UNIVERSITY OF CAMBRIDGE
DEPARTMENT OF ENGINEERING

A 21st Century Laboratory Testing Device for Geotechnical Engineering

Author:	Suzie Neave
Supervisor:	Sam Stanier
Date:	24th May 2020

I hereby declare that, except where specifically indicated, the work submitted herin is my own original work.

Signed Neave *date* 24/5/2020

A 21st Century Laboratory Testing Device for Geotechnical Engineering - Technical Abstract

Suzie Neave

24th May 2020

1 Introduction

This project covered the initial design process for a new device for geotechnical testing.

Triaxial test rigs are a commonly used apparatus in commercial laboratories for soil characterisation. A large amount of the data gathered from these tests is however not representative of the soil mechanisms occurring inside the cell.

Shear bands form during triaxial tests because of instabilities in strain and are impossible to prevent in even the cleanest of triaxial tests. Beyond the point of their formation, the triaxial test loses its validity as an element test as strains begin to occur non-uniformly.

Triaxial tests provide only a limited view of the strains occurring within their cell as they measure average axial and volumetric strain. Beyond the formation of shear bands, this information has little relevance as the vast majority of strain is concentrated at the shear bands. It is necessary therefore to identify the point at which these bands form as data beyond this point should not be used for soil characterisation.

2 Design

The new device was designed to measure strains more accurately than previous attempts which all relied on 2D data for measurements. The device incorporated cameras surrounding the test cell to capture sample particle displacement over time. 3D particle displacements could then be calculated from the multiple camera views and used to create a 3D model of sample deformation. Strain values could then be calculated from the model.

The device was developed with the aim of being low cost and operable without specialist knowledge. Three elements of the device design were developed. The camera mount, the control system and the calibration procedure.

The camera mount was designed to hold 36 Raspberry Pi V2 cameras against a triaxial test cell wall. The mount design was in the form of CAD file which could be 3D printed as required.

The control system allowed the calibration setup and testing procedure to be run from one computer with a number of simple terminal commands. The system required adequate synchronisation in camera operation to allow 3D reconstruction of the sample. An LAN was used to connect to the Raspberry Pis and commands were broadcast to the entire LAN network. The synchronisation was tested by pointing the cameras at a common stopwatch and comparing the capture time.

A calibration procedure was required in order to minimise the distortion caused by light refraction through the triaxial cell. This was necessary in order to provide accurate 3D reconstruction. The procedure made use of a 3D target. A Charuco and Circuco calibration procedure were tested and refined for calibration accuracy.

3 Results

The synchronisation indicated an expected range in capture time of 0.768s. Calibration produced an optimum result of ± 0.852 RMS error. When projected to 3D, these errors produced a minimum discernible particle displacement of between 0.029mm and 0.042mm depending on the strain rate of the triaxial test.

4 Conclusions

The limit on synchronisation was caused by interference within the 2.4GHz band. This may have been improved using an LAN operating on the 5GHz band.

Calibration precision was limited by the focus of the Raspberry Pi cameras. This prevented corners from being accurately identified by the computer vision algorithms when using Charuco board calibration. This issue was not present with the Circuco boards however they introduced their own inherent error due to circle projection. This error was larger than that caused by in-accurate corner detection, therefore the Charuco boards provided a more precise calibration overall.

Comparing with recent experiments measuring strain within shear bands, the precision of the device is expected to be more than capable of detecting shear band formation in sands and just capable of detecting shear band formation in clays. The design of this device is unlikely to produce sufficient precision measurements to measure small-strains however due to LAN synchronisation and lack of perfect camera focus.

It was concluded that this device, using an LAN for control and an automatic 3D calibration procedure, would be sufficient for detecting the formation of shear bands in triaxial tests. As these methods are low cost and do not require expertise to set up or run, this device would be suitable for adoption in geotechnical laboratories more widely.

A 21st Century Laboratory Testing Device for Geotechnical Engineering

Suzie Neave

24th May 2020

Contents

1 Introduction

1.1 Background

Geotechnical engineering forms the foundation of all modern day construction projects. Data from on-site and laboratory experiments provide an insight into the expected mechanical response of land to planned developments. One important measurement is how soil deforms under stress. The triaxial test is the most frequently used method that attempts to answer this question (Figure 1).

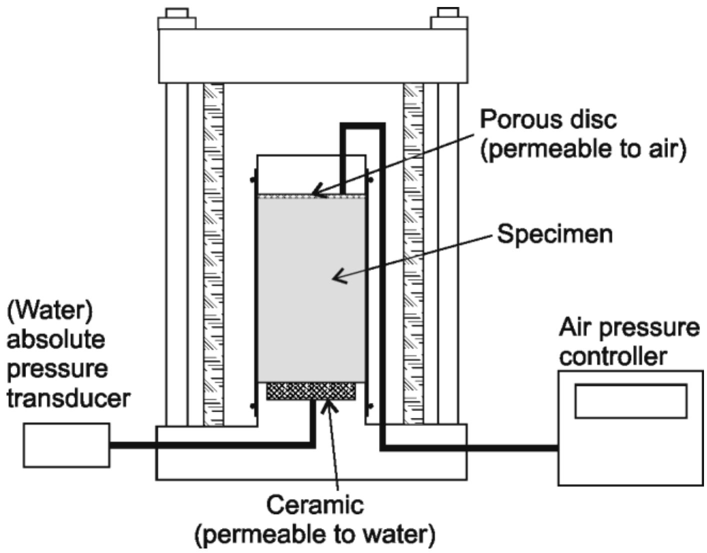


Figure 1: Schematic of a triaxial test set-up [7]

The triaxial test rig can be used to apply compression or tension to a test sample in an axisymmetric stress state. A soil sample is placed inside the triaxial apparatus. The cell is sealed, flooded with water and pressurised. The air pressure provides a constant equal force around the curved surface of the sample. It is then compressed or extended axially. The water in the sample can be allowed to drain during the test (drained test) or can be prevented from draining (undrained test). For this experiment, the cell pressure is a fixed variable. During an experimental run, the axial pressure is controlled and the axial shortening and water drainage volume are measured [11]. These measurements are used to calculate the volumetric strain caused during testing. The results can then be used to characterise the soil. They may be used to form the basis of expected settlement calculations, for example.

The triaxial testing method presents a number of problems. The test relies on the assumption that the specimen, whilst inside the cell, acts exactly as it would when surrounded by a continuum of the same soil type, ie. the results produced assume the test is an element test. At the beginning of test runs, this is a relatively accurate assumption. The soil deforms uniformly and the boundaries at the cell wall and end plates do not cause any unexpected effects during compression. As the test progresses however, these assumptions often break down as soil samples bifurcate.

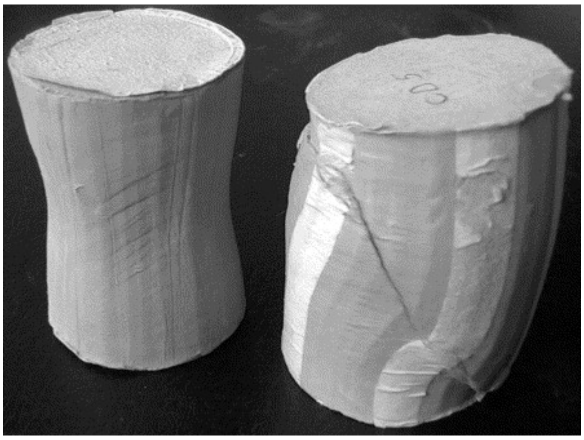


Figure 2: Soil sample showing bifurcation and shear band localisation in tension (left) and compression (right) [9]

As samples deform, the soil particles align and form shear bands (Figure 2). A contributing factor to the formation of these shear bands is the friction provided by the end plates; boundaries which would not be present in the field [15]. There is a reduction in friction between shear bands which allows large sections of samples to displace relative to one another at much lower stresses than would otherwise be required [22]. The stress-strain response measured is therefore highly dependant on the point at which these bands forms. At this point the sample no longer accurately represents a soil element and predicting in-situ soil behaviour from these results becomes difficult [8].

After shear band formation, soil no longer deforms isotropically, however due to the design of the triaxial test, there is no way to separate the strain localisation at the shear bands from the continuing strain within the homogeneous sections of the sample. Both the axial shortening and escaped water volume only provide average measurements of strain. In order to better understand this stage of the soil response, it is necessary to determine a method to detect the point at which these shear bands localise and to provide a method to continue to take local strain measurements beyond this point.

