 The Camera Calibration Toolbox 34
5.1.1 Determining Intrinsic Camera Parameters 35
5.1.2 Determining Extrinsic Stereo Parameters 39
5.1.3 Deciding on the Final Rig Setup 40

## 6 Laser Line Extraction

6.1 Extracting the Line 41
6.1.1 Design of a Laser Extraction Procedure 41
6.1.2 Brute Force Algorithm 42
6.1.3 The Problem of Laser Point Identification 42
6.1.4 On Board Camera Pre-processing 43
6.1.5 The Solution to Laser Point Identification 45
6.1.6 Accuracy 46

## 7 Point Cloud Reconstruction

7.1 Finding the Stereo Matches 48
7.1.1 Left Coordinate Produces Right Epipolar Line 48
7.1.2 Searching Along the Epipolar Line to Find a Matched Point 49
7.1.3 3D Stereo Triangulation 50

## 8 Surface Reconstruction and Visualization

8.1 The Reconstruction-Visualization Relationship 52
8.2 Surface Triangulation 53
8.2.1 Nearest Neighbour Triangulation 53
8.2.2 Delaunay Triangulation 54
8.2.3 Ensuring Neighbourhood Relations 54
8.3 Visualization 57
8.3.1 Novel Mesh Views 57
8.3.2 Texture Mapping 58
8.3.3 Lighting Model 59

## 9 LSU Accuracy

9.1 Ground Truth from the CCT 61
9.1.1 Obtaining Data for Ground Truth Comparison 61
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1.2 Comparing the CCT and the LSU reconstructions</td>
<td>61</td>
</tr>
<tr>
<td>9.1.3 Interpretation of the Gaussian error distribution</td>
<td>64</td>
</tr>
<tr>
<td>9.2 Accuracy as a function of mean distance from left camera optical centre</td>
<td>65</td>
</tr>
<tr>
<td>9.3 Data smoothing</td>
<td>67</td>
</tr>
<tr>
<td>9.3.1 The smoothing process</td>
<td>67</td>
</tr>
<tr>
<td>9.3.2 The effect of smoothing on accuracy</td>
<td>68</td>
</tr>
<tr>
<td>9.4 Comments on accuracy</td>
<td>69</td>
</tr>
<tr>
<td>10 General 3D recognition</td>
<td>70</td>
</tr>
<tr>
<td>10.1 2D recognition</td>
<td>70</td>
</tr>
<tr>
<td>10.2 3D recognition</td>
<td>71</td>
</tr>
<tr>
<td>10.2.1 Direct feature based recognition</td>
<td>72</td>
</tr>
<tr>
<td>10.2.2 Eigen-space based recognition</td>
<td>73</td>
</tr>
<tr>
<td>10.2.3 Acknowledging the need for accurate registration and estimation of orientation</td>
<td>74</td>
</tr>
<tr>
<td>10.2.4 Orientation and expression invariant recognition</td>
<td>74</td>
</tr>
<tr>
<td>11 Steps towards recognition via spherical embedding</td>
<td>76</td>
</tr>
<tr>
<td>11.1 Finding the nose</td>
<td>76</td>
</tr>
<tr>
<td>11.1.1 A brief description of singular value decomposition</td>
<td>76</td>
</tr>
<tr>
<td>11.1.2 Description of the nose finder</td>
<td>77</td>
</tr>
<tr>
<td>11.2 Spherical embedding</td>
<td>78</td>
</tr>
<tr>
<td>11.2.1 Finding the radial vector through the tip of the nose</td>
<td>79</td>
</tr>
<tr>
<td>11.2.2 Mapping the point cloud onto a sphere</td>
<td>79</td>
</tr>
<tr>
<td>11.2.3 Conversion to 2D parametric coordinates</td>
<td>80</td>
</tr>
<tr>
<td>11.2.4 Recognition</td>
<td>81</td>
</tr>
<tr>
<td>12 Conclusions, recommendations and future work</td>
<td>82</td>
</tr>
<tr>
<td>12.1 Conclusions</td>
<td>82</td>
</tr>
<tr>
<td>12.1.1 High level objectives met</td>
<td>82</td>
</tr>
<tr>
<td>12.1.2 LSU design followed specified objectives</td>
<td>82</td>
</tr>
<tr>
<td>12.1.3 Cameras used not fast enough</td>
<td>82</td>
</tr>
<tr>
<td>12.1.4 Camera calibration too slow and sensitive to disturbances</td>
<td>83</td>
</tr>
<tr>
<td>12.1.5 Laser extraction possible within a general environment</td>
<td>83</td>
</tr>
<tr>
<td>12.1.6 Camera pre-processing reduces software oriented computational requirements</td>
<td>83</td>
</tr>
<tr>
<td>12.1.7 Laser line extraction affects scanning accuracy</td>
<td>83</td>
</tr>
<tr>
<td>12.1.8 Point cloud reconstruction contains no immediately meaningful surface information</td>
<td>83</td>
</tr>
<tr>
<td>12.1.9 Delaunay triangulation is effective within the context of the LSU</td>
<td>83</td>
</tr>
<tr>
<td>12.1.10 Scanning accuracy decreases with distance from the camera</td>
<td>84</td>
</tr>
<tr>
<td>12.1.11 Smoothing improves scanner accuracy</td>
<td>84</td>
</tr>
<tr>
<td>12.1.12 Detection of facial orientation for a 3D recognition system is critical to the success of the system</td>
<td>84</td>
</tr>
<tr>
<td>12.1.13 Spherical embedding yields orientation invariant recognition capabilities</td>
<td>84</td>
</tr>
<tr>
<td>12.1.14 Simple nose tip detection is possible</td>
<td>84</td>
</tr>
<tr>
<td>12.2 Recommendations and future work</td>
<td>85</td>
</tr>
<tr>
<td>12.2.1 Improve the scanning system</td>
<td>85</td>
</tr>
<tr>
<td>12.2.2 Improve calibration setup and procedure</td>
<td>85</td>
</tr>
<tr>
<td>12.2.3 Produce an active stereo-vision system</td>
<td>86</td>
</tr>
</tbody>
</table>
12.2.4 BUILD A FULL 3D RECOGNITION SYSTEM

13 REFERENCES

14 APPENDIX A
Chapter 2: Overview of 3D scanning

1 INTRODUCTION

1.1 Background to thesis

The majority of people see the world, around them in three dimensions. This comes so naturally that it is difficult to appreciate what the world might seem like if only two dimensions were available to our senses. Even closing one eye is not enough to experience a two dimensional world because head movements result in sequential image parallax which is used by the brain to create a sense of three dimensions.

Researchers and philosophers initially assumed the perception of three dimensions would be simple to reproduce within a computer environment; but after decades of trying the solution is still not firmly within our grasp. However technology and mathematics have come a long way since the first tentative investigations into three-dimensional reconstruction and since then there has been much research into the subject.

There are many reasons for wanting to be able to recreate the three dimensional structure of the world around us within a computer framework. Equipped with this knowledge we are better able to understand the workings of our own brains. Furthermore, creating more realistic artificial realities is a key pursuit of many game developers and psycho-therapeutic centres. Probably the most useful application to date is that of automated object matching and recognition. Augmented reality is a direct follow on from this research and it can yield novel applications such as in-situ, real time, field-of-view intelligence capabilities. This is especially useful in security situations where certain items or people must be checked against a database as possible security threats.

The mathematics behind three-dimensional geometry is well understood and relatively simple to implement in software or hardware. Companies have been producing three-dimensional vision systems for many years. The majority of these systems are static 3D scanners designed for a specific purpose and until very recently 3D scanners were extremely expensive custom built units with limited capabilities. As can be expected cameras and computers have improved in quality and speed and have gone down in price allowing for superior performance at a reasonable price. The
construction of cost effective scanning units is now feasible and is one of the motivating factors behind this thesis.

No scanning system is of any use without something to scan and therefore an application has been chosen for this project which will combine well understood technology with a problem that is still on the cutting edge of research, that of face recognition.

1.2 Objectives of the project

The objectives of this project were initially overambitious. The aim was to produce a generic scanning unit with the ability to robustly recognize a large dataset of faces. Initial research and practical experience revealed that the objectives should be more limited and made more feasible for a three month thesis project. The objectives are therefore classified according to their relevance.

1.2.1 High level objectives

The list below describes the meta-aims of the project. The intention is for these to become apparent through the material presented in this paper and through the unit constructed for the project. They are to:

- illustrate an ability to use the knowledge learned during a four year Mechatronics degree at the University of Cape Town
- illustrate the ability to learn new material and use it effectively

1.2.2 Project specific objectives

This list of objectives specifically defines what the desired practical outcomes of the project should be. They are to:

- design and construct a portable 3D scanning unit
- develop an eye-safe scanning procedure using an eye-safe laser
- operate in a relatively general environment
• use cheap, easily available components
• make it simple to modify and configure
• scan a subject in less than 10 seconds
• attain relatively real time data logging
• enable modular construction of hardware and software
• develop for scalable design
• generate and use good order-of-complexity algorithms
• provide a useful structure visualization
• perform useful error analysis

1.2.3 Low priority objectives
The initial aim of the project was to provide a 3D facial recognition system but coupled with the task of constructing a scanner this proved to be beyond the time constraints available. As a result the aim was relegated to low priority to help focus on the immediately practical elements.

The objective for recognition was to:
• review current technologies
• attempt an implementation of one of these technologies

1.3 Limitations

Any given project is almost fully defined by two opposing constraints: it’s objectives and it’s limitations. Properly defined these two can go a long way towards helping a developer focus his time effectively. Three months is a short time in which to do a thesis and therefore time was the major limiting factor. Most other limitations are in some way or other time related. The following short list describes the main limitations. They are

• a three month time constraint thereby cutting short any further research and development.
• limited knowledge of three-dimensional epipolar geometry
• the funding available. This helped to define the cost effective objective.
1.4 Plan of development

The structure of this thesis has been chosen to be as sequential and modular as possible. Each chapter aims to describe the construction of either hardware or software components based on what has come before. The results of each chapter are therefore presented and analyzed throughout the chapter rather than at the end of the thesis in a “Results and analysis” section as is commonly found in undergraduate thesis projects. This has been done in order to help the flow of the material and also mimics the way in which the project was undertaken. In other words this thesis presents a sequential, incremental development of the laser scanning unit and all its components.

Chapter 2 gives an overview of 3D scanning techniques. Methodologies such as dense stereo that were not considered for the project are also included for the sake of completeness.

Chapter 3 covers the design and construction of the scanning unit. The material in this chapter is critical for understanding the operation of the scanner in later stages.

Chapter 4 briefly covers the basic geometry needed to understand the point reconstruction process while Chapter 5 goes on to describe how this geometry is modelled and extracted through calibration.

The mechanism by which the laser line is extracted is described in some detail in Chapter 6. The laser line extraction is critical to the success of the scanner and Chapter 7 illustrates how the 3D point cloud is reconstructed based on the material in Chapters 4, 5 and 6.

Chapter 8 considers how the data can be augmented with additional information so that it can be viewed and manipulated as a surface.

No scanning system would be complete without an analysis of its accuracy and this is done in Chapter 9. Point cloud smoothing is also covered in this chapter and highlighted as being a way of improving the scanning accuracy.

Chapter 10 provides a general overview of current facial recognition techniques and then Chapter 11 goes on to present the steps to implementing spherical embedding which is a promising way to represent faces.
2  OVERVIEW OF 3D SCANNING

Any system in which 3D reconstruction is required uses the principle of triangulation. Two or more locations are known to represent the same position in space and this information is then used to triangulate the actual 3-space position coordinates. The process of triangulation is very simple and at its most basic requires nothing more than high school geometry. This means that the most fundamental problem in 3D reconstruction is reduced to identifying corresponding locations. The problem of stereo correspondence appears trivial since the human visual system does stereo pairing and scene reconstruction in a fraction of a second without any help from the conscious mind. However, it turns out that to implement this in a computer is extremely difficult.

Where the brain differs from a computer is that humans have built in feature extraction facilities that automatically match corresponding features between stereo images. In our case those stereo images come from the optic flow through our eyes and the matching is done by hundreds of thousands of neurons networked together that have been fine tuned by millions of years of evolution and years of life experience. Unfortunately, the matching process is so poorly understood that it is currently impossible to match it and more mechanical, less robust methods have had to be developed.

A lot of research has gone into stereo matching algorithms, but the simplest way of creating image matches is to illuminate a target with known structured lighting. This assists the matching process as it dramatically reduces the search space in which to find corresponding features.

It is important to understand the details of different techniques before embarking on the design of a scanner which uses structured lighting. This section discusses some of these different methods and their advantages and disadvantages.