1.2 Literature review

Scientists have been aware of the shortcomings of triaxial tests at least since the 1980s [2] however that hasn't stopped them being used. There have been plenty of attempts to update the testing method to reflect these known errors however no method has yet been widely adopted in the construction industry. This is often due to new methods being incredibly complicated, requiring specialist skills to operate, being very temperamental (hence risking the possibility of introducing human error) or requiring very expensive equipment on top of the pre-existing triaxial rig.

The effects of shear-banding have been studied extensively over the past 40 years. A wide range of theories have been produced in order to model soil behaviour post-bifurcation. Their accuracy often relies on the assumption that bifurcation occurs at the peak of the stress-strain curve however this has been shown to not always be the case [9].

1.2.1 Traditional measurement techniques for local strain

The first attempt at introducing local strain measurements to triaxial tests was carried out by Brown and Snaith in 1974 [4] by placing LVDTs (a linear displacement measuring device using electric transformers) directly onto the soil samples. Despite providing a high accuracy in strain measurement, their large size, high cost and temperamental nature made them an impractical device for routine application.

An alternative method for measuring local strains was used by Hird and Young in 1989 [12] with proximity transducers. This relied on measuring the displacement of target rings placed around the sample. Although also providing accurate measurements, this set-up was expensive and often could not be fully submerged or withstand pressure changes, making it almost impossible to implement routinely in standard triaxial tests.

The issue of sample disturbance was inherent to any method where measuring equipment was applied directly to the sample (Figures 3 and 4).

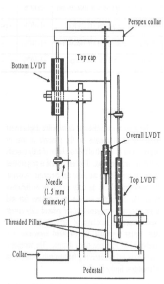


Figure 3: LVDT strain measurement system [18]

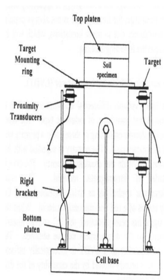


Figure 4: Proximity transducer strain measurement system [12]

LVDTs were also used in measuring radial strain however this area has been less well explored and the same difficulties apply. There are inherent limits on the potential of this type of solution, as only one direction of measurement (axial or radial) can be taken at one time and only at discrete intervals. Despite this, these methods confirmed the need for local strain measurements. In their 1984 paper, Jardine et al [13], concluded that, "...conventional external measurements of displacement [in triaxial tests] contain errors which are frequently so large that their use in the determination of soil stiffness at working levels of stress is invalid."

1.2.2 Detection of shear band formation

Separately from local strain measurement testing, experiments have also been conducted into shear band formation. Their existence has been recognised for some time however detection of their point of formation during triaxial tests has not yet been standardised in equipment.

Most current methods used for shear band analysis use either X-rays Computed Tomography (X-ray CT) and electron microscopy [5] [8] or rely on detection by the human eye. The first two methods provide a 3D model of the internal structure of a test sample which can be viewed in slices (Figure 5).

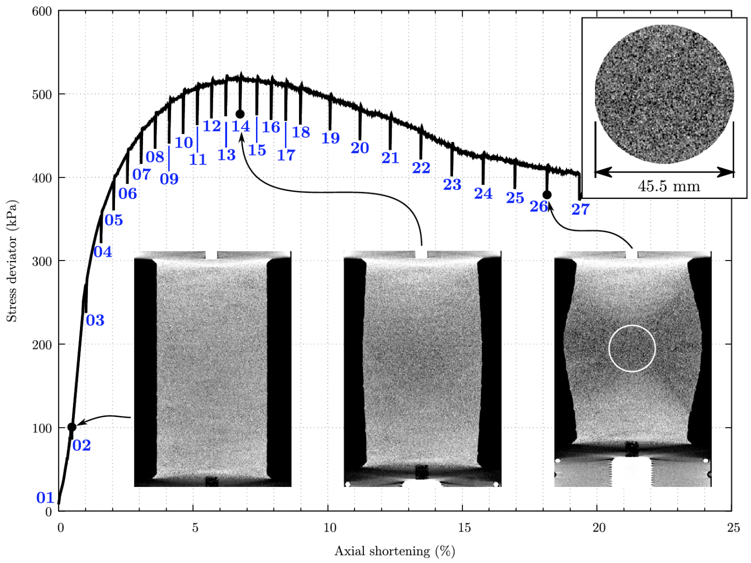


Figure 5: Results from a triaxial test analysed using X-ray CT (2018) [8]

X-ray CT analysis provides a detailed and comprehensive model of the internal mechanisms of the triaxial test. The scans however require a halt in the triaxial test as each one takes approximately two hours to complete. The scans appear on the stress-strain curve in Figure 5 as stress relaxations, each marked with a number. Therefore this method cannot be applied directly onto a standard triaxial test procedure and is unsuitable for detecting the exact moment of shear band formation. This method also relies on human interpretation to identify shear bands, introducing subjectivity to the test.

1.2.3 Computer vision methods

To improve and combine the methods of triaxial test analysis mentioned previously, a number of recent developments have focused on using computer vision techniques to provide a more holistic data set for strain analysis. These methods have the advantage of allowing measurements to be taken without any additional equipment coming into contact with the sample. Additionally they allows data to be gathered for the entire sample surface both axially and radially and do not require tests to be halted in order to take measurements.

Digital imaging systems for triaxial tests were developed independently in 2007 [16] and 2012 [3]. Both used high quality cameras placed outside the triaxial cell to capture images of the sample surface (Figure 6). Whilst both systems proved capable of providing high resolution data that agreed with previous results from LVDT methods, the cost of the cameras and complexity of the set-up meant that the method has not been widely adopted for application in routine triaxial tests.

This topic has been approached by both the University of Cambridge and the University of Western Australia with the most recent trial of a measurement scheme created by Shapally in 2017 [19]. This method used Raspberry Pi (RPi) cameras positioned up against the surface of the triaxial test cylinder (Figure 7).

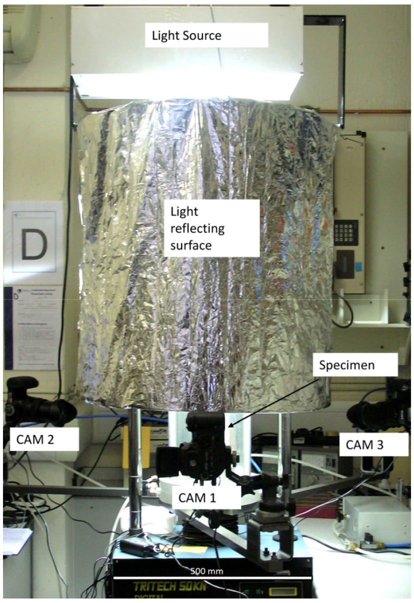


Figure 6: Experimental set-up used by Bhandari et al [3]

Figure 7: RPi camera set-up using zip ties to mount cameras

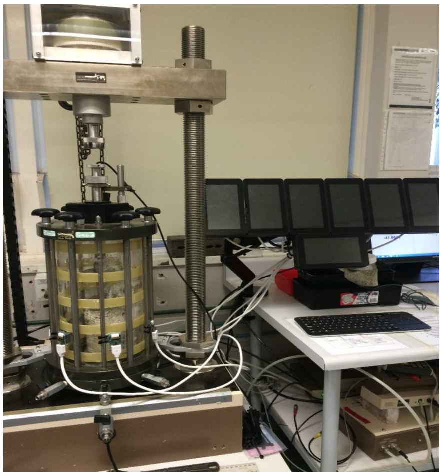


Figure 8: Control system for RPi cameras using one RPi screen per camera

Using RPi cameras reduced the cost of the system significantly and the simple method used for mounting the cameras led to an easy set-up process. Shapally's paper provided evidence that this method for gathering data from triaxial tests was possible, however with an unwieldy control system (Figure 8) and no account made for image distortion, there were a number of developments required before the device would be suitable for use.

A shortcoming of the computer vision systems developed so far has been the use of 2D data for strain measurements. This approach assumes that the sample surface remains completely flat and parallel to the camera during testing. In reality, samples barrel and deform, resulting in inaccurate measurements. In order to properly account for sample deformation, it is necessary to consider the 3D effects displayed by the sample.

1.3 Overall design principles

The aim for this project was to develop a device to gather as much information regarding strain as possible from a standard run of a triaxial test. This information can then be used to evaluate local strains and to identify the onset shear band formation. The method adopted stemmed from that used by Shapally in order to design a device that could be adopted more widely due to moderate price and simple operation and set-up.

The device was developed with the intention of applying a more advanced analysis of sample deformation and strain than previous methods. By measuring 3D particle displacements and sample distortion, a complete view of the surface deformation could be obtained during a test run.

The design of the device can be split into three sections; the rig, the control system and the imaging system.

1.3.1 The rig

For the device rig, it was important that the design left little room for human error during installation. It also had to be easy and inexpensive to manufacture.

One problem that exists with using external cameras is the refraction caused by the cell walls and the water inside the cell. This issue was addressed by Bhandari et al. by running a calibration procedure prior to soil testing, then using post-processing of the images to eliminate distortion. Shapally's solution did not take into account image distortion but instead minimised it by placing all cameras as close as possible to the cell surface thus reducing the curvature of the cell surface within the field of view of the camera. In order to provide the most accurate results possible, the new design incorporated both methods.

It was also required that the entire surface was within the field of view of at least two cameras to allow image processing to locate particles in 3D space.

1.4 Control system

To provide an easy method for running tests, it was desired to have a centralised control system such that experimental runs could be started, monitored and analysed from one computer. This would allow laboratory tests to be conducted without requiring a detailed understanding of how the system operates.

As previously mentioned, in order to locate particles in 3D space, it was necessary that each point on the surface appeared in at least two images. Therefore another important part of the control system was that it should operate all cameras simultaneously and should be able to synchronise the data taken from the imaging system with the data output from the standard triaxial test.

1.5 Imaging system

The imaging system constitutes the key part of the device. Strain measurements rely on the 3D location and tracking of points on the sample surface. To locate points accurately, it was necessary that any image warping caused by refraction was accounted for and corrected.

To locate soil particles in 3D, it was also necessary to know the exact relative location and rotation of all cameras used for the device.

As they would need to be run before a triaxial test run, the calibration and camera location procedure needed to be quick, simple and require little to no understanding of the principles behind it. The parts required needed to be either standardised or simple to manufacture in a workshop.

The system eventually should be able to take local strain measurements from image sets automatically, plotting these against the average stress-strain values taken by the triaxial test.

2 The camera rig set-up

2.1 Design process

There were a number of constraints in place for the design of the camera rig:

1. It should position the cameras as close to the cell wall as possible to minimise image distortion.
2. It should ensure that every point on the surface of the sample is within the field of view of at least two cameras.
3. It should be simple to manufacture without requiring heavy machinery or expertise.
4. It should leave little room for human error when being set up.
5. It should be inexpensive.

The triaxial test rig chosen to act as the basis of this device had a 200mm diameter, a completely clear cell and six vertical bars for connecting the base plate. The camera placement was determined based off a soil sample size of 70mm diameter and 140mm height.

Raspberry Pi V2 cameras were chosen for the device as they are small, inexpensive and capable of taking high resolution images. Each camera was connected to an individual Raspberry Pi Zero W for control purposes.

Camera	Raspberry Pi V2
Resolution	3280 x 2464 pixels
Horizontal viewing angle	62.2 degrees
Vertical viewing angle	48.8 degrees

Table 1: Raspberry Pi V2 camera technical specifications

In order to ensure full surface coverage, a total of 36 RPi cameras were required; three rings of twelve cameras. Mounting the cameras sideways allowed each ring of cameras to provide double coverage of the circumference and ensured that the coverage of each ring overlapped.

The design of the mount itself was completed in CAD and 3D printed. This method allowed the mount to be reproduced easily by anyone with a 3D printer and access to the CAD file. On this occasion, the mounts were printed using an Ultimaker 2 Extended 3D printer using PLA filament. The design went through a number of iterations (Figure 9 and Figure 10) with improvements made in between.

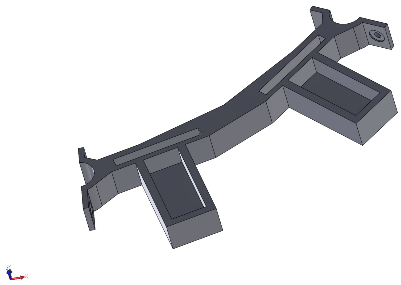


Figure 9: Original mounts design

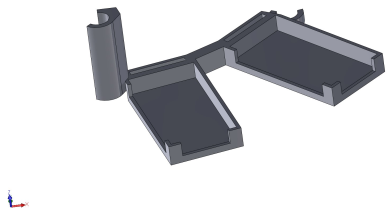


Figure 10: Final mount design

All design iterations were modular; designed to hold two RPi cameras mounted between two of the vertical metal bars. The dimensions were based on the dimensions of the RPi Zero and camera whilst inside the dedicated 'Adafruit' case. For the RPis themselves, this was purely to prevent damage. For the cameras, the arms of the 'Adafruit' RPi camera case extended from the main body which provided a convenient method for mounting the cameras without obscuring their field of vision.

The final mount design attached to two of the vertical bars using clip-on attachments (Figure 10). These were staggered such that if a ring of mounts were attached, the connectors would stack and the ring would align. The left arm/connector was extended upwards to increase stability and to act as a spacer between the rings of mounts. The RPis were mounted flat behind their respective cameras. Power cables could then be connect to each RPi without preventing the rings from stacking at the necessary spacing (an issue encountered with previous design iterations). The RPis were oriented such that the camera cables protruded from the back of the mount. This limited any excessive bending of the camera cables which would have otherwise been required. Each camera was held on its side by inserting one arm of the camera case into the designated slot. These were positioned to ensure that the lens of the camera was positioned as close as possible to the triaxial test cell wall.

2.2 Final design

In order to demonstrate and test the system, three copies of the mount design were printed. A scaled down version of the final test rig was set up with six cameras spanning one of the six segments of the triaxial rig. All components used are listed in Table 2. The additional 3D printed spacers were used on the metal fixing pole between the long arms of the mount to mimic the arm of the mounts used on the next segment. These would not be required for the full set-up. Figure 11 and Figure 12 demonstrate how the RPis and cameras were held in the mounts and how the mounts were connected to the rig. Additional lighting was used to provide better contrast in the images. This was in the form of a ring light placed on top of the triaxial cylinder.

Figure 11: Two-camera mount with RPis and cameras in place

Figure 12: One of six segments set up with camera mounts

Component	Quantity
Raspberry Pi Zero W	6
16GB microSD cards	6
Raspberry Pi Module V2 Camera	6
Raspberry Pi Zero Camera Connector	6
Adafruit Raspberry Pi Camera Board Case	6
Adafruit Raspberry Pi Zero Case	6
Custom 3D printed mounts	3
Additional 3D printed spacers	2
Anker PowerLine 3ft Micro USB Cable	6
Anker 6-port USB Hub	1
Soft-focus ring light	1

Table 2: Table of components used for the scaled down rig set-up

2.3 Discussion

The advantages and disadvantages of this set-up became clear when it was put into use for further development in the project.

The clip on attachments were effective in holding the cameras stationary when they were not touched. The front of the camera cases pressing against the cell wall provided additional stability. This meant that, although the RPi section of the mount had some freedom of movement due to bending, the camera lenses remained fixed in position. All measurements from this system relied on a high level of accuracy of relative camera position calculations. Because of this, alongside the fact that triaxial tests have a long run-time, it may be desirable to improve the stability by lining the clip on attachments with small segments of rubber.

The slot for the camera was effective in positioning the cameras correctly (against the cell wall) and holding them in place. Although this improved stability, it caused some issues with camera focus. The V2 camera has a fixed focus which can be adjusted manually using a camera focusing ring. Doing so removes the glue that holds the focus in place. After this, the focus becomes susceptible to inadvertent accidental adjustment. To bring the RPi camera's focus to the required range, the cameras had to be taken out of their cases. However, when put back into their cases, the focus was accidentally adjusted due to the lens pressing against the front of the camera case. This issue was mainly due to ill-designed cameras cases. It was minimised (but not eliminated) by removing the tape between the camera module and the PCB board, giving the camera lens more space.

In general the camera cases and cables chosen for this rig caused a number of problems. The cases were ill-designed and resulted in breakages of some of the more vulnerable parts of the RPis (SD cards and the camera connection clip). As the purpose of the cases was purely for protection in the first place, it was clear that the cases were causing more damage than they were preventing. It would therefore be safer to mount the RPis without their cases. This may require the size of the RPi slot to be reduced to hold the RPis more firmly. The system would have also benefited from more flexible, longer power cables, as the power cables used often pulled the RPis out of their mount if they were knocked, causing further damage.

3D printing the mounts was successful. Each individual mount was printed in around 5 hours and cost only £2 in filament. A 3D printer with quality of the Ultimaker 2 Extended or above (100 micron resolution) was required as printing the mount in a lower quality printer (RS/IdeaWerk 3D printer) resulted in distortion and a poor fit of the components.