2.1 Laser range finders

Laser range finders (LRFs) are sensors which produce an output that is somehow related to the distance of a target. This information coupled with a known position of the sensor easily yields 3D coordinate data.
The main advantage of LRFs is that they have a huge operating range depending on the desired application. The disadvantage is that they are single shot mechanisms, so in order to produce target information the laser needs to be mechanically scanned from point to point across the scene.

There are three types of LRF: time of flight, amplitude modulation, frequency modulation. These are described below.

### 2.1.1 Time of flight

Time of flight (ToF) is the simplest LRF available. It operates on a pulse-echo time of flight measurement. The sensor sends out a pulse and times how long it takes for the first return pulse. These LRFs are typically used for medium to long range applications such as in vehicle speed traps or terrain mapping.

![Standard format of a TOF range finder](image)

The key design component is the timing mechanism. Light travels at close to 300 thousand kilometres per second. With a 900nm laser this means that to be accurate to within 1 metre the timing sensor and circuitry need to be able to measure echo intervals of as short as 6.6ns. This corresponds to a frequency of 151MHz. Avalanche diodes that can operate at this sort of speed are readily available but comparators and other circuit components become fairly expensive when...
required for speeds above a few megahertz. At these speeds the temperature of the components seriously affects the operating conditions and outputs. This requires additional circuitry and housing to maintain correctly compensated temperature levels.

The major advantage is that the accuracy of TOF improves with increased target range because the error is relatively fixed and range independent. However with increased range comes the need for increased beam strength and therefore energy requirements and eye safety become issues.

2.1.2 Amplitude modulation

Amplitude modulation (AM) techniques involve modulating the laser intensity with sinusoidal amplitude. When the signal is echoed back, its phase is measured relative to the first signal by means of a phase-locked loop. The phase difference is directly related to the distance of the target and relatively accurate measurements can be made using this technique. Figure 2 illustrates the process.

A serious disadvantage of this method is that it is unable to differentiate between objects offset by multiples of the chosen modulation wavelength. If the modulation wavelength is chosen to be 20 metres then the range finder will be unable to discern the difference between targets at 12m, 32m and 52m away. This is due to the periodic nature of a sin wave. The solution is to increase the modulation wavelength to the range over which the LRF is designed to operate. This solution
requires a trade off based design decision because the measurement process increases in time and noise becomes more of a limiting factor. [3] describes a good way of producing high accuracy measurements from amplitude modulated sensors by using two sensors in one measurement unit. The first one is modulated at 10MHz and the second is modulated at 0.24MHz allowing for a coarse-fine measurement technique. The author reports a non-ambiguous accuracy of 0.35mm.

2.1.3 Frequency modulation
Frequency modulation works on a similar principle to amplitude modulation except that instead of measuring the relative position of the returned amplitude it measures the relative position of the returned frequency. Figure 3 shows the principle of operation.

![Figure 3: Frequency modulation](image)

The transmitter sends out a signal linearly increasing in frequency and the receiver electronics measure the difference in transmitted and returned frequency. This difference is then used to determine the distance to the target. The technique is very accurate but suffers from the same modulo ambiguity as amplitude modulation. It is also considerably more difficult to design and build owing to the delicate nature of frequency based measurements.
2.2 Structured lighting

Structured lighting refers to any configuration in which a known lighting configuration is setup. In the case of 3D stereo it is used to simplify the problem of stereo correspondence. A number of methods are available depending on the application, required accuracy and complexity.

2.2.1 Structured stripes, grids, shadows

Projecting a known pattern onto an object and viewing the object from an angle offset from the projector is one of the most common non-contact ways of obtaining 3D data. The core principle is that a stripe projected onto a surface will appear straight when viewed from the projector’s point of view, but will have a specific shape profile from any other viewing angle if it lies on a non-planar target. A CCD camera is usually used to view the stripe and the positions of the apparent contours are measured relative to the known positions within the projectors reference frame. It is then easy to use these disparity measurements to generate depth information and generate a point cloud of 3D data.

![Figure 4: Various useful structured stripe patterns [5]](image)

Figure 4 illustrates the different kinds of projected patterns that can be used when scanning an object. One of the key advantages of this type of scanning is that the picture of the subject is taken in a fraction of a second. This means that there is little possibility of the subject moving and generating motion inaccuracies which are difficult to account for.

---

1 The scanning method chosen for this project requires a subject to stay stationary for a few seconds
Chapter 2: Overview of 3D scanning

The projected shapes in Figure 4 all use coloured lighting to help distinguish the shapes. However, this becomes a problem when scanning coloured objects, as the colours will be distorted; but the effect can be partially cancelled by taking a reference picture of the object in full lighting beforehand and then compensating the intensity differences in post processing.

2.2.2 Accordion Fringe Interferometry

Accordion fringe interferometry (AFI) is currently one of the fastest, most accurate and most robust scanning methods available. It is very similar in nature to the structured lighting described above however its implementation is considerably more complicated. The process requires beam splitting of a single laser point source. The two beams are then widened by lenses and shone onto the target producing interference fringes as shown in Figure 5.

![Figure 5: Standard setup of AFI equipment [6]](image)

The width of the fringes can be varied by changing the angle between the two lenses. This allows for a coarse-fine iterative measurement process. Projectors can do the same thing but not with the same level of accuracy which results from interference between highly monochromatic, highly collimated laser beams. Another advantage is that strong eye-safe laser light is visible in many lighting conditions and the specifics of the ambient lighting are not critical.
The AFI4000 produced by Dimension Photonics is said to be accurate down to 15 micrometres and can operate over a huge working volume. Unfortunately, due to the complex nature of the beam splitting and calibration process, the equipment is expensive and the cost can only be justified where precision measurement is required such as in component shape verification.

2.2.3 Dense stereo

Dense stereo (DS) involves directly finding matching correspondences between image pairs. It has been a topic of extensive research and work is ongoing in the field. Accurate matching is an extremely difficult task especially when the configuration between cameras is initially unknown.

DS usually follows a set sequence of algorithms [8]. First, the images are processed to find features and then these are matched together. This is done in order to find corresponding feature points within both images so that the rotation and translation of the cameras relative to one another can be determined. The projection matrices of the cameras can also be determined from known matched data. Then, once the calibration matrix is known the images are searched with a DS algorithm which triangulates the point matches it finds to produce a point cloud representation of the target scene.

There are some major disadvantages with the current state-of-the-art DS algorithms. Apart from being computationally expensive, they often do not deal well with discontinuities and end up producing streaky or patchy looking images as shown in Figure 7. As might be expected they produce serious errors in the presence of lighting artefacts as most of the algorithms are primarily intensity based.
The main advantage with DS is that it is uncalibrated and requires nothing more than the capture of images or image sequences. If a very robust DS algorithm is ever developed that can match or surpass the biological stereo system then it is fair to assume the use of virtually all other stereo methods will become obsolete\(^2\), as precision measurement simply requires increasing the resolution of the viewing cameras.

### 2.2.4 Single stripe structured lighting

Single stripe structured lighting is a highly simplified version of the projector pattern technique. It involves shining a single stripe of light onto a target and extracting 3D information through the use of either one or two cameras. The use of two cameras coupled with a laser stripe is a robust way of determining matching points and is one of the most widely used techniques for handheld or small prototyping scanners. Single stripe scanning can be extremely accurate depending on the efficacy of the algorithms used to extract the laser line and perform the matching. Figure 8 shows a polygon model produced by the team of the Digital Michalangelo Project\[^10\] who used a high accuracy single stripe scanner.

---

\(^2\) It is the opinion of this author that this is inevitable.
Chapter 2: Overview of 3D scanning

With one camera and a stripe the camera views the shape profile of the laser line on the image and uses the disparity between the known line shape and the projected line shape to calculate depth information. This has a disadvantage in that the position of the laser stripe has to be well calibrated beforehand; then the stripe has to be scanned over the surface at perfectly constant speed or, alternatively, the surface has to be moved beneath the scanner setup at perfectly constant speed. The latter is often preferred as it does not disturb the calibration of the laser with respect to the camera. However, this is inconvenient for general purpose scanning because it requires a motion platform.

A more robust method is to use two calibrated stereo cameras and an uncalibrated laser line. In this configuration the laser is projected onto a target object. Each camera records its view of the laser line and the corresponding views are matched together to generate 3D data points. This makes the process of matching points almost trivial as all that is required are the calibration parameters and a linear search for laser points along the epipolar lines.

This thesis uses an uncalibrated laser to perform 3D scanning and the remainder of this paper discusses its implementation and application.
Chapter 3: Laser scanning unit design, construction and operation

3 LASER SCANNING UNIT DESIGN, CONSTRUCTION AND OPERATION

3.1 Features of the scanner

The full construction of the laser scanning unit (LSU) took over three weeks. The key design principles were that everything should be cheap, adjustable, easily accessible and try to conform to various standards. To this end the following features were part of the scanning unit:

- Individually adjustable left/right camera translation and rotation positions
- Adjustable forwards/backwards translation position of laser head
- Adjustable forwards/backwards translation position of scanning unit
- Tri-axial adjustable photographic tripod
- Loosely fixed circuit boards on compressible foam bases
- Standard 5-12V power plug
- Standardised serial connections 3
- Low cost colour cameras
- Basic motor and scanning unit sourced from an old dot matrix printer
- Simple operating interface via RS232
- Relative portability

3.2 Design and Construction

The scanner was designed in four phases. Each stage was then constructed and tested. The following sections describe the different phases.

3 FireWire and RS232
3.2.1 Phase 1: Camera placement

The cameras came as isolated units with no coverings. In order to protect the circuitry from short circuits a foam base is fixed to each camera. Because the cameras are the most expensive components each camera is not solidly fixed but rather held against its metal frame by nuts and screws. This means the cameras can be taken off and used in other applications when the project is fully dismantled. It also allowed for the cameras to be slightly adjusted when they were badly aligned.

The camera back frame is a 2mm thick steel plate bent at 90 degrees. The base of the back frame has a hole through it. A 6mm screw is placed through the hole and a nut is screwed on tightly when the camera is in place. The screw is positioned such that it goes through a slot on the LSU base plate and fixes the camera in position when the nut is tightened.

The initial design in this phase had the cameras slide along two steel rods but it proved to be ineffective as attaching the rods to anything else meant the cameras were more difficult to take off when the need arose.

The design shown in Figure 9 was the simplest to build and alter as it only required cutting the slots and bending metal sheets, all of which are simple procedures.

![Figure 9: Base plate and camera back frame](image)
3.2.2 Phase 2: Scanning Unit

Once the base plate had been slotted and the cameras had been attached to their back frames, the next step was to design a scanning unit. There were two options available:

- Install a rotating mirror
- Install a linear stepping unit

The linear stepping unit seemed more attractive as linear operations are simpler and less prone to error than rotational ones. To understand this fact it can be seen that a laser shone at different angles will have a greater divergence error the further away the beam is. However a linear platform will provide constant error in measurement irrespective of how far away the beam is from the unit.

Figure 10 shows the differences in error between different kinds of scanning mechanisms. The green line indicates the desired direction and position of the laser line. The blue line indicates the actual position of the laser line. The red line indicates the error between the desired and actual position. The diagram clearly illustrates that if accurate positioning is required, it is easier to reduce error through the use of a linear scanning mechanism.

However, in the end the position measurement was not used for the purposes of the project and a rotational laser positioning would have been a more generic solution with greater application to larger scanning volumes.
Nevertheless the linear scanning unit was chosen and effectively implemented through the use of the scanning mechanism in an old dot matrix printer. Everything needed to produce the linear scanning movement was retained and all other components discarded. What is left is a four pole stepper motor with its housing. A toothed belt connects the stepper motor to the scanning platform which slides along a smooth steel rod. The rod is lightly covered in grease which helps to lubricate and hence smooth the motion of the platform and laser.

The LSU has the laser attached to the sliding unit. The laser is a simple component and generates a vertical laser line by diffracting a highly chromatic, collimated beam through a small lens. The frequency of the laser is around 671nm and the power is tuned sufficiently low so as not to cause damage to eyes.

### 3.2.3 Phase 3: Sensors, circuitry and output

This phase consisted of attaching the end sensors as indicated by 7 in Figure 11, connecting the scanning unit to the base plate and building the LSU circuitry. Initially the slots as shown in Figure 9 were simply holes but this proved to be too difficult to configure accurately. The whole scanning unit can be adjusted forwards or backwards by adjusting screws on the two metal blocks (not numbered) seen to the left and right of label 8.

![Figure 11: LSU scanning elements and circuitry](image)

1: SAA1027 motor driver circuitry
2: End sensors comparator
3: Motorola HC08 GP32 motor controller circuitry and computer serial interface
4: Laser
5: Right camera
6: Left camera
7: End sensors
8: Laser slide
9: Laser tooth belt
10: Motor housing
Figure 11 also shows the circuitry. There are three distinct boards that hold the necessary circuitry. In a neater implementation these boards would be designed onto one board and a printed circuit board would be used. This is a more professional approach and any commercial implementation would require this type of work.