3 Control and synchronisation

3.1 Design

3.1.1 Design principles

The purpose of the control system was to connect the RPi hardware to the calibration and test-run software. There were three main aims for the control system;

1. To provide a system where minimal knowledge was required to set-up and operate it.

2. To achieve an appropriate level of synchronisation in image capture.

3. To provide a method for allowing newly captured data to be compared to the equivalent data collected from the standard triaxial test rig.

In order to fulfil the first criteria, the control system was developed using a local area network (LAN) to communicate between the components. This was chosen over hard-wiring the components together as this would require a certain level of expertise and professional equipment to set up. It would also result in the final system (all 36 cameras) being difficult to assemble and disassemble and soldering may have been required with the RPis in place.

In order to provide centralised control for the system, a designated laptop was assigned as the 'control laptop'. During normal operation, all controls, data collection and data analysis could be made from this laptop, preventing the need for connecting the RPis to individual displays for any reason other than initial configuration.

3.1.2 Initial configuration

The control system developed requires a number of steps to be completed upon first set-up of the test rig. This only needs to be done once, before the first test run. Any subsequent uses of the device would not require these steps to be taken.

1. A copy of the correctly configured Raspbian operating system should be downloaded onto the control laptop. Three python code files should also be downloaded onto the control laptop; 'configuration.py', 'test_run.py', and 'query_image_time.py'.

2. A copy of the correctly configured Raspbian operating system should be installed on each RPi's SD card. This will include the standard operating system, a copy of the code required to respond to requests from the control laptop and a pre-configured setting to set this code to run from boot.

3. Each RPi should be numbered to allow images to be matched to the cameras they were taken from.

4. MAC addresses provide a unique identification for each of the RPis via its network chip. Each RPi should be started whilst connected to a display, keyboard and mouse. The MAC Address of each RPi should be queried by running `ifconfig -a link` in the terminal.

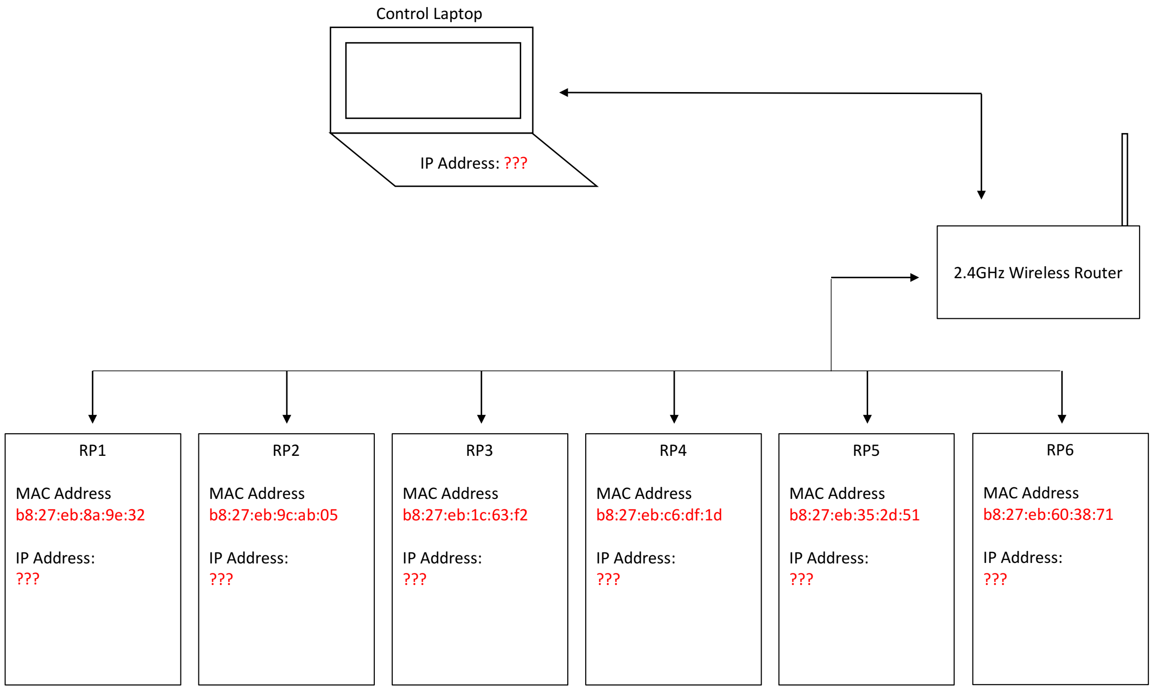


Figure 13: Diagram of scaled-down network set-up

5. The MAC addresses should be entered next to their corresponding RPi number in the 'configuration.py' and 'test_run.py' code files on the control laptop.

6. As discussed further in Section 4, the cameras connected to each RPi should be focused. To do this, the camera connection must first be activated within the configurations menu. A continuous view of the camera output can be displayed by running `raspistill -t 0` in the terminal. The cameras should then be focused using the standard focusing ring provided with the RPi cameras.

3.1.3 System operation

Once the initial configuration procedure has been completed, the set up is in a state where it can be operated solely using the central laptop (Figure 13). The system can then be used for calibration image capture or during a test run. The procedure runs as follows;

1. All RPis and cameras are mounted and connected to power. The router is connected to power and the control laptop is connected to the router's network.

2. 'configuration.py' is run from the terminal on the control laptop. This code:

    (a) Creates a number of RPi objects containing information including their label and MAC Address.

(b) Checks for a file, 'pi_data.txt', of saved RPi IP addresses. If found, saves the IP addresses to the respective RPi objects.

(c) Attempts to open a web socket connection with every possible IP address on the LAN. If a connection is made at an IP address, this indicates that an RPi within the system has adopted that IP. The MAC address at that IP is queried and the IP is saved to the corresponding RPi object.

(d) Having scanned through all addresses, the 'pi_data.txt' file is created/updated with any newly identified or changed IP addresses.

All RPis run a listening code from boot. This allows the web socket connection to be established when set up by the central laptop. Once the connection is established, the RPi saves the IP address of the central laptop for further communications and moves on to running a continuous listening code for use during a test run.

3. The 'test_run.py' code is run from the control laptop. Two versions of the 'test_run.py' code were tested. Both involved the control laptop broadcasting commands on the LAN which are picked up by the listening RPis. These broadcasts include instructions for the RPis to run a specific Python code stored in their own memory which operates the camera and saves the images. The time each image is taken (according to the clock on the RPi) is saved into the image metadata.

4. Once the test run is over, the images are collected from each RPi by the central laptop. This can be done using the scp command run from the terminal; `scp pi@[ipaddress]:home/pi/images/* path/to/folder/on/central/laptop`. IP addresses for all the RPis in the system can be read out in order by running `cat pi_data.txt` in the terminal.

5. The saved image time data can be queried by running 'query_image_time.py'. The time between image captures can be determined by finding the mean of the time difference between subsequent images. Note that this is set initially within the 'test_run.py' code however is not always accurate. This is explained further within Section 3.2.

3.1.4 Synchronisation

To calculate 3D coordinates of surface points accurately, synchronisation between the RPi cameras was important. Initially the 'test_run.py' script was configured to send one broadcast to all listening RPis. On receiving this signal, each RPi would run an internally saved script, capturing one image every x seconds y number of times. For calibration and synchronisation testing, x was set to 3 seconds and y to 10. During triaxial testing, the chosen x and y will depend on the test being run. x could vary at least from 1 second [19] up to 4 minutes [3] depending on whether it is being used for an drained/undrainded and consolidated/unconsolidated test. y will depend on the length of the test but will also be limited by available memory of the RPi SD cards. With approximately 13GB of space left on each 16GB SD card after installing the operating system and code files [10], there should be enough space for approximately 2600 images per camera. This corresponds to a run time of around 45 minutes to 10 hours without intermittent image recovery and deletion procedures.

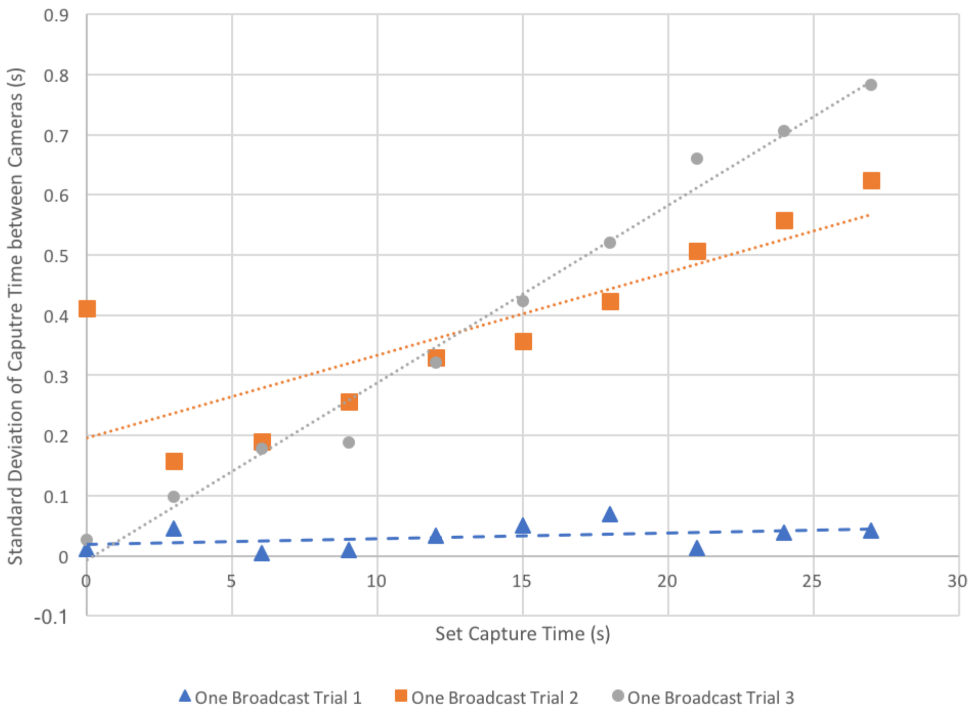


Figure 14: Variation of capture time between cameras using single broadcast method

To measure the synchronisation between cameras, all cameras were pointed at a stop-clock. The 'test_run.py' script was run and the time visible on the stop-clock was compared between cameras. 10 images sets were captured; one image every three seconds.

3.2 Results and discussion

Figure 14 shows how the capture time varied between the cameras. For each test run, the standard deviation of the capture times increased over time.

Using the single broadcast method meant that, beyond the first image, the time between image captures was measured independently by each RPi's internal clock. No further communications were made beyond the initial broadcast. This resulted in synchronisation deteriorating with time as the RPis internal clocks were not synchronised. Full runs with a triaxial test would last far longer than the test run therefore this method would not be suitable to limit the variation between capture times.

Another effect observed was that, despite the time between captures being set to three seconds on all RPis, all RPis lagged behind this significantly, accumulating up to 4s of lag over a 27s run. Although there was no need for the time between image sets to be set exactly, it was important to be able to measure it in retrospect in order to fit the new data to the data measured by the standard triaxial test rig. This was achieved by writing the image capture time into the metadata of the images. The relative capture times could then be used by the central laptop to provide an indication of the mean image capture time.

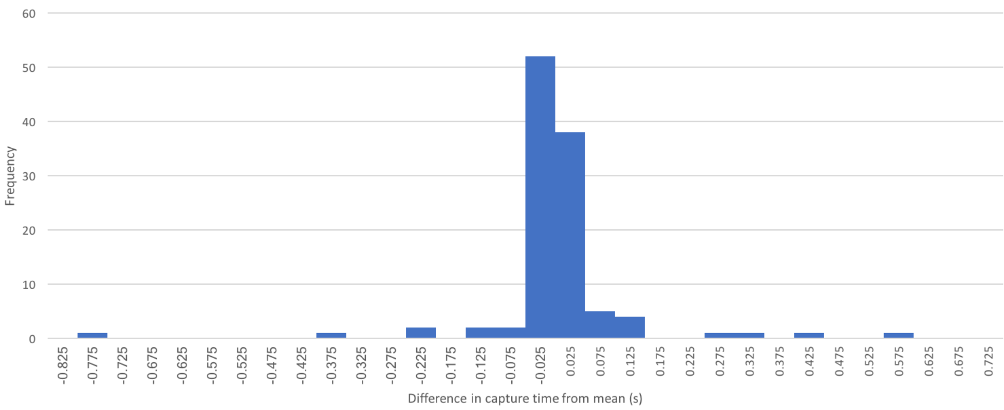


Figure 15: Variation in capture time between cameras using multiple broadcast method

In response to the lack of clock synchronisation, 'test_run.py' and the camera code were adapted. The second configuration measured the time between image captures on the central laptop, sending out broadcasts at each interval. The RPis responded to each broadcast by capturing one image. This eliminated the problem caused by the difference in clock speed between RPis. The synchronisation therefore purely depended on the send-and-receive time of broadcasts over the network.

Figure 15 shows a histogram of the capture time variation around the mean in response to individual broadcasts. Typically most cameras were activated within 0.05s of the mean however outliers push the standard deviation of the distribution considerably higher. These outliers occurred at random points during a test run, usually with all but one camera maintaining relatively good synchronisation. This suggested that the strength of individual RPis connection to the LAN varied over time. As all RPis were close to the wireless router and not moved during the experiment, it was possible that the random variations in capture time were caused by other loads in the 2.4GHz frequency space [1]. In order to combat this, a router running on the 5GHz band could be used. This could supply higher speeds of connection whilst minimising disturbances as the band is usually less cluttered. The short range that comes with routers operating at this band would not pose issues as the router remains close to the device throughout test runs.

With six cameras using a router operating at 2.4GHz, the standard deviation of the distribution was 0.128s. As the lag between command and capture time on each camera is independent, it can be assumed that this would extend to all 36 cameras. Using three times the standard deviation as the expected limits, this experiment suggests an expected range of approximately 0.768s between the first and last image capture for one image set[1]. This was higher than anticipated, mostly due to the outlying data points. As previously stated, these could likely be minimised by switching to the 5GHz frequency space in order to avoid interference in the network.

[1]An image set describes the set of 36 images, one from each camera, used to create a 3D model of the sample surface at a single time step.

4 Calibration

The calibration of the Raspberry Pi cameras was necessary to ensure accurate 3D modelling of the sample surface. A number of methods of calibration were analysed, tested and compared.

4.1 Theory

4.1.1 Pinhole camera model

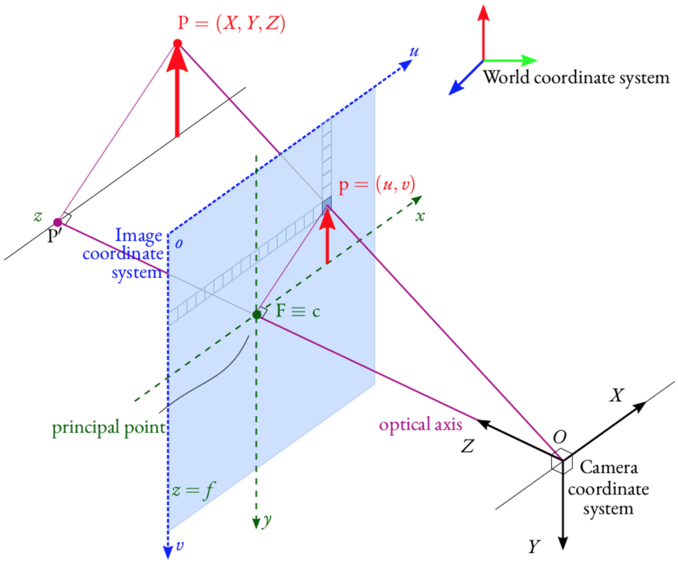


Figure 16: Pinhole Camera Model [20]

The camera calibration theory used was based on the pinhole camera model. The concept behind the model is that all light rays from the 'real world' are focused to a point in the camera, referred to as the camera centre or optical centre. The image captured by the camera is a projection of these rays onto the 'image plane'. Figure 16 shows the world point **P**=(X,Y,Z) projected to a point **x**=(x,y) in the image plane. The point **p**=(u,v) indicates the location of the point **x** measured in pixels from the top left corner of the image[2].

The projection theory provides equations that relate point **P** to point **p**. It is useful to note that projection theory often uses homogeneous coordinates which allow points at infinity to be represented with a defined ratio between x and y coordinates (Equations 1 and 2).

$$\boldsymbol{x} \equiv (x, y) \equiv (x', y', z') \tag{1}$$

$$x = \frac{x'}{z'} \qquad y = \frac{y'}{z'} \tag{2}$$

[2]This is the coordinate system used by most image-editing software packages

The equation mapping world points to pixel coordinates is show in Equation 3. Note that this equation assumes no non-linear distortion.

$$\boldsymbol{p} \equiv \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \equiv \boldsymbol{P_c P_p P_r P} \tag{3}$$

Three projection effects are accounted for in the standard full-camera model equation:

1. A rotation and translation between the world coordinates used to measure the object points in and the set aligned with the camera. Accounted for in $\mathbf{P_r}$ which can be separated into a rotation matrix (encapsulating r parameters) and a translation matrix (encapsulating t parameters).