Element 3 in Figure 11 is the core component of the whole LSU driver system. It interfaces with the computer via the serial port and controls the laser and the motor. A more detailed image is displayed in Figure 12. The circuit was built to be reprogrammable at any time. This caused a large number of difficulties as wires need to be disconnected from one another when programming the microcontroller in-situ. The set of DIP switches indicated under component 7 in Figure 12 serves this function. The switches are shifted such that no connected wires interfere with one another when in monitor mode\(^4\). The switches are reversed when in stand alone mode.

![Figure 12: LSU microcontroller/computer interface board](image)

Component 1 in Figure 12 is the GP32 microcontroller and it holds the code for all the operations that need to be performed. This circuit was designed to be very general so that any changes would be easy to make. The microcontroller can be removed and replaced if it breaks which happened three times during testing. The strength of the laser line can be adjusted by tuning the

---

\(^{4}\) Monitor mode allows a programmer to program and observe the operation of the microcontroller. Apart from programming this is very useful for debugging.
potentiometer as indicated by 5 in Figure 12. This is useful because the laser line has a maximum output which is painful when shone on the eyes. Furthermore, the stronger the laser light, the wider its spread which increases the noise in its output, thereby confusing the cameras.

The GP32 sends two signals to the SAA1027 stepper motor driver. One signal controls the direction. The other signal sends each step pulse. A train of pulses causes the motor to drive continuously until the train ends. There is little feedback from the motor circuit as the stepper motor is assumed to move only one step for each pulse and the number of pulses it is sent is controlled by the GP32.

The length of the scanning track is 470 pulses but at high stepping speeds the momentum causes oversteps and only 467 pulses are needed for the full track. This would be a problem if accurate positioning was necessary.

The simplest circuit in the configuration is the sensor comparator. This circuit takes the current produced by two beam break sensors and converts it to a voltage. If the beam is broken the circuitry sends a high signal, otherwise it sends a low signal.

### 3.2.4 Phase 4: Programming the controller and interfacing with the computer

The program was initially designed to interface with the computer in such a way that communication would be kept to a minimum so that the scanning unit would be semi autonomous. However, this design turned out to severely limit the range of activities and commands the scanner could perform.

The initial program had the following capability. It was to

- send the scanner scanning backwards and forwards until a break signal

The program was augmented with a more accessible control interface so that the computer can have specific control resulting in the following capabilities. These are to

- send the scanner scanning backwards and forwards until a break signal
- send the scanner to a specific position
turn the laser on or off

move the scanner a known number of steps

A flowchart of the program operation is shown in Figure 13. This chart illustrates the sequence of instructions that are performed in the microcontroller. It does not include the interfacing flowchart for the computer, as this is constantly changing depending on the application. The main part of the interface simply involves sending the GP32 the commands via the serial cable. The information sent to the computer is: camera data via the FireWire and byte indicators via the RS232 cable when the platform hits one of the end sensors. This is never used but it could be useful in future applications.

All the parts of the scanning unit shown in Figure 14 are simple to put together to produce the full LSU as shown in Figure 15.
The unit is easy to assemble and to dismantle and therefore it lends itself to portable operation but unfortunately in this case the computer is a tower case and not portable. The scanning unit uses two FireWire cameras but it could very easily be modified to use two cheap USB cameras. This would be an advantage as most laptops have at least 2 USB ports.
It is also important to note that RS232 is fast becoming an obsolete standard\(^5\) and any improved version of the LSU should include a USB interface to the microcontroller.

### 3.3 Operation of the LSU

The operation of the scanner mainly involves moving the laser backwards and forwards while using the video cameras. This means that some form of synchronization has to happen between the scanner movement and the camera capture. This section discusses the scanning methods and the disadvantages with continuous scanning.

#### 3.3.1 Scanning procedure

The scanning procedure is variable depending on the application. Nonetheless the main scanning technique known as stop-snap scanning is fairly straightforward. It is

1. turn laser on
2. move the scanner
3. stop the scanner
4. take a picture
5. repeat 2 to 4 as necessary

In the flowchart of Figure 13 one step involves identifying whether the scanning is manual. This means that the computer sends all the motion signals to the GP32. If the scanning is not manual then the scanner moves backwards and forwards in continuous motion until a break signal is sent.

#### 3.3.2 Undesirable features of continuous scanning

Any design process aims to design away all aspects of a system that are undesirable or unnecessary. In terms of this project there are a number of features that were not designed into the system but reared their head after the construction phase. In a commercial environment there would be a redesign and rebuild process. As this project deals mainly with a prototyping unit there is no need to eliminate all the problems as identifying them suffices for our purposes. However, one aspect of the LSU’s operation was impossible to design out and a work-around had to be formulated. The problem was in the continuous scanning method. As described in the

---

\(^5\) Many laptops are no longer made with DB9 connector ports
objectives the scan should last no more then 10 seconds and be as fast as possible. The most obvious way of doing this is to scan continuously and capture images continuously.

The problem that becomes apparent when using the LSU is that if the unit is used in continuous scan mode then the speed of the scanner seriously affects the laser lines stability. At speeds of less than 180 steps per second the effects of stepping cause vibrations of the platform which in turn results in a wide spread in the captured laser line (beam widening).

![Figure 16: Example of beam widening](image)

Beam widening can be clearly seen in Figure 16 where the left line is an image of a stationary beam and the right line is an image of a laser line moving over the page at around 100 steps per second. Beam widening is due to two separate effects:

- Rig vibration due to slight overstep resulting in stepper motor rotor oscillations
- Camera shutter speed (15fps)

The effect of vibration is nonexistent at higher speeds and beam widening is purely a result of residue intensity on the cameras photocells. Unfortunately beam widening is quite serious as it causes multi pixel direction dependent inaccuracies when the scanner runs in continuous mode. To restate this would be to say that when the scanner runs continuously to the left, the laser line positions have errors that are towards the right and when running to the right, the errors lie to the left of the laser line. To further compound the problem each camera takes images at slightly different times even though they are programmed to operate simultaneously. There is a capture time discrepancy of 50ms which is longer then the time between consecutive frames. This means that scanning in continuous scan mode is not possible for the following reasons. They are that

- high speed operation has no vibration effects but it has camera capture time discrepancy and pixel residue intensity due to shutter speed resulting in beam widening
• low speed operation has no noticeable camera discrepancy but it causes vibrations which in turn cause beam widening from pixel residue

These two facts guide the formulation of the scanning process adopted for this project, namely stop-snap scanning as described earlier.
Chapter 4: Camera geometry

4 CAMERA GEOMETRY

This project relies heavily on the parameters determined by the calibration procedure which is described in Chapter 5. These parameters need to be explained and a basic understanding of the epipolar geometry used by the laser extraction and point reconstruction algorithms is necessary in order to grasp the fundamental operation of the LSU.

4.1 Camera parameters

4.1.1 Intrinsic camera model
The simplest model of a camera is the pinhole model. This model states that a camera involves an infinitely small hole through which light enters and forms an inverted image on a canvas as illustrated in Figure 17. This is a relatively accurate assumption and is able to sufficiently model the majority of undistorted camera setups.

Figure 17: Basic pinhole model [12]
Chapter 4: Camera geometry

The model is made simpler by considering the geometry as if the image were on the same side of the focal point as the target. This is achieved by inverting the image.

![Simplified pinhole model](image)

**Figure 18: Simplified pinhole model [12]**

p is a coordinate in 2-space on the plane of the virtual image and P is a coordinate in 3-space on the image in the camera reference frame. P and p are the projection of the target point M on to the camera’s image frame. O is the camera’s optical center and c is the focal point. The length of the focal line Oc is f and is known as the focal length.

If the point M has 3-space coordinates \([X_m, Y_m, Z_m]\) then the projective mapping onto the image plane yielding point P with coordinates \([X_p, Y_p, Z_p]\) can be done by simply dividing the coordinate values of M by \(Z_m\) to reduce them to a focal plane with focal length 1. This is then multiplied by f to get onto the image plane.

Thus

\[
P = \begin{bmatrix}
X_p \\
Y_p \\
Z_p
\end{bmatrix} = \frac{f}{Z_m} \begin{bmatrix}
X_m \\
Y_m \\
Z_m
\end{bmatrix} = \begin{bmatrix}
f * X_m / Z_m \\
f * Y_m / Z_m \\
f
\end{bmatrix}
\]

*Equation 1*

The point p is the xy coordinate set of P and this is directly related to the pixel coordinates of the projection of the target image thus P is reduced to p which is in 2-space homogenous coordinates in a new 2-space reference frame defined on the camera’s virtual image plane [13].
Equation 2

The values $X_P$ and $Y_P$ are not the final pixel values as there are other camera factors to take into account. The first of these is that the image plane center is usually in the top left hand corner so it is necessary to determine the pixel offsets of the principal point of the image.

Figure 19: Pixel distortion

Figure 19 shows an ideal pixel on the left and a model of a non-ideal pixel on the right. Manufacturing processes are often less than perfect and non-square pixels are not uncommon. The aspect ratio of a pixel is defined by $S_x/S_y$. As can also be seen in Figure 19 the pixel axes are not at exactly 90 degrees.

All these factors can be taken into account by the following relations:

$$X_{pixel} = \frac{fX_m}{S_xZ_m} - \frac{f \cos(\theta)Y_m}{\sin(\theta)Z_m} + P_x$$

$$Y_{pixel} = \frac{fY_m}{S_y \sin(\theta)Z_m} + P_y$$

Equation 3

In terms of this project, the angle $\theta$ is so close to 90 degrees that $\sin(\theta)$ can be set to 1 and $\cos(\theta)$ can be changed to a skew coefficient $\alpha$. Furthermore we can assume that the pixels are
square yielding a single scaling factor $S$. The equations can now convert homogenous 3-space coordinates to their pixel image plane projections via the matrix transformation [13]

$$\begin{bmatrix}
X_{\text{pixel}} \\
Y_{\text{pixel}} \\
w
\end{bmatrix} = \begin{bmatrix}
\frac{f}{S} & -f\alpha & P_x & 0 \\
0 & \frac{f}{S} & P_y & 0 \\
0 & 0 & 1 & 0
\end{bmatrix} \begin{bmatrix}
X_m \\
Y_m \\
Z_m \\
1
\end{bmatrix}$$

Equation 4

where the [3 by 4] transformation is known as the intrinsic camera matrix. A quick analysis of this formulation reveals that $w$ is always equal to $Z_m$ and so to restore the homogenous pixel coordinates to their image plane equivalents, dividing through by $Z_m$ produces the desired result.

4.1.2 Extrinsic camera parameters

Any camera is situated within a world environment and therefore the points it observes must somehow be related to the same points in a world reference frame. A formulation of this change of perspective requires a rotation and a translation of the coordinates of the points to change reference frames. The rotation is usually specified about the camera axis. This means the transformation from world coordinate frame to camera coordinate frame is

$$X_c = RX_w + T$$

Equation 5

and in matrix terms [13]

$$\begin{bmatrix}
X_c \\
Y_c \\
Z_c \\
1
\end{bmatrix} = \begin{bmatrix}
R^T & -R^T T \\
0^T & 1
\end{bmatrix} \begin{bmatrix}
X_w \\
Y_w \\
Z_w \\
1
\end{bmatrix}$$

Equation 6
Chapter 4: Camera geometry

In the situation relevant to this project there are two cameras. One camera is defined as being the world reference frame while the other is defined by R and T. Both cameras have intrinsic camera matrices known as \( K_1 \) and \( K_2 \) respectively.

The equivalent right camera image projections of the points in the left cameras reference frame, which is defined as the world reference frame, are

\[
\begin{bmatrix}
X_{e2\,\text{pixel}}^* \\
Y_{e2\,\text{pixel}}^* \\
w
\end{bmatrix}
= \begin{bmatrix}
\frac{f}{S} & -f\alpha & P_x & 0 \\
0 & \frac{f}{S} & P_y & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
R^T \\
0^T \\
1
\end{bmatrix}
\begin{bmatrix}
X_{cw} \\
Y_{cw} \\
Z_{cw}
\end{bmatrix}
\]

Equation 7

This is the full, undistorted, projective mapping using R, T and \( K_2 \).

4.1.3 Distortion model

The classic fish-eye effect introduced by circular lenses needs to be accounted for because it introduces distortion. The lens distortion is modeled as affecting the normalized image coordinates. These are the coordinates sitting on the focal plane with \( f=1 \) and the principal point being \( \text{PP}=[0,0] \). Various models have been proposed but the 6\(^{th}\) order model used by the Camera Calibration Toolbox is described here. \( r^2 = X_{n}^2 + Y_{n}^2 \) where \([X_n,Y_n]\) are the normalized pinhole image coordinates and hence \( r \) corresponds to the distance of an image point from the image principal point PP. The distorted image points are

\[
\begin{bmatrix}
X_d \\
Y_d \\
t_x \\
t_y
\end{bmatrix}
= (1 + d_1 * r^2 + d_2 * r^4 + d_3 * r^6) \begin{bmatrix}
X_n \\
Y_n
\end{bmatrix} + \begin{bmatrix}
t_x \\
t_y
\end{bmatrix}
\]

Equation 8

where \( d_1,d_2,d_3 \) are the distortion coefficients and \([t_x,t_y]\) are the tangential distortion parameters. The tangential distortion is often negligible but it is taken into account and the equation for \([t_x,t_y]\) is
Chapter 4: Camera geometry

\[
\begin{bmatrix}
I_x \\
I_y
\end{bmatrix} = \begin{bmatrix}
2* t_1 * X_n * Y_n + t_2 *(r^2 + 2X_n^2) \\
2* t_2 * X_n * Y_n + t_2 *(r^2 + 2Y_n^2)
\end{bmatrix}
\]

Equation 9

with \( t_1 \) and \( t_2 \) being the tangential distortion coefficients.

The main effect of this distortion is shown in Figure 20. The pixels further from the principal point are pulled further in, resulting in apparent bulging whereas in actual fact the images have contracted. The tangential distortion is so small that it is difficult to observe. If it was more pronounced there would be an apparent shearing effect on the images.

Figure 20: Left and right distortion effects in stereo cameras used in project. Green is original pixel image, blue is left camera distortion, red is right camera distortion

4.2 Epipolar geometry

The following few sections describe the ideas behind epipolar geometry. This theory is needed to be able to understand and implement the line extraction and 3D reconstruction algorithms.

4.2.1 Stereo setup

Epipolar geometry relates how a point in one view is constrained in another view. It is the fundamental basis behind searching for point correspondences and it reduces single line laser scanning to an almost trivial linear search. At the core of epipolar geometry are the essential and fundamental matrices. These matrices take a point in one view and constrain its corresponding position in a second view to lie on a line.
Figure 21 illustrates the basic principles of epipolar geometry. The projection of the optical center onto the other cameras image plane produces an epipole. $O_1$ projects onto the right cameras image frame as epipole $e_2$ and similarly $O_2$ projects as epipole $e_1$ onto the left camera. The points $p_1$ and $p_2$ are the image plane projections of the target point $M$. The line through $e_1$ and $p_1$ is known as the left epipolar line and the line joining $e_2$ and $p_2$ is known as the right epipolar line. The points $O_1$, $p_1$, $e_1$, $O_2$, $p_2$, $e_2$ and $M$ are coplanar so the epipolar lines can also be thought of as the lines of intersection of the plane defined by $M$, $O_1$ and $O_2$ with the left and right camera image planes.

Only three points are needed to define any plane and if $O_1$ and $O_2$ are known, then it is easy to see that $p_1$ will define a line in the right camera’s image plane and similarly if $p_2$ is known then it defines a line in the left camera’s image frame. This leads to the essential matrix.

### 4.2.2 Essential matrix

The essential matrix defines a transformation of normalized coordinates on one camera image plane to a line on the second image plane. It is only useful if the intrinsic and extrinsic camera parameters are known. It is the algebraic transformation of a point to a ray and hence the matrix must be singular because the mapping is not unique as many points map to the same line.

There are numerous derivations of the essential matrix all of which give another perspective on the meaning behind the matrix and how it operates. One of the simpler derivations, but less meaningful is as follows [14]:

![Figure 21: Epipolar geometry of stereo camera setup](image)
Taking the vector $P_2$ and crossing it with the vector $O_2 - O_1$ will produce a vector that is orthogonal to both these vectors by definition of the cross product. However the vector $O_2 - O_1$ is the translation vector $T$ described as one of the extrinsic parameters. The formula $P_2 = RP_1 + T$ is used in place of $P_2$ to produce the equation

$$\overrightarrow{(O_2 - O_1)} \times \overrightarrow{P_2} = \overrightarrow{T} \times RP_1 + T = \overrightarrow{T} \times RP_1$$

\text{Equation 10}

which is an orthogonal vector. A vector in terms of $P_1$ is now available and in order to relate it to $P_2$ the scalar product is taken

$$P_2 \cdot (\overrightarrow{T} \times RP_1) = 0 = P_2^T (T_x, R) P_1$$

\text{Equation 11}

This is all that is needed: a relation between $P_1$ and $P_2$ described by known values, namely $R$ and $T$. Hence the essential matrix is

$$E = T_x R$$

\text{Equation 12}

### 4.2.3 Fundamental matrix

The fundamental matrix is a generalization of the essential matrix and it can be found without knowing the extrinsic and intrinsic parameters. Furthermore it can be used on un-normalised pixel coordinates. Various algorithms exist to find the fundamental matrix from corresponding coordinate pairs. However, in this case it is unnecessary because the essential matrix and fundamental matrix are related by the following equation
\[ F = K_1^{-T} E K_2^{-1} \]

\textit{Equation 13}

which is useful because both \( K_1 \) and \( K_2 \) are known and the essential matrix can be easily derived as shown above.

The fundamental matrix applied to a point in the left image produces a representation of the epipolar line in the right image. Lines in 2-space can be described by \( ax + by + c = 0 \) and in dot product form

\[
\begin{bmatrix}
  a & b & c
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix} = 0 =
\begin{bmatrix}
  a \\
  b \\
  c
\end{bmatrix}
\]

\textit{Equation 14}

This equation shows that one representation of a 2-space line in homogenous form is simply \( L = [a, b, c] \). The fundamental matrix operates by stating that the epipolar line in the second camera’s reference frame is simply

\[ L_2 = FP_1 \]
Chapter 6: Laser line extraction

5 CAMERA CALIBRATION

All cameras view a scene through a lens so as to focus the available light. However most low cost lenses manufactured today introduce some kind of undesirable\(^6\) distortion as described in Chapter 4. Key camera parameters including the viewing frustum and the focal length are not easily determined by mechanical means on low cost, unmarked lenses. These factors mitigate a software based approach to parameter measurement. However, there is readily available software that makes it relatively simple to take a number of calibration images and determine the most likely parameter values.

Many mathematical models are available that model the distortion introduced by camera lenses. This chapter describes the process used for this thesis to take into account distortion and to determine the individual camera parameters of each camera. The global extrinsic parameters are also determined.

5.1 The Camera Calibration Toolbox

The Camera Calibration Toolbox (CCT) used for this project is a Matlab based toolbox designed by Jean-Yves Bouguet of IBM. The toolbox is powerful and easy to use and has a number of diverse functions that have been used extensively throughout this project.

The toolbox has two modes of operation both of which are taken advantage of. These are the ability to

- determine intrinsic camera parameters including lens distortion coefficients
- determine extrinsic rotation and translation parameters between stereo cameras

\(^6\) Some cameras have lenses which intentionally warp an image so as to gather more information such as a 360 degree lens which maps a 360 degree panoramic view onto a circular plane.
Chapter 6: Laser line extraction

Much of the mathematics involves determining homographies\(^7\) within different checkerboard images and is based on solid theory that has been developed over many years of camera calibration techniques. Although the procedure that has been used to implement the CCT is robust and well understood it is not simple and will not be dealt with here.

5.1.1 Determining intrinsic camera parameters

The toolbox requires a set of calibration images to be captured. These images must be of a checkerboard in slightly different poses and at different depths from the camera. Thirty images is specified as being a good data set however after a number of laborious capture sessions it was found that ten was sufficient to produce a good model.

The intrinsic parameters are described below together with the names they have been assigned in this project and in the CCT.

- Focal length: \( f \) \( f_c \) [2 by 1]
- Principal point: \( PP \) \( cc \) [2 by 1]
- Pixel axes coefficient: \( \alpha \) \( \text{alpha}_c \) [scalar]

\[
\begin{pmatrix}
  d_1 \\
  d_2 \\
  d_3 \\
  t_1 \\
  t_2 \\
\end{pmatrix}
= \begin{pmatrix}
  k_1 \\
  k_2 \\
  k_3 \\
  k_4 \\
  k_5 \\
\end{pmatrix}
\]

- Distortion coefficients: \( \text{kc} = \begin{pmatrix}
  k_1 \\
  k_2 \\
  k_3 \\
  k_4 \\
  k_5 \\
\end{pmatrix} \)

Figure 22 shows the calibration images used. At the same time as capturing for the right camera, images were captured for the left camera so that the stereo extrinsic parameters could be determined.

\(^7\) Projective transformations
Chapter 6: Laser line extraction

Figure 22: Calibration images

After having taken the images the CCT loads them and the grid corners must be located. This portion of the procedure is the most time consuming as it requires manual checking. The yellow dots in Figure 23 represent the corners that are hand picked. The red crosses in the left image are automatically placed as starting points for an iterative corner finder. The corner finder has sub-pixel accuracy. These new corners are displayed in the right hand image.

Figure 23: Manual steps

Sometimes the distortion near the edges causes severe disparity between the initial red crosses and the corners they are supposed to be close to. This needs even further manual handling requiring entry of an initial distortion estimate.
Once this has been done with all the images the iterative calibration procedure can be applied. It produced the following output which is used for the rest of the project.

<table>
<thead>
<tr>
<th>Left camera</th>
<th>Right Camera</th>
</tr>
</thead>
</table>
| **Focal Length:** | fc = \[
\begin{pmatrix}
775 \\
771
\end{pmatrix}
\] fc = \[
\begin{pmatrix}
770 \\
768
\end{pmatrix}
\] |
| **Principal point:** | cc = \[
\begin{pmatrix}
317 \\
214
\end{pmatrix}
\] cc = \[
\begin{pmatrix}
349 \\
238
\end{pmatrix}
\] |
| **Pixel axes coefficient:** | alpha_c = 0 | alpha_c = 0 |
| | \[
\begin{pmatrix}
-0.4200 \\
0.2101
\end{pmatrix}
\] | \[
\begin{pmatrix}
-0.4079 \\
0.1633
\end{pmatrix}
\] |
| **Distortion:** | kc= \[
\begin{pmatrix}
0.0042 \\
0.0023
\end{pmatrix}
\] | kc= \[
\begin{pmatrix}
0.0032 \\
0.0008
\end{pmatrix}
\] |

Figure 24: Intrinsic camera parameters

Here the output has been formatted but in actual fact the program provides error estimates as well. The focal length and principal point are measured in pixels and they are given a ±4 pixel accuracy. This project is not focused on improving the accuracy of these measurements but it would be useful to investigate how the camera calibration errors affect the 3D extraction process described in later chapters.

Before continuing, however, it is important to point out that a 4 pixel accuracy range seems quite large but in actual fact it is quoted with a 99% certainty, meaning that it is three times the standard deviation. The true value most likely lies very close to the quoted value.

The CCT computes the re-projection error which is the difference between the coordinate points found by the corner finder and those predicted by the final model.

The medians in Figure 25 show that the model predicts accurately to within a quarter of a pixel. The large values to the right of the graph indicate a few outliers most likely poorly detected by the corner finder.
The CCT provides the option to undistort any image using the distortion model. This is useful if a scene needs to be properly analyzed as it means the geometry of the scene becomes affine. However, the CCT calculates colouring coefficients and indices that are unnecessary for our purposes. This makes the undistortion process somewhat slower than is useful. Figure 26 shows a captured image on the left and its undistorted version on the right.
5.1.2 Determining extrinsic stereo parameters

Determining the extrinsic parameters is as crucial to the success of the LSU as determining the intrinsic parameters. Luckily the process is a lot simpler and faster as it is fully automated. The CCT takes in the full calibration data from running the intrinsic calibration on both the left and right data sets. As stated before, the data captured were pairs of left and right images of 10 different poses of the calibration board. The data sent to the stereo calibration process were the corresponding re-projected coordinate positions.

The algorithm used is that of a gradient descent minimization. It estimates $R$ and $T$ and uses this to triangulate each left-right data pair to get an estimated 3D position. It then re-projects these 3D positions on to the left and right views using the camera projection matrices and determines the re-projection error. It runs through a number of iterations until the re-projection error has been minimized to below a certain threshold and produces the following output

Rotation vector: $\omega = \begin{bmatrix} 0.035 \\ 0.290 \\ 0.016 \end{bmatrix}$

Rotation matrix: $R = \begin{bmatrix} 0.9581 & -0.0106 & 0.2861 \\ 0.0206 & 0.9993 & -0.0320 \\ -0.2856 & 0.0365 & 0.9577 \end{bmatrix}$

Translation vector: $T = \begin{bmatrix} -121.7 \\ 1.8 \\ 14.5 \end{bmatrix}$

These parameters were derived with the left camera optical center and focal direction as defining the world coordinate frame. The $\omega$ vector is a compact and convenient way of representing a rotation in Euclidean 3-space. Its magnitude represents the angle through which to turn and its direction represents the axis about which to rotate. It is related to the rotation matrix by the Rodrigues formula [15].