2. The projection effect that determines the intersection between a ray and the image plane. This acknowledges the dependency of the image points on the distance of a world point from the optical axis and the focal length of the camera. Accounted for in $\mathbf{P_p}$.

3. A scale and shift which maps coordinates in the image plane to pixel coordinates. Accounted for in $\mathbf{P_c}$.

4.1.2 Image distortion due to refraction

Viewing the test sample through a the cell wall and a column of water introduces non-linear distortion. This is due to light refraction caused by the differing refractive indices of air, water and the cell wall. The amount of refraction depends on the angle of incidence. Assuming the pinhole camera model, this will vary from the top to the bottom and from the left to the right of the image due to the angle of the rays. The curvature of the cylinder will also affect the angle of incidence. The larger the curvature present within the field of view of the camera, the greater the distortion (Figure 17).

The camera calibration module used for this project was the OpenCV Python module. In addition to the standard pinhole projection parameters, the OpenCV calibration algorithms include parameters for radial and tangential distortion due to lens warping. This distortion is modelled by a warp applied about the centre (principle point) of the image plane prior to the conversion to pixel coordinates.

$$x' = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 xy + p_2(r^2 + 2x^2) \tag{4}$$
$$y' = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2y^2) + 2p_2 xy \tag{5}$$
$$r^2 = x^2 + y^2$$

In this case, refraction acts similarly to lens warping. The one-directional curvature of the cylinder however will cause additional distortion in the x direction over the y direction; something not allowed for in the OpenCV equations (Equations 4 and 5). In order to minimise the directionality of the distortion, the cameras were placed as close as possible to the triaxial cell walls, minimising the curvature within the field of view of the cameras. It was necessary to test whether this alone would be effective or if additional parameters would be needed in the distortion equations to separately correct the x and y distortion.

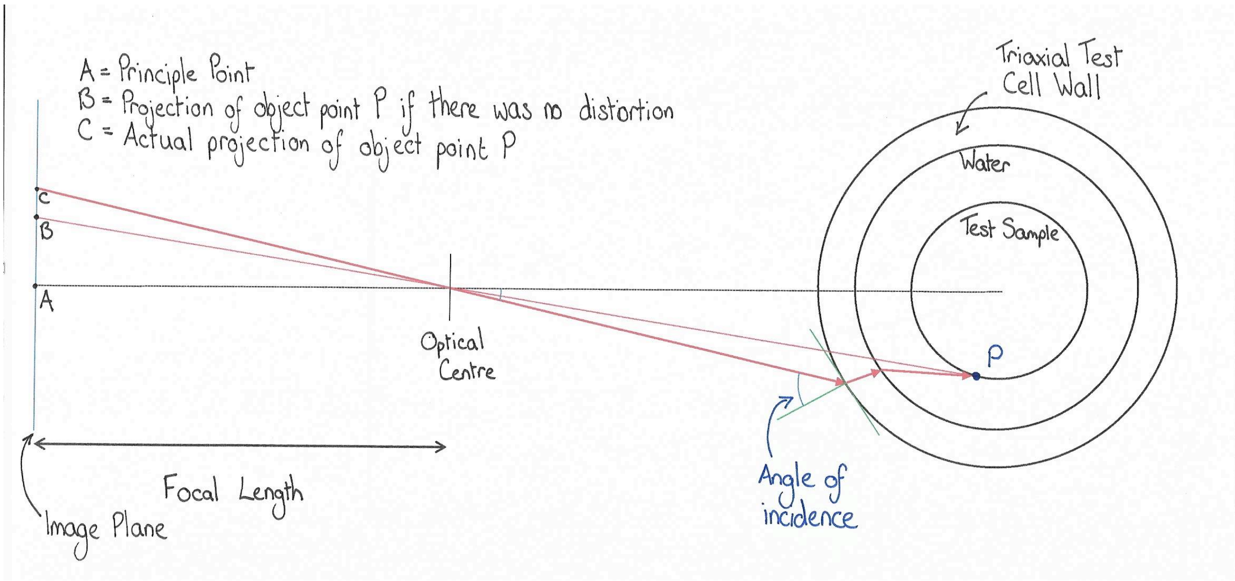


Figure 17: Light refraction due to curved cell surface

4.1.3 Calibration procedure

Calibration targets are used to help determine the parameters of the distortion equations. Once the parameters have been determined, the distortion equations (Equations 4 and 5) can be applied in reverse to images taken by that camera to remove the effects of radial and tangential warping.

To determine the distortion parameters, a number of known object points (real world 3D coordinates) and their corresponding image points (pixel location of the object point as it appears in the image) are required. The traditional calibration procedure uses images of 2D chessboard patterns to provide these calibration points. The corners of the chessboard squares act as the object/image points. Images of this board are taken at multiple angles to gather a set of non-coplanar points to use for calibration[3].

The object point's coordinates are known simply by measuring the chessboard and can be measured relative to any axis as chosen by the user. The relative image points are then found by searching for large 2D changes in intensity within cropped sections of an image [14]. Relating the object and image point coordinates in a set of images provides a system of simultaneous equations which can be solved for the distortion parameters of that camera.

4.1.4 Relative camera position calculations

Just as important to the imaging system was the calculation of the relative location of cameras. This allows 3D reconstruction of the triaxial test surface. For this to be possible, multiple individual world points need to be matched between images from the multiple cameras. Rotation and translation matrices for each camera can then be calcu-

[3]If points collected for calibration are coplanar, the simultaneous equations generated to determine the parameter values would not be independent.

lated relative to the same set of world coordinates. This cannot be done using a simple calibration board as each world point needs to be distinguishable and its position relative to other world points known.

4.2 Design

4.2.1 3D calibration target

To calibrate the cameras for the distortion caused by the triaxial cell, the calibration procedure needed to take place with the cameras in position and the test cell filled with water. This requirement made it almost impossible to use a 2D calibration board without having to empty and refill the triaxial cell multiple times in order to capture object points in different planes. To overcome this, a 3D calibration target was created using an aluminium cylinder with a screw fixing holding it where the test sample would usually sit. Calibration patterns were then printed onto sticky-back vinyl and attached to the target. Using a 3D target allows non-coplanar points to be detected from a single image. The calibration process can then be run by placing the calibration target into the cell and capturing images with the target in one position.

To be suitable for calibration, any calibration board must have points that are easily identifiable using computer vision techniques. For this test, two different categories of calibration board were created.

4.2.2 Charuco boards

OpenCV Charuco module combines 3D reconstruction with camera calibration. It produces camera calibration chessboards (Figure 18) with ID markers (Aruco markers) inside the white squares. Each ID marker uniquely identifies the corners surrounding it. Image point detection works the same way as the simple chess-board calibration algorithm. Corners are extracted from the image and used as image/object points for calibration.

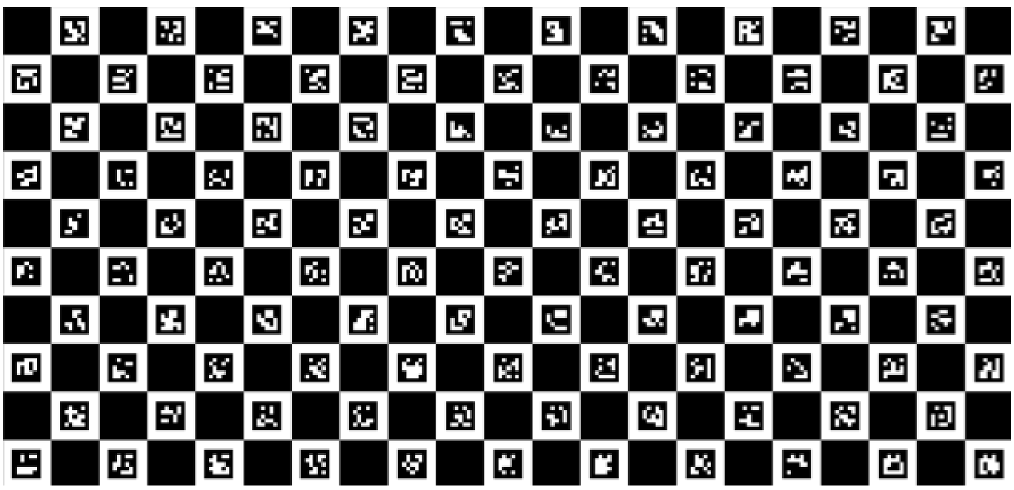


Figure 18: OpenCV Charuco board pattern

As the Charuco algorithms were designed to be used with flat calibration boards, for the calibration of this device, the Aruco marker and Chessboard calibration methods were combined. The 3D Charuco calibration procedure therefore was designed to use the Aruco module for corner extraction and identification, and the chessboard calibration module to calculate the distortion parameters.