Figure 27 shows what the setup of the system looks like. The grids represent the points chosen on the corresponding point pairs after $R$ and $T$ were determined.
5.1.3 Deciding on the final rig setup

The calibration procedure is a tedious\textsuperscript{8} process. It requires manually selecting and checking hundreds of points for stereo rigs. The points are simple enough to pick but it is still time consuming. To make matters worse the slightest perturbation of the stereo rig settings can dramatically affect the extrinsic parameters. Also the adjustable lenses were slightly loose so the focal point and principal point could also be disturbed by a bump to the rig. These two factors contributed to many recalibrations and the calibration described in the above two sections was the last one performed. Once it had been done the cameras were firmly fixed and the lenses fixed in place.

The choice not to test the efficacy of different stereo setups was based on the problem just described. It meant that an engineering decision had to be made. If the cameras were placed far apart this would improve the accuracy of the LSU because the point disparity would be amplified while the error stayed the same. However, the further apart the cameras were positioned the more inwards they had to point resulting in mild to severe occlusion of facial features due to viewing angle. Ultimately a tradeoff between accuracy and non-occlusion had to be made. In the end the choice of position turned out to be adequate yielding good accuracy and little occlusion.

\textsuperscript{8}[16] hints that “recent advances in computer vision indicate that we might be able to eliminate this process altogether”
6 LASER LINE EXTRACTION

Laser extraction turns out to be the most crucial element in the operation of the LSU. The accuracy of the system is almost totally determined by the accuracy of the extraction process. When a person looks at a laser line it seems obvious where the line is and appears to be a simple task to determine the position of centre of the line. However the case is not so ‘simple’ for a computer. The blindingly fast parallel processing that happens in the brain is replaced by single step computations in a computer. This means that intuitive ideas of how we extract features from our visual system must be treated with care as algorithms resulting from these ideas are likely to be unfeasible with high order of complexity $O^9$.

For example one might logically say that our vision system uses the position of our eyes, their focus and the focus of elements in an image to determine how far away something is. Of course this is a good analysis but implementing this sort of computation requires a tremendous amount of processing power and demands accurate dynamic models of camera focus behaviour and direction. Clearly this is not a good approach for a simple project such as this one. The next section describes an algorithm that could be implemented in an efficient language.

6.1 Extracting the line

The following sections discuss the procedure for extracting the laser line from an image. Special attention is given to the computational requirements for the procedure.

6.1.1 Design of a laser extraction procedure

For vision systems one of the best ways of identifying efficient algorithms is to imagine a roving eyeball that can only see a few pixels at a time. Calculations are done with these pixels and the results are then used to guide further motion of the eyeball. In this way the eyeball moves around only where it needs to as opposed to traversing a full input image.

This approach could be applied to extracting laser points within an image by having an eyeball that roams around a page until it finds at least one laser point and then uses this point and various
constraints to define where the next laser points are likely to lie. Finding the first laser point could be reduced to a statistical search. For a 640 by 480 image this means that for a single laser point to be found on average 6500 points (Appendix A1) must be looked at before a laser is found, assuming 45 pixels in a laser line on average. In terms of this project a quick study of the data set (Appendix A2) showed that a laser line is likely to be found within a 50 pixel range of column data. This means that a good searching algorithm would use a 100 pixel wide\textsuperscript{10} scanning window once the first laser point is identified. Effectively this means that the search space of 640*480=307200 pixels would be reduced to 6500+100*480=54500 pixels yielding more than 5 times speed improvement over searching the whole window.

It would appear this method could be used to the advantage of this project, however, various aspects of Matlab make it somewhat difficult to do this without incurring a lot of overheads through the use of for loops and if statements. The algorithm could be efficiently implemented in a language such as C++.

6.1.2 Brute force algorithm
The procedure described above was slightly over complicated for the uses here and it was not implemented. The brute force approach appeared to be well suited to Matlab as its implementation favoured the use of vectors. This meant that for each scan image the full image is searched for laser pixels.

6.1.3 The problem of laser point identification
After having chosen the extraction technique the next stage was to decide how to recognize a laser pixel. This initially appeared to be a difficult problem. It can be best stated as a question:

\textit{If the scanner is to be able to be used in varying lighting conditions then how can laser pixels be differentiated from red objects or bright objects?}

Figure 28 illustrates an image of a laser line on a face in normal lighting conditions and in dark lighting conditions

\textsuperscript{10} The scanning window has to be 100 pixels wide because the first pixel found might be on the edge of the window
Figure 28: Two images taken one after the other, the one with the light off, the other with it on

It is quite clear that having the light off makes the laser line the only visible feature in the image and thus reduces the laser point determination to a maximum value search along each horizontal line. However having the light off for a scan is not a viable option. The two images show that the laser line is very bright as it causes considerable sub-skin surface diffusion. These observations lead directly to the solution for the problem of laser point determination.

6.1.4 On board camera pre-processing
There are numerous settings that can be changed on almost all digital cameras. The same is true of the cameras used for the LSU. Ultimately it was the configuration of these settings that yielded the optimal solution for the problem at hand. The settings of the camera for scanning are shown in Table 1.
Figure 29 shows the effect camera settings have on the captured image. The far left image has no features other than the laser line because the laser is the only light present. The middle image shows the same scene but with the light on. The far right image is the image captured with normal lighting.
Chapter 6: Laser line extraction

It can be seen that the middle image is very similar to the one on the left with no lighting. This is almost exactly what is needed to be able to extract the maximum point along the line and hence determine the laser’s pixel coordinates. However, as indicated in the middle picture there is a lighting artifact which is not part of the laser line. Clearly this cannot be included as it is not part of the laser line so a means of ignoring it needed to be found.

6.1.5 The solution to laser point identification
White light in the image is composed of high levels of red, green and blue. However laser light is mainly red. This fact allows for a simple threshold to be done on every candidate laser point. Candidate laser points are all those pixels which have red values above a certain threshold. A candidate laser point becomes a laser point if its blue value is below a certain threshold. Figure 30 shows the final laser identification process in full.

Each line in the image is thresholded according to the red and blue thresholds as described above. Then the maximum red value is taken as the laser line. The sixth image in Figure 30 shows that the points lie on the line laser line.
6.1.6 Accuracy

The main drawback of this method is that it produces laser line coordinates with pixel level accuracy. This means that the accuracy decreases with distance from the camera. This can be resolved by doing sub-pixel laser point detection but this has not been implemented for the project. There are a variety of methods by which this can be done. Active snake contours or quadratic fitting are two methods that could be used but these are not discussed here.

Another issue that might be an explanation for the inaccuracy observed later on is that the laser lines are highly collimated and as such the maximum intensity distribution is close to Gaussian as described by Phong lighting models [17]. We will assume that a small patch of skin, which is the primary target surface, is oriented with the normal at some angle to that of the incident laser light. The Phong lighting model indicates that for highly reflective surfaces the intensity of the distribution changes with the nth power of \( \cos(\theta) \), which is close to a Gaussian distribution, where \( \theta \) is the incident angle of the light to the surface.

This model means that the maximum intensity of reflected light is not guaranteed to point back towards the cameras at the same angle. The extraction algorithm uses the perceived maximum intensity but because there are two cameras there are two orientations towards which the laser light will be reflected back. This means that one camera will perceive the maximum intensity in a slightly different 3D position to the other camera resulting in a slight stereo mismatch.

![Figure 31: Laser pixel disparity resulting from difference in viewing perspective](image)
Figure 31 attempts to illustrate the affect of laser pixel disparity. The black line at the top represents a highly exaggerated laser width. The three circles represent the magnitude of intensity of return at three positions using a Phong lighting model [17]. The warped intensity return distributions are shown as the small red intensity graphs where the perceived maximum intensity is shifted away from the true position of maximum intensity. One might assume that the disparity would get worse the further away the target is but in actual fact the perceived maximum intensity position quickly approaches the true maximum intensity position the further away the target is because there is less distribution warping. One might also assume that a closer target would yield less disparity but the disparity resulting from the perceived maximum intensity position increases faster than that from a change in depth. [18] describes one possible way of eliminating this problem. In chapter 9 this problem is considered as being one of the potential sources of inaccuracy observed in the 3D model that is produced.
7 POINT CLOUD RECONSTRUCTION

Having found the points on the laser line in each image the next step is to create stereo matching pairs to be triangulated. There are a few details that need to be addressed for this to be done correctly but at its most basic level it is quite simple.

Procedure:

- Go through the list of all laser coordinates in the left image
- Use the fundamental matrix to determine the representation of the equivalent epipolar line in the right image
- Search along this epipolar line in the right image to find corresponding laser coordinates
- If any laser point is found along the right epipolar line add this point and the left point that generated the line to a list of stereo matching coordinates
- Upon completion of the stereo matching list triangulate the points to generate 3D point cloud data

The function used to triangulate the 3D points is provided by the CCT and called `stereo_triangulation`. This function takes in distorted pixel coordinates along with R, T, the two intrinsic camera matrices and the distortion coefficients for each camera which means that the matching points do not have to be undistorted. However, this fact does not eliminate the need to undistort the points as will be explained in the next section.

7.1 Finding the stereo matches

The following sections refer to the process applied to every pixel pair to generate the point cloud.

7.1.1 Left coordinate produces right epipolar line
The section on epipolar geometry describes how and why a pixel coordinate in one image produces an epipolar line in the other image. The detail that is left out in that description is that coordinates
in the left image are distorted and need to be undistorted before the fundamental matrix can be used to generate the epipolar line.

The first step is to take the coordinate that corresponds to a laser point and undistort it. Once the coordinate has been undistorted the right epipolar line is generated by $L_2 = Fx_1$. This line is in undistorted format and so the second step is to take the line and it must be distorted for the right image because the full right image is still distorted.

### 7.1.2 Searching along the epipolar line to find a matched point

Once the epipolar line has been distorted it is searched to find a laser point. The coordinates of the distorted epipolar line are stored in a vector making the actual search simple and linear. The program looks for the position of a 1 at all the coordinates in the epipolar vector and then uses the index into the position of the one to extract the corresponding coordinates.

![Figure 32: Purple: Laser points, Blue: current left image laser point, Green: distorted epipolar line, Red: matched right image laser point](image)

Figure 32 clearly illustrates that the effect of distortion is quite significant. Were the green line to be straight, the stereo match would be unacceptably far from the correct point. The red and blue points are now a stereo corresponding pair that can be sent for 3D triangulation.
7.1.3 3D stereo triangulation

There are a number of ways of performing 3D triangulation. The CCT performs a geometric triangulation procedure vectorized for efficient use by Matlab. Another way to perform triangulation is to minimize the reprojection error of the coordinates. Neither of these methods will be expounded on because they were not investigated for this thesis. It might seem odd that the triangulation function which is at the absolute core of the LSU is not described here, however, it must be noted that it was never necessary to alter the code or even inspect it for that matter. It had all the features of an engineering black box; there are well specified inputs and well specified outputs.

The black box notion is very convenient: matching stereo pairs were fed in and 3D coordinates were fed out. Figure 33 shows a 3D plot of a sample point cloud from various viewing angles. Note that the points have no knowledge of one another and are therefore completely independent.

![Figure 33: Four different views of a sample point cloud as viewed from the left cameras reference frame](image-url)
Chapter 7: Point cloud reconstruction

The first thing that is apparent is that although the various facial features are quite obvious to a human eye there is no clear structure. There is no sense of depth other than that given by the bounding box. This fact together with the independence of the points is the basis for an investigation into generating point relationships and forms the material presented in the rest of this thesis.
8 SURFACE RECONSTRUCTION AND VISUALIZATION

3D scans represent the surfaces of objects. These surfaces must be able to be displayed and manipulated if the scan is to serve any useful function. This section details this process.

8.1 The reconstruction-visualization relationship

Chapter 7 dealt with the reconstruction of a point cloud from stereo image data. As pointed out, this data has no structure to it and all the points are independent from one another. In order to be able to visualize the data in a useful way an approximation of the target surface must to be obtained. There are a number of ways in which the point cloud can be converted to a surface format depending on the type of surface and the context.

Any reasonable definition of a surface allows for the surface normals to be extracted at various positions. This is so that the surface can be approximated by small planar patches which reflect light in ways defined by the lighting model. The lighting model renders the patches using the normals to the patches.

Currently NURBs is the most popular industry standard way to represent a surface for high quality models such as those required by the aerospace industry. The name NURBs derives from the notion of Non-Uniform Rational B-Splines which are curves or surfaces that lend themselves to different levels of parameterization. One of the many advantages of NURBs is that they allow surfaces to be represented algebraically and hence surface normals are simple to extract at any point on the surface, irrespective of whether or not there is a real data point there. Unfortunately, constructing a NURBs model from a point cloud is difficult [19] and, although not beyond the scope of this project, was deemed too complicated for implementation within the given time frame.

There are other techniques by which a surface can be represented algebraically such as through the use of radial basis functions. A RBF toolbox is available for Matlab called FastRBF, produced by
FarField Technology, which enables the reconstruction of point cloud data using RBFs. The toolbox produces good surface models but was found to be unnecessarily powerful for the LSU and also somewhat slow.

The easiest and most commonly used method for constructing surfaces is to generate the data cloud via triangulation and produce a polygonal mesh. In the simplest case the polygons are triangles that consist of a set of oriented vertexes. This is the model that is used to represent LSU scan data and it is described in the next few sections. It should be noted that all scans depicted in the figures are of faces because the LSU was specifically designed to scan faces.

8.2 Surface triangulation

The first problem encountered when attempting to construct a polygonal mesh is in determining how the points should be connected to form the surface triangles. This process is often known as triangulation because it involves joining the points together with triangles.

8.2.1 Nearest neighbour triangulation

A naïve approach to triangulation is to connect each point to it’s 3 closest neighbours which results in a clumped, poorly distributed set of connections as shown in Figure 34.

![Figure 34: Nearest two neighbour triangulation](image-url)
The coordinates on this graph are 2D but the same sort of clumping happens in 3D with nearest neighbour triangulation. Clearly a better method needs to be found if a reasonable polygon mesh is to be generated.

### 8.2.2 Delaunay triangulation

The main problem that needs to be solved is to find a good way of connecting points together to produce a natural looking triangulated surface. One way of doing this is to use a method known as Delaunay triangulation (DT). There is a function implemented in Matlab which computes the DT of a set of points and the first line of its help file states that:

```
“TRI = DELAUNAY(X,Y) returns a set of triangles such that no data points are contained in any triangle's circum-circle”
```

![Figure 35: Delaunay triangulation](image)

Figure 35 gives an example of how the triangulation eliminates possible triangles. The left image has a red and a blue circle. The blue circle lies on the points of a valid triangulation because none of the black data points are inside the circle. However the red circle is the circum-circle of three data points for which the grey line triangle is not valid because there is a black data point within the circle’s boundaries. The image on the right shows the final triangulation. There is a natural visual appeal as the triangles seem “fairly” chosen.

### 8.2.3 Ensuring neighbourhood relations

A neighbourhood is a set of point cloud coordinates which are defined to be connected to a point on a point cloud surface. Each point has a neighbourhood. Different definitions produce different neighbourhoods but they should all produce approximately the same surface. Maintaining these
relations causes a complication that needs to be addressed for 3D to 2D mapping if DT is to be used and a smooth surface reconstructed.

Triangulating points in 2D with DT is relatively simple, whereas the same process in 3D is more complicated. This is because n dimensional DT needs n+1 data points to construct a unique shape. For the 3D case this means 4 data points are used to generate tetrahedrons. A tetrahedron cannot be used to model a surface\textsuperscript{11} area so DT seems as if it might not be of any use. However, this is not true since a mapping from surface onto a 2D plane is always possible. DT can then be performed and the resultant 3D coordinate indices used to generate the 3D mesh.

The problem is that for points on surfaces that wrap around such as spheres or any 3D surface that has overlapping folds there is no direct projective mapping that will maintain the neighbourhood relations of the point cloud. There are a number of clever ways of uniquely projecting surfaces onto planes that approximately maintain the neighbourhood relations. One example involves projecting the points onto the surface of a sphere from inside the object. Initially this was considered as a good way of dealing with the 3D LSU scan data as the surface of a face is approximately one side of a sphere\textsuperscript{12}. Ultimately a far simpler and more convenient method was found.

The operation of the scanner is such that when the LSU performs the stop-snap scanning procedure it stripes a unique laser line on the target face every time. All the laser pixel coordinates that are found in the whole scan are stored. No laser coordinate will be the same as any other laser coordinate because only coordinates that are seen by both cameras are retained. This means that the 2D projection that is needed is already available and no extra work needs to be done. Figure 36 shows an image of the laser pixel coordinates found in the scan superimposed on to an image of the target.

\textsuperscript{11} Tetrahedrons can be used to model volume elements otherwise known as voxels
\textsuperscript{12} A better approximation is that of an ellipsoid
The blue mesh in Figure 36 shows the DT of the laser coordinates. The triangles of the mesh are stored as a set of triples where each triple is made up of the indices of the vertices of the triangle being represented. The program code is intentionally structured so that the indices of the 2D laser pixel coordinates are in the same order as the corresponding 3D coordinates which means that the output of the DT is already the triangle description required.
The last step required for creating a useful reconstruction is to determine the neighbourhood relations of each point. This is not difficult to do and can be done very efficiently in Matlab.

The first step is to create a set of unique edges where an edge is represented by two indices. The edges are then ordered by either the first or second index which will result in a row ordered list with the ordered column having increasing indices but many repeats for each index. The second index of each repeated value in the first column is a neighbour of the first index.

8.3 Visualization

Matlab has a number of powerful functions specifically designed for visualising 3D data. These functions are used to their maximum effect in this project so as to illustrate what kind of visualization is possible.

There are a few different ways in which the data can be viewed. Fundamentally what happens is that a colour is assigned to the faces of the triangles based on whatever surface data needs to be represented.

8.3.1 Novel mesh views

The triangular mesh generated can be viewed from a number of novel directions. Figure 37 shows these novel directions with four different colour maps applied.
Figure 37: Mesh views with different colour maps

The images in Figure 37 are the meshes connected to the point cloud that is seen from novel viewpoints as in Figure 33 in Chapter 7. Once again it is apparent that there is not enough information to be able to easily make out facial features. Furthermore the artefacts that are present on either side of the face are somewhat distracting. These artefacts are as a result of the scanner capturing points on the subject’s top.

8.3.2 Texture mapping

The best way to truly represent the scanned 3D structure is to map the colours perceived by the scanner onto the scanned surface. This is achieved by assigning each vertex to have the colour of the left image laser pixel coordinate and then use Matlab’s texture mapping facilities which are part of its visualization suite. Figure 38 shows a number of novel views of a texture mapped surface.
The long edges that were connected to the data points in Figure 37 have been removed because they are unnecessary. This is done by removing all triangles with edges longer than a certain threshold from the triangle structure. The resulting images coupled with the far more useful texture information provide an effective visualization. Proof of this is that the facial features are easily identifiable by eye.

8.3.3 Lighting model
Viewing the mesh with a lighting model is another highly effective visualization technique. In this manner facial ridges and features are highlighted by virtual lighting from a virtual light source. The colour map of the face is set to white and then the lighting suite in Matlab is used to cast rays onto the face. Figure 39 shows this technique put to use.
The top three images illustrate how different lighting conditions can highlight various facial features such as the nose, the eyebrows and the mouth.

The lower image in Figure 39 leads straight on to the next chapter. This is because the important features in this image are not those of the actual face but more the bumpy surface elements of the face which highlight the inaccuracies inherent in the LSU’s scanning abilities.
Chapter 9: LSU accuracy

9 LSU ACCURACY

The LSU was specifically designed to be low cost. This meant a low cost laser was used for the laser stripe. As mentioned in the chapter on laser line extraction it is primarily the accuracy of the extraction procedure that affects the accuracy of the whole scan. At its core this inaccuracy is caused by uncertainty in the laser line position as a result of the width of the laser stripe. A more expensive laser with a higher quality cylindrical lens would yield a thinner and brighter laser beam. The pixel level extraction is also a major contributing factor to the inaccuracy of the scanner.

9.1 Ground truth from the CCT

No other laser scanners were available at the time of writing and as such a direct comparison between the LSU and another scanner could not be made. The next best option was to use the CCT to provide ground truth measurement as it provides sub-pixel accuracy which is therefore guaranteed to be more accurate than pixel level detection.

9.1.1 Obtaining data for ground truth comparison
The calibration procedure described in Chapter 5 used ten images from each camera. The calibration poses were set up in a stable fashion on a desk where they could not be easily disturbed. Once an image had been taken an LSU scan of the pose was also done. This meant that a comparison between the 3D reconstruction of the pose by the CCT and the laser scanner could be performed.

9.1.2 Comparing the CCT and the LSU reconstructions
Once the 3D data points from both methods had been obtained the points were plotted as shown in Figure 40.
Chapter 9: LSU accuracy

The red points are those generated by the CCT and the green polka dot shaped surface is that produced by the LSU. The reason there are large holes in the LSU generated surface is because the black squares on the checkerboard do not reflect sufficient laser light to be detected. A visual inspection of the images in Figure 40 shows that the LSU surface is somewhat bumpy but appears to lie on the same plane as the CCT data points.

A more quantitative measure of accuracy can be made by fitting a plane to the CCT points and specifying the accuracy of the LSU as the mean distance of all the points from this fitted plane. Figure 41 shows the normal to the plane defined by the CCT points.

The equation of a plane is
Chapter 9: LSU accuracy

\[
ax + by + cz + d = 0 = \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}
\]

Equation 15

However, to simplify this a bit the all the points were translated so that one of the CCT points was set as the origin resulting in \(d=0\). The normal vector \(n=[a,b,c]\) has magnitude 1.

In order to determine the shortest distance from a point to a plane the following equation based on the scalar product is used:

\[
d = n \cdot X = \begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = |X| \cos(\theta)
\]

Equation 16

Having calculated the distances to the plane the standard deviation can be found so as to verify whether or not the data is normally distributed and to determine the accuracy. The standard deviation was found to be 1.98 pixels for the first data set tested. Table 1 compares the number of elements that lie within the range of the standard deviation to those predicted by a Gaussian model of the distribution.

<table>
<thead>
<tr>
<th>Multiples of standard deviation</th>
<th>Percentage of distances within standard deviation</th>
<th>Percentage predicted by Gaussian model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>66</td>
<td>68.2</td>
</tr>
<tr>
<td>2</td>
<td>95.8</td>
<td>95.4</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>99.7</td>
</tr>
</tbody>
</table>

Table 2: Comparison of Gaussian distribution model with actual model

Table 1 indicates that the distribution is very close to Gaussian. The distribution of the graph in Figure 42 further supports this suggestion.
For a true Gaussian, the mean and the median have the same value. The mean distance to the plane is 0.098mm and the value of the median is 0.10mm. These are close which is further evidence that the distribution is Gaussian.

### 9.1.3 Interpretation of the Gaussian error distribution

The data presented above suggests that the distribution of the distance, otherwise known as the error, of the LSU points from the plane on which the CCT points lie is Gaussian. This means that there is almost random noise entering the laser extraction process for this data set. It also means that if the noise can be eliminated entirely then the model will be accurate to within 0.1mm of the actual surface. The process of mesh smoothing is one way of reducing the noise and will be dealt with shortly.

Modelling the distribution as being Gaussian allows for a statistically reliable accuracy measure. For the LSU scan data used to obtain the above information, every point can be quoted as being accurate to within 4mm with 95% certainty and accurate to within 6mm with 100% certainty.
Chapter 9: LSU accuracy

9.2 Accuracy as a function of mean distance from left camera optical centre

A more useful assessment of the accuracy of the LSU can be made by taking into consideration the mean distance of a target from the scanner. The same procedure as described in the above section is followed for all the calibration poses. The poses were then ordered according to how far away their centroids’ were from the left camera’s optical centre. Figure 43 shows the results of this study.

Figure 43 holds useful information about the general relationship between mean target distance and accuracy. The thin red and blue lines are the median error and mean error respectively. They lie almost on top of each other helping to substantiate the claim that the distributions are Gaussian throughout the scans. The thin green line is the standard deviation. A visual inspection of these data points suggests that they can be reasonably well modelled by straight lines. The thick green
Chapter 9: LSU accuracy

line models the standard deviation while the thick red line models the median distance to the plane which represents the scanner's error.

The thick green line representing the modelled standard deviation is as one might expect. The further away the target the greater the inaccuracy of the reading.

The thick red line is somewhat more difficult to analyze. One possible explanation is that it represents the average behaviour of a constant error offset or bias as a function of distance from the camera. As mentioned in the Chapter 6 on laser line extraction the difference in viewpoints of the two cameras means that there will be a slight disparity in the perceived intensity of the laser line that is captured by each camera. This small disparity will be larger for laser points closer to the cameras as the angle at which the cameras perceive the same line increases closer to the cameras as illustrated in Figure 44.