4.2.3 Circuco boards

Another feature easily identifiable by computer vision techniques is blobs or circles. There are multiple methods available for detecting these features. The first is a blob detector. Blobs are defined as ‘an area of uniform intensity in an image’ [6]. They are detected by convolving images with a 2D laplacian of a guassian at varying scales and detecting peaks in the output (Figure 19).

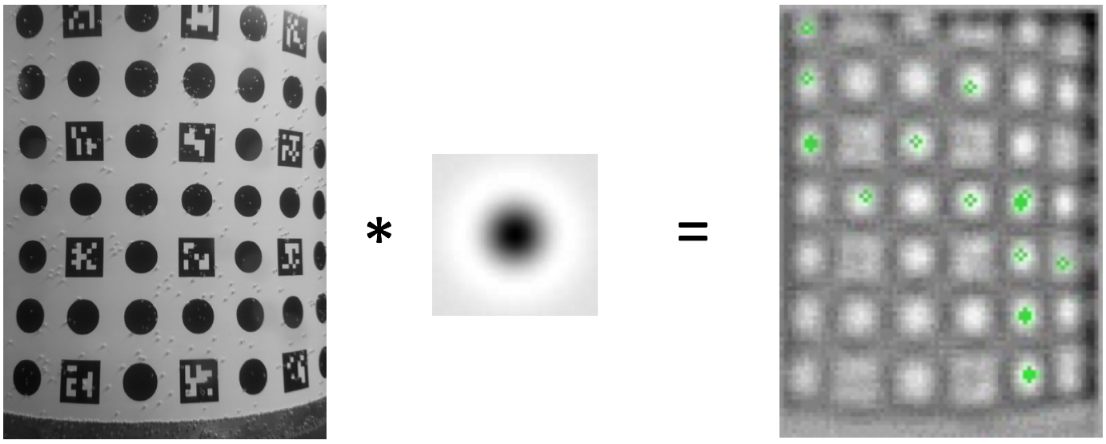


Figure 19: Blob detection

The second method is the Hough circle transform (Figure 20). The algorithm first applies an edge detector which identifies edges in an image. It then builds a list of all the possible circle diameters and radii that would have each of those edge points on their perimeter [17]. Circles are identified when a large number of the edge points identify the same circle equation.

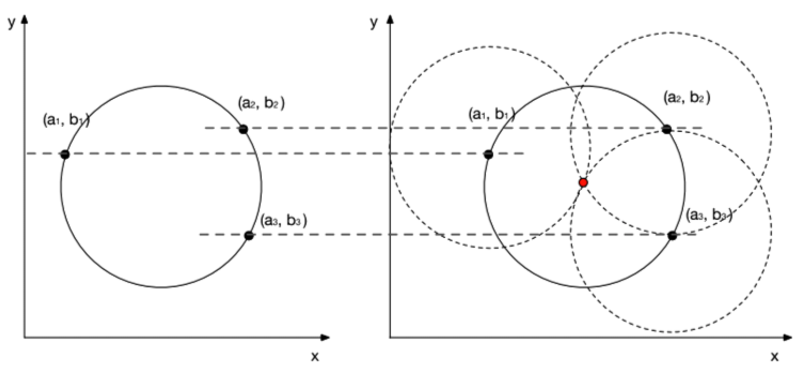


Figure 20: Hough circle transform example

The Circuco pattern was created as a second calibration board option. This incorporated the Aruco markers from the Charuco board and circle features for use as object/image points. The Circuco calibration algorithm was designed to operate similarly to the Charcuco algorithm. To provide unique identification, each circle was given an identification number related to its relative position between Aruco markers (Figure 21). In calibration, the Aruco markers are detected and their corner locations are used to crop the image to show specific individual circles. The coordinates of the circle centres are then refined using either the Hough transform or the blob detector.

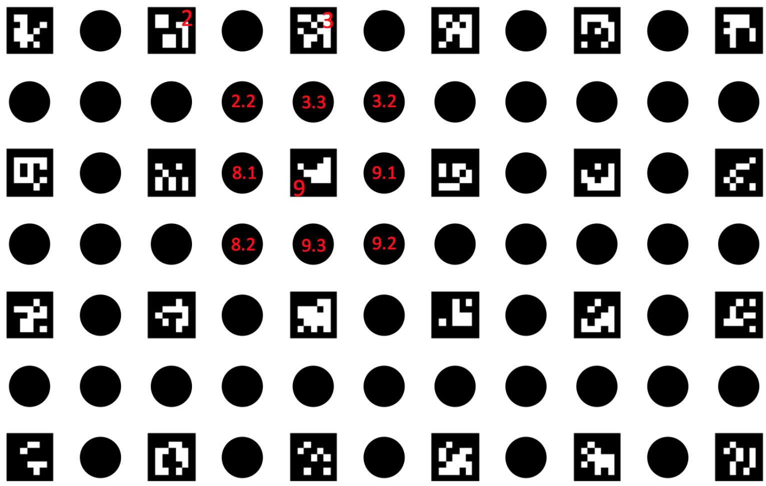


Figure 21: Circuco board pattern showing the ID referencing system. The ID of one Charuco marker is used to label the circles to the right, bottom-right and bottom of the marker.

Both the Hough transform and blob detector often identified more than one relevant feature in the cropped image provided (Figure 22 and 23). The Hough transform relied on a number of parameter inputs including 'minDist'. The 'minDist' parameter set the minimum distance allowed between detected circles. In order to narrow down the 'correct' circle, the Hough transform was set to run a number of times for one cropped image, increasing 'minDist' if too many circles are detected and lowering 'minDist' if none are. This process loops up to twenty times at which point, either a single circle is detected and recorded as an image point or the point is discarded. The blob detector initially also identified a number of the bubbles on the surface of the triaxial cell as blobs. This was managed by activating the 'filterByArea' tag in the blob detector paramaters and setting the minimum area to 10,000 and the maximum area to 100,000.

Using circles for calibration is not as simple as standard calibration as it inherently introduces some error into the procedure. Projecting a circle from an oblique plane onto the image plane causes distortion of its shape. Due to the convergence of parallel lines, this causes the projection of the centre and the centre of the projected ellipse to diverge (Figure 24). It was therefore important that this effect was taken into account when evaluating the final precision of calibration techniques .