![Figure 44: Closer points are perceived at larger angles](image)

Figure 44 does not indicate the disparity but simply shows how the perceived angle changes the closer the laser line is to the cameras. Essentially the disparity causes an offset or a bias which should result in an overestimate or underestimate of the true 3D data point. Therefore the closer the laser point to the optical centre, the greater the error offset.

In actual fact this explanation would require a nonlinear fit to the data and a hyperbolic function would seem more appropriate but there is not enough data to do proper trending and the straight line at least shows the general behaviour over the region plotted.
Chapter 9: LSU accuracy

Even if the above explanation is wrong, the trend at least shows that the closer to the camera a target is the bigger the error bias or dc error and the further away the target is the smaller the offset but the greater the impact of noise.

9.3 Data smoothing

Because the scanner is known to be inaccurate, it makes sense to perform some kind of smoothing function on the extracted data. Ideally this should reduce the standard deviation, increase the accuracy and create a smoother surface.

9.3.1 The smoothing process

It is not difficult to perform smoothing on data that has known neighbourhood relations. There are a number of different techniques but the simplest one involves moving each data point a small way towards the centroid of its neighbours. Unfortunately this method has a drawback in that centroid directed smoothing causes a contraction of the data points as a whole. The reason for this is that the nature of smoothing is to pull things together. The points on the edges will move furthest from their starting point as there is no counter force pulling them back out.

The smoothing function used for this project results in mesh contraction and therefore in order to perform smoothing without causing too much distortion of the initial scan many iterations need to be done with small scaling factors. The scaling factor simply dictates how far along the vector from the point to the neighbourhood centroid the point must move.

Figure 45: Different levels of smoothing
Figure 45 shows the effects of different levels of smoothing. The first image shows the scanned surface. The second image shows 20 iterations of smoothing at 0.1 scaling. The third image was iterated 40 times at 1.0 scaling and the last surface was iterated 500 times at 1.0 scaling. Clearly the second image is the most suitable as it shows good smoothing and well defined facial features.

The last image in Figure 45 illustrates the dramatic contraction effect of over-iteration. One method of preventing mesh contraction is to perform an expansion after smoothing. This involves making the scaling factor negative which pushes the points outwards. An investigation into contraction followed by expansion has not been done.

9.3.2 The effect of smoothing on accuracy

When smoothing the data the effect should be predictable. Firstly the standard deviation should be significantly reduced. This is due to the fact that the noise inherent in the measurements has been reduced which will obviously result in less deviation from the true value. Secondly, the bias that is part of the scan data will most likely not experience much change. This is because the impact of pulling points towards their neighbours will be evened out and no net effect on the average bias should be apparent.

The LSU data from the calibration poses was smoothed over 30 iterations with 0.1 scaling and the accuracy analysis graph produced is in Figure 46.
The predictions seem to match the data well. The thick green line has been shifted down and the gradient made less steep indicating reduced standard deviation. The thick red line on the other hand has stayed in almost exactly the same position.

9.4 Comments on accuracy

Smoothing is clearly necessary both from a visualization point of view and from an accuracy point of view. The accuracy of the system is seen to be better closer to the cameras and, after smoothing, can be quoted to be accurate to within 0.9mm\(^\text{13}\) when around 30cm from the rig. There is around a 0.5mm offset bias at this point so the true accuracy of a scan at 30cm should be quoted to be within 1.4mm. The data suggests that a scan at around 30cm is optimally accurate for the LSU.

\(^{13}\)3 * std dev = 3 * 0.3 = 0.9mm
10 GENERAL 3D RECOGNITION

People recognize faces with the greatest of ease. The parallel processing neural networks in human brains have specific architectures for picking out facial shapes in almost any environment. Studies have shown that people are able to identify between 2 and 4 faces in less then a fifth of a second [20]. This means that people are capable of analysing and identifying 16 people a second. The kind of processing that is going on merely to be able to identify a face exists in a field of view is staggering. With this in mind it is not surprising that facial recognition is a challenging task to implement within a computer. This is because in order to replicate the recognition capabilities of people, a computer system has to produce results invariant under a variety of lighting conditions, expressions, poses and target distance [21].

10.1 2D recognition

The majority of recognition systems currently available rely on flat 2D image inputs. These can be obtained by digital video camera or normal digital camera. There are two different approaches to facial recognition [21]. These approaches are

- feature based
- image based

Feature based recognition relies on a program’s ability to extract facial features and produce a compact representation for comparison with other faces in a database such as a difference vector from a template.

Image based recognition uses the various averaged properties of the image of the same face under different conditions to produce some kind of generalized image representation of a face. Currently one of the most widely used aggregation methods is the construction of eigen-faces via principal component analysis. In this method a large set of faces is recorded and an eigen-space of faces is generated. Any face presented to the system is then reconstructed from a weighted vector of the
eigen-faces in the database. The eigen-face weight vector can be used to reduce the search space, and all faces within a given distance metric of the weight vector can be reconstructed and compared to the initial image [22].

It turns out that a combination of feature and image based techniques combined provide even more robust recognition by using eigen-face descriptions as well as eigen-features such as eigen-eyes and eigen-noses.

All systems have different conditions under which they operate maximally and most flat image systems work best when a face is presented to them under non-extreme lighting conditions and with a face forward orientation. For applications where non-intrusive recognition, automated recognition is required a face forward pose for a camera is not very likely. This fact and the fact that lighting conditions can seriously reduce the recognition rate of many state-of-the-art systems have led researchers to consider lighting and pose invariant techniques. Systems based on 3D scans of a subject provide a way to eliminate lighting and pose effects.

**10.2 3D recognition**

Obtaining and representing the 3D data to be used is the first step towards recognition. The specification of this step is important as it constrains the way in which the recognition system will operate on the data. As with 2D recognition, 3D systems can use either feature or eigen-space based classification methods or both combined.
10.2.1 Direct feature based recognition

There are many different ways to define facial features. Studies [23] have shown that the bi-lateral symmetry of a face greatly assists people in the process of recognition. Symmetry therefore is one possible metric by which to measure a face. Identifying the bilateral symmetry axis has also been used to assist the process of recognition in both 2D and 3D data. For example the authors of [23] developed an algorithm which extracts the bilateral symmetry axis and then determines the symmetry profile which is defined as being the intersection of the facial surface and the bilateral symmetry plane. This profile is then used as a facial description vector as it is considered unique for faces with specific expressions.

![Bilateral symmetry axis and symmetry profile][1]

For feature measurements an intuitive feature set is comprised of the prominent elements of a face including the nose, lips, eyes, eyebrows, forehead, cheekbones and chin. If the positions and sizes of these features are known then a face can be compactly and comprehensively described by a short biometric description vector. This vector can be matched to a database of other facial vectors for identification.

If the requirement of a meaningful feature set is relaxed then various mathematical operators can produce feature sets. [24] presents one such way of doing this. An initially unstructured point cloud is fitted with a NURBS surface and the knot vectors which describe the surface are then pruned and used as feature vectors.

---

[1] Meaningful in this context means that a person can comprehend the measured quantities as reflecting some real world observable property.
Unfortunately, the process of feature detection in the majority of unconstrained scanning environments is far from trivial. A minimally constrained polygon mesh without a known orientation is difficult to search to find facial features in a computationally feasible manner. The reason for this is that changing the orientation of a face means that the 3 degrees of freedom available for rotation make the search space absolutely massive.

10.2.2 Eigen-space based recognition

Eigen-space based recognition works identically to eigen-space methods in 2D except the data is derived from a 3D scan process and then a conversion process produces a 2D image. [25] describes an approach that converts an unstructured point cloud to a mesh. The mesh is then registered to a known template mesh using the iterative closest point algorithm (ICP) proposed by Besl and McKay [26]. Once the mesh is oriented it is viewed face on and a 2D depth map is created and passed to the eigen-space generating algorithm or any other 2D based algorithm to be used.

![Figure 49: 3D projection from depth map and resulting principal eigen-faces [25]](image)

Every new face is processed in the same way and its eigen-vector description is matched to a database of other faces. The results presented by the authors of [25] are telling. The best recognition rate for the particular implementation of 3D eigen-space based recognition was 94%. However, this was achieved with manual registration as opposed to using the ICP alignment algorithm which produced a best recognition rate of 80%. This is good but not as good as state-of-the-art 2D recognition systems which operate with around 99.9% accuracy under controlled conditions.
A depth map is not expression invariant for changing facial expressions as the depth contours will change for different expressions such as when someone is smiling or frowning.

10.2.3 Acknowledging the need for accurate registration and estimation of orientation

The general trend with recognition systems is that manual registration processes produce considerably better recognition rates than the automated registration equivalents. The problem of ascertaining a subject’s pose is now seen to be a problem in both 2D and 3D systems. 3D systems, however, contain more data about the pose and orientation of a subject. Therefore, if there is a means of finding the orientation of a face very accurately then the recognition rate of a 3D system can be greatly enhanced. This illustrates the fact that the registration and detection of facial orientation needs to be effectively applied before any other processing can take place.

10.2.4 Orientation and expression invariant recognition

An ideal recognition system would be both orientation and expression invariant. Although this might sound like an unattainable requirement there is in fact good evidence that it is possible. It has been shown [27] that there are a variety of transformations that can yield expression invariant images within a certain range of expressions. A technique known as spherical embedding has been proposed in [28].
In this method the texture map or depth map obtained from a 3D model is mapped onto the surface of a sphere yielding a 2D mapping with spherical coordinates. This process only requires two control points which are relatively easily identified. The tip of the nose and a point directly behind the plane of the surface model of the image are used for the transformation. These two control points are used to define a radial vector and a radial distance is chosen. This then defines the sphere on to which the data is mapped. The 2D image can be processed by standard methods. However this once again introduces the problem of orientation although the problem has been reduced to a single degree of freedom search as the only orientation now necessary is that of spherical rotation of the spherical mapping about the radial vector defined previously.

The authors of that paper propose a clever but simple method by which the need for this orientation is eliminated. The method involves taking the magnitude of the spherical harmonic components of the mapping. This is equivalent to doing a 2D Fourier transform which will be orientation invariant. The reason for this invariance is that the phase of the harmonic spectral components is linearly related to the angular rotation of the data in the same way the phases of harmonic coefficients of a linear signal are related to the position of the signal in time or space.

The authors report a recognition rate of around 98% which is impressive. The material is only a few years old with the first presentation of the approach being in 2003. Future improvements are expected.
11 STEPS TOWARDS RECOGNITION VIA SPHERICAL EMBEDDING

The scope of this project was primarily limited to the construction and testing of the LSU together with reconstruction of scanned data. However there was some time available after these tasks were completed and so the low priority objectives of face recognition were tackled. Instead of trying to develop a full face recognition system the idea was to illustrate the conceptual steps towards recognition.

Spherical embedding was described in the previous chapter and appears to be a very promising form of 3D face recognition. The pre-processing steps are not particularly difficult to implement and so this section describes the first few steps towards recognition via spherical embedding.

11.1 Finding the nose

The problem of detecting the orientation of a face has been highlighted as being a major factor in developing robust recognition systems. As proof of this fact many systems perform better when the orientation has been done manually thus illustrating that automatic orientation introduces error and causes a reduction in recognition rate. Further proof is in acknowledging the large number of papers produced each year dedicated specifically to determining and tracking the facial pose of a subject. Only three points are needed to uniquely define a facial orientation. Detecting the tip of the nose is a good starting point.

11.1.1 A brief description of singular value decomposition

Singular value decomposition (SVD) is one of the most powerful tools available for determining the fundamental structure of dimensional data [29] and it is most useful in higher than three dimensions where determining the structure of data becomes less intuitive.

The basic SVD theorem is as follows:
Every [m by n] matrix can be written as the product of three matrices in the form
\[ A = USV^T \]

where \( U \) is an \( m \) by \( n \) matrix with the property that \( U^T U = I \), \( S \) is a square diagonal \( n \) by \( n \) matrix of singular values and \( V^T \) is a square \( n \) by \( n \) matrix of orthogonal basis vectors also resulting in the property that \( V^T V = I \). The factorization of \( A \) into \( U \), \( S \) and \( V \) can be done relatively efficiently so algorithmic complexity is not of concern, unless the matrix being factorized is inordinately large.

The interpretation of the factors is as follows. The unit magnitude, orthogonal vectors in \( V \) down the columns represent the principal directions of the data and form an orthogonal basis. The singular diagonal elements of \( S \) represent the weights of these vectors and can be thought of as axis scaling elements. The \( n \) dimensional row vectors of \( U \) are the data elements of \( A \) but with coordinates relative to the orthogonal, scaled reference frame defined by the columns of \( V \).

### 11.1.2 Description of the nose finder

The most prominent feature on a face is the nose. It is also the most central feature which means that if some or all of the points on a nose can be determined then it is possible to search for other features such as the eyes based on an aggregate template of a face.

A quick survey of the literature indicated that because the nose is the most prominent feature it is often the easiest to detect. The main problem involved in determining the position of the tip of the nose is that some people have foreheads that are further off the facial plane than the nose resulting in incorrect classification of the nose tip. To solve this problem one needs to try eliminate the possibility of including the forehead in this search. With this in mind the following procedure is followed to find the tip of the nose:

1. Find the centroid of the facial surface.
2. Create a reduced point cloud of all the points within a sphere of a certain size with its center at the centroid.
3. Use SVD to determine the principal directions of this reduced point cloud.
4. Use the maximally weighted principal direction to define a normal to a plane going through the centroid.
5. The furthest point from this plane in the reduced point cloud should be a close approximation to tip of the nose.
Figure 51 shows the process by which the tip of the nose is extracted. The red points are those that are within the radius of a sphere centred on the centroid with a threshold radius. Note that only a small portion of the forehead is included in this point set. If a smaller threshold radius had been used points on the forehead would have been totally excluded. The green points are a small set of points found on the nose within a certain distance of the surface defined by the normal produced by SVD going through the facial centroid. The yellow points are the projection of all the red points on to this plane and are simply there to show the plane relative to the nose points. The blue point is the detected tip of the nose and it is made visually apparent that the blue perpendicular line from the plane to the blue point is the longest of all such possible lines. Note that even though the forehead has been included, the plane is oriented such that the tip of the nose is still correctly determined.

11.2 Spherical embedding

Processing a point cloud for spherical embedding is relatively easy. The steps are as follows:

- Determine the position of the tip of the nose.
- Find a radial vector that is approximately normal to the face and that goes through the tip of the nose.
- Find a useful radius and map the point cloud onto the surface of a sphere defined by the radial vector.
Chapter 11: Steps towards recognition via spherical embedding

- Convert the mapping into 2D parametric coordinates.

Once the last step has been completed the data would be FFT’d and processed by a recognition algorithm to determine the closest fit to a face in a database of similarly processed point clouds.

11.2.1 Finding the radial vector through the tip of the nose
Having found the tip of the nose a reasonable radial vector needs to be defined.

One way of doing this is to find the normal to best fit plane through the points thresholded within a spherical radius of the nose tip. If a head is modeled by an ellipsoid, the normal vector of this plane will be approximately parallel to one of the two minor axes of the ellipsoid. The reason for this is that the points around the nose tip are approximately evenly distributed on the ellipsoid surface.

![Figure 52: Ellipsoidal model of a face](image)

The purple arrow represents the ideal radial vector and the gray plane represents the ideal facial plane.

11.2.2 Mapping the point cloud onto a sphere
Having defined a radial vector, a center for the sphere on which to map the face must be decided on. This can be done by translating the point cloud along the radial vector for a certain distance and then using this distance as the radius of the sphere. The actual choice of the radius must be empirically found through testing the change in recognition rate for different radial lengths. Once the mapping has been performed all the points can be normalized onto a unit sphere.
Figure 53: The normal derived from an actual point cloud

Figure 53 shows the normal found by fitting the real data points in red to a plane and using this normal fitted to the tip of the nose to define the radial vector. The black circle represents the sphere onto which the points will be mapped.

Figure 54: Spherical mapping

The first image on the left in Figure 54 is the front on view of a 3D scan and the other images are the projections of this scan onto a sphere as specified by the process just outlined.

11.2.3 Conversion to 2D parametric coordinates
Converting the points on the sphere to parametric coordinates can be done by considering two angles. These angles are essentially the longitude and latitude relative to the radial vector though the nose.

To calculate the coordinates is somewhat tricky. An orthonormal basis must be defined with the radial vector as one of the reference directions. Any other arbitrary point on the sphere can be crossed with the radial vector to produce another orthogonal direction and the 3rd orthogonal vector is computed from the first two. Once the orthogonal basis vectors have been found all the points
are transformed into the reference frame defined by these vectors. The reason for doing this is that it makes projecting the points onto the plane defined by the radial vector as simple as setting that component to 0 because it is now a unit vector of the form \([1,0,0]\).

![Figure 55: Measurement of spherical coordinates](image)

Determining the coordinates is now a matter of finding the elevation angle and the rotation angle as shown in Figure 55.

![Figure 56: Parametric axis mapping](image)

Figure 56 shows the final parametric mapping onto a 2D plane. The physical interpretation of the image goes against ones intuitive understanding of space. The bottom row represents all the points on a circle directly on top of the nose. Each row moving on up represents a projected circle with an increasing angular displacement from the radial vector.

### 11.2.4 Recognition

The next step is to provide a 2D recognition algorithm with the magnitude of the harmonic coefficients of the image. This step has not been implemented as a result of insufficient time.
12 CONCLUSIONS, RECOMMENDATIONS AND FUTURE WORK

12.1 Conclusions
Based on the observations made throughout this thesis the following conclusions have been drawn.

12.1.1 High level objectives met
New material on camera geometry, epipolar geometry and interfacing camera hardware in Matlab were all studied for the first time and effectively applied to produce a functional scanning unit thus illustrating an ability to use previous knowledge and apply new material.

12.1.2 LSU design followed specified objectives
The decision to use a scanning laser platform with stereo cameras was justified through a comprehensive review of all available techniques, the cost and availability objectives and the time available for completion of the project. The review of the current state-of-the-art scanning devices and methods indicated that there are considerably better scanning techniques available but they are more expensive or more conceptually challenging.

The cameras used for the project are generic as is the method by which they interface with the computer. USB cameras can be attached without any change in functionality but with the potential for lower cost by eliminating the need for installing a FireWire port.

The LSU is portable and highly configurable with all specifications based on some specific design criterion.

12.1.3 Cameras used not fast enough
The computer interface to the cameras is not fast enough. Continuous scanning is faster then stop-snap scanning but is not possible with the cameras used for this project. Simultaneous triggering of the cameras at 30 fps was found to be accurate to within 50ms which is more then the length of time between frames resulting in unacceptable point correspondence errors.
12.1.4 Camera calibration too slow and sensitive to disturbances
The procedure used for calibrating the cameras so as to determine the extrinsic and intrinsic parameters is too slow. Furthermore slight disturbances of the camera setup results in distortion of the model thus requiring recalibration.

12.1.5 Laser extraction possible within a general environment
The environment in which the LSU was used was representative of the conditions under which the scanner could be used. A simple method for extracting the laser line in the presence of lighting artefacts was found and shown to work correctly.

12.1.6 Camera pre-processing reduces software oriented computational requirements
The cameras used for the LSU allow for on-board image processing. This was affected by changing the internal capture settings of the cameras. As a result the computational requirement of the laser line extraction software is reduced.

12.1.7 Laser line extraction affects scanning accuracy
The extraction of the laser line was found to seriously affect the scanning accuracy of the LSU. Further research into accurate sub-pixel laser detection is needed.

12.1.8 Point cloud reconstruction contains no immediately meaningful surface information
The information generated by the stereo reconstruction algorithm produces unstructured point cloud information. Apart from residing in the same Euclidean reference frame, the extracted 3D coordinates have no meaningful relation to one another and require processing in order to determine their surface level relationships.

12.1.9 Delaunay triangulation is effective within the context of the LSU
It was determined that DT could not be used in the general context of 3D triangulation but is effective within the context of the LSU. This conclusion is based on the fact that no apparent overlapping of surface elements appears to take place during visualization.
12.1.10 **Scanning accuracy decreases with distance from the camera**

Based on the linear graphs describing the scanning accuracy in chapter 9 it can be concluded that the accuracy decreases linearly with distance from the camera. This is because of noise present in the extraction process that is as a result of pixel-level detection.

The optimal position of a target to be scanned is around 30cm from the center of the stereo rig meaning that the effective scanning volume is perfectly situated for scanning faces thus eliminating background data.

12.1.11 **Smoothing improves scanner accuracy**

Iteratively smoothing data using the centroid directed neighbourhood relations results in increased accuracy. The smoothing process is seen to be effective at reducing the noise injected by the extraction process. On the other hand over-smoothing decreases scanning accuracy due to surface contraction.

12.1.12 **Detection of facial orientation for a 3D recognition system is critical to the success of the system**

A review of the literature on recognition technology indicated that, in general, state-of-the-art 3D recognition systems perform at a lower recognition rate than their 2D counterparts. It appears that the automated detection of facial orientation results in reduced recognition rates indicating that the techniques used need to be improved.

12.1.13 **Spherical embedding yields orientation invariant recognition capabilities**

The nature of spherical embedding is to reduce the degrees of freedom of facial orientation from three to one. Further processing can eliminate this degree of freedom resulting in an orientation invariant representation of a face.

12.1.14 **Simple nose tip detection is possible**

The detection of the tip of the nose by the procedure described in Chapter 11 means that complex detection methodologies are unnecessary. The detection process is dependent only on the distribution of the scanning data that produces the centroid to be used.
Chapter 12: Conclusions, recommendations and future work

12.2 Recommendations and future work

The material presented in this thesis has all been of a very general nature. Each chapter deals with a specific aspect that is common to all 3D laser based scanning systems. Ultimately this means the scope for future work is unlimited. Each and every module can be researched further and improved on.

12.2.1 Improve the scanning system
As it currently stands the scanning system operates with a linear platform and two cameras. There is no positional feedback from the platform which is one of the advantages of using uncalibrated laser extraction. However a better scanning system would implement both calibrated and uncalibrated 3D laser extraction.

The two cameras used for the project produce a lot of information that is discarded. Even when there are laser points detected in one image the same laser points may be occluded in the other image. With a calibrated laser extraction system only one camera is needed to generate the points because the geometry between the camera and laser plane is known. The clear advantage of this technique is that for the points which are not visible to one camera they will be visible to the other meaning the information isn’t lost. When they are visible to both then there is a triple level of redundancy which can be used to increase the accuracy at the given point.

Another improvement to the scanning system would be to change the linear platform to a rotational one with special attention to ensuring accurate positioning.

12.2.2 Improve calibration setup and procedure
Improving the calibration process is necessary for future development of the LSU. One technique for doing this is by using positioned laser points on a canvas in the viewing space of both cameras. If a calibrated rotational scanner is used to generate the laser points together with a sub-pixel laser detection algorithm then the whole calibration procedure can be fully automated. Once the intrinsic camera parameters are known the extrinsic parameters can be easily determined by doing single shot laser extraction for both cameras resulting in almost exact position correspondences. These correspondences can then be used with the 8-point algorithm to calculate the fundamental matrix.
12.2.3 Produce an active stereo-vision system
A considerably more advanced vision system could be constructed with active camera positioning. The cameras can be placed on precision control servo motor platforms and calibrated to move together or independently. If cost was not a primary concern then motorized focus cameras could be used allowing for depth-from-focus 3D reconstruction. The system would provide an advanced platform for research and prototyping of different stereo methodologies. Depth from motion or tracking under control of neural networks are possible research areas.

12.2.4 Build a full 3D recognition system
The work already done for this project means that a full 3D recognition system is a feasible plan of action. The scanner should be improved and the improved scans can be used for creating a full 3D recognition system. Spherical embedding should be used as a starting point because it is relatively simple to implement.
13 REFERENCES


Chapter 13: References

Last Access: 18 October 2005

[15] Wolfram Research, Rodrigues Rotation Formula,


[18] Brian Curless and Marc Levoy, “Better Optical Triangulation through Spacetime analysis”, 1996,


http://www.siam.org/siamnews/04-03/face.htm, Last Access: 18 October 2005

[22] Photobook/Eigenfaces Demo, 2003,

[23] Liyan Zhang1, Anshuman Razdan, Gerald Farin, John Femiani, Myungsoo Bae, Charles Lockwood, “3D Face Authentication and Recognition Based on Bilateral Symmetry Analysis”, National Science Foundation, 2003


Figure 57: Graph illustrating how many pixels are found in a laser scan for each scan line. The peak in the middle is because the center of the image is the most visible to both cameras. For the scans on the sides there are much fewer pixels. The average is 47.

If there are 640*480 pixels in an image then the best probability of randomly finding a pixel is $P_{pixel} = \frac{90}{(640*480)} = 2.93e^{-4}$. The best case only happens once and on average the probability will be half that. This means on average 1 in 6500 pixels are laser points.

Figure 58: Sample data set representing the distribution over which the horizontal data points are found in the scan. The biggest range comes near the middle where the profile of the nose causes a wide range of x values and y values.