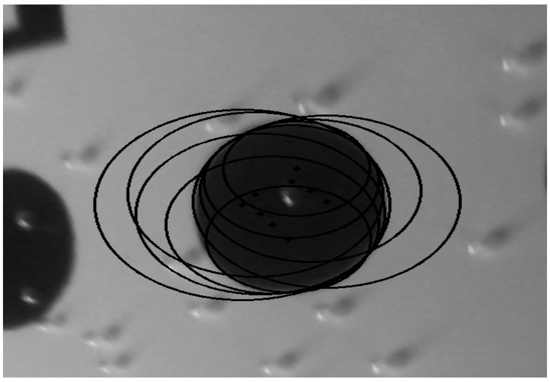


Figure 22: Hough circle transform with parameters incorrectly set

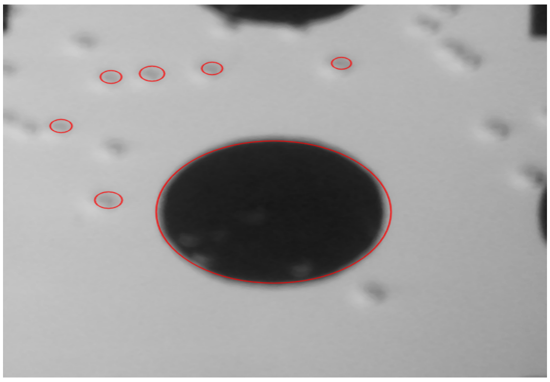


Figure 23: Blob detector without filtering by area

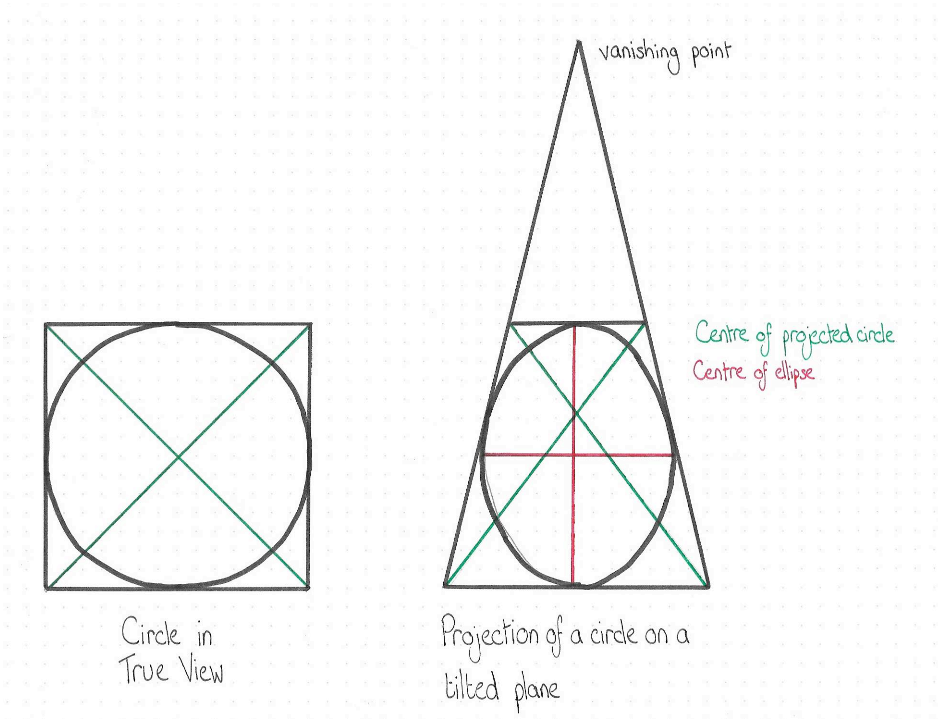


Figure 24: Divergence of the centre of a projected circle and the centre of the resulting ellipse

4.2.4 Intricacies of calibration

In order to use a 3D board for calibration, the OpenCV chessboard calibration function requires an initial guess of the camera matrix (the product of $\mathbf{P_c}$ and $\mathbf{P_p}$ in Equation 3). This encapsulates the internal camera calibration parameters.

Two of the parameters, u_0 and v_0 can be estimated relatively accurately. They represent the pixel coordinates of the principle points; the projection of the focal point onto the image plane. This can be taken as the centre of the image and therefore u_0 and v_0 were initialised as half the width and height of the input image respectively. The values of fk_u and fk_v are less simple to estimate and although they are used as only an initial estimate, their value affects the outcome of the calibration algorithm massively. Not only

Board Type	No. Cameras	No. images captured by each camera	No. different angles captured in the image set from one camera
Charuco	1	10	3
Circuco	6	10	1

Table 3: Data set of images acquired for calibration testing

this but the input values producing the best results vary between calibrations. To ensure that the calibration achieved the optimum results based on the calibration points it was presented with, the algorithm was designed to iterate over a set of initial k values (where k equals fk_u and fk_v) to find the most effective initial matrix guess (defined as the matrix resulting in the lowest corresponding RET value).

4.3 Results and discussion

4.3.1 Data set

Table 3 above summarises the data set that was collected and available for full analysis.

4.3.2 Error measurements explained

A number of error measures were used to evaluate the calibration algorithms. During calibration, rotation vectors, translation vectors as well as the camera matrix and distortion parameters were calculated. The object point coordinates of every calibration point could then be projected onto the image plane using Equation 3 and the distortion equations. If the parameters were calibrated perfectly, each object point would project to the exact location of its corresponding image point. These projections were known as re-projected object points. All error measurements were calculated using the distance between image points and re-projected object points.

The OpenCV calibration algorithms have an inbuilt error measure labelled ‘RET value’ which measures the RMS error between image points and re-projected object points. A number of additional error measures were used for further analysis including the standard deviation of the error in the x and y direction. A large difference in the standard deviation in each direction would indicate that the distortion is direction-dependant and that calibration could be improved by including directionally-biased warping.

4.3.3 Calibration using image points from all collected images

Figures 25, 26 and 27 show the results of initial calibration testing before and after outlier removal using all three methods. Outliers were identified as calibration points

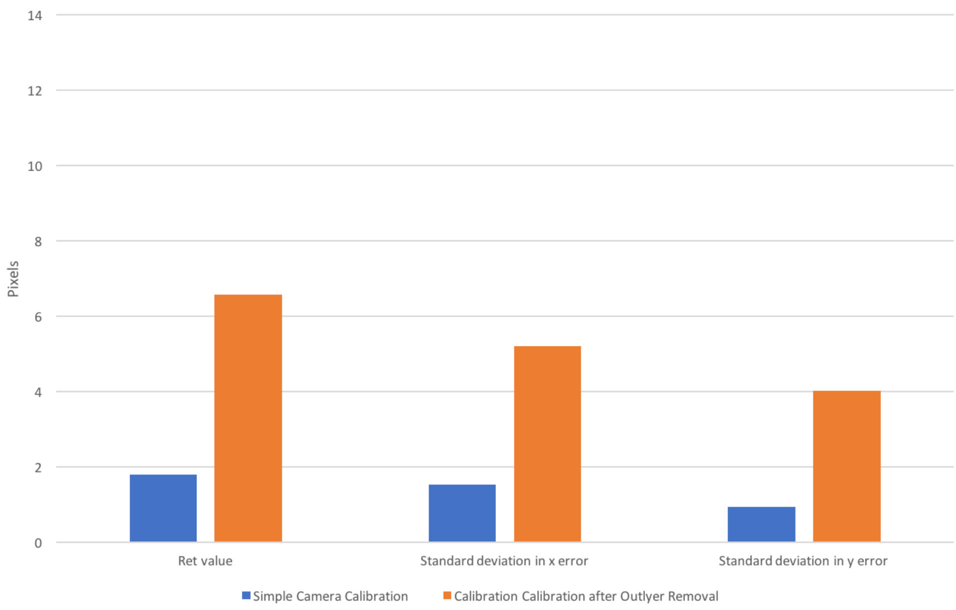


Figure 25: Average calibration results using Charuco calibration and ten images for calibration

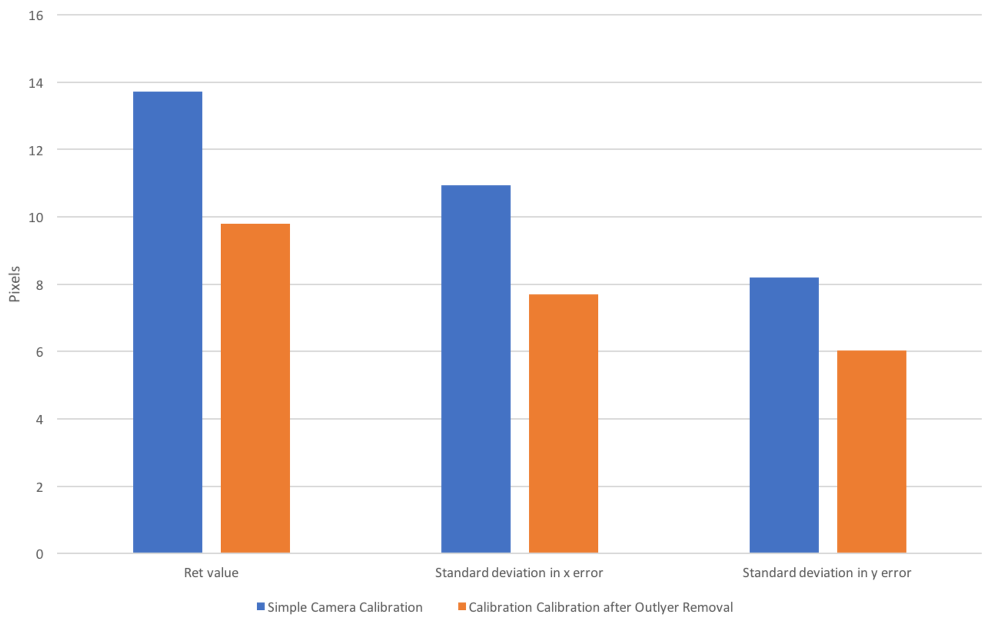


Figure 26: Average calibration results using Circuco Hough Tranform calibration and ten images for calibration

with a RET value over three times the standard deviation of the error for that calibration. Outlier removal involved the removal of these points from the list of identified calibration points and recalculating the calibration parameters and corresponding error without these points. A result of this analysis was that, although outlier removal apparently improved the calibration for both the methods using the Circuco board (Figures 26 and 27), it caused the Charuco calibration method to produce much worse results (Figure 25).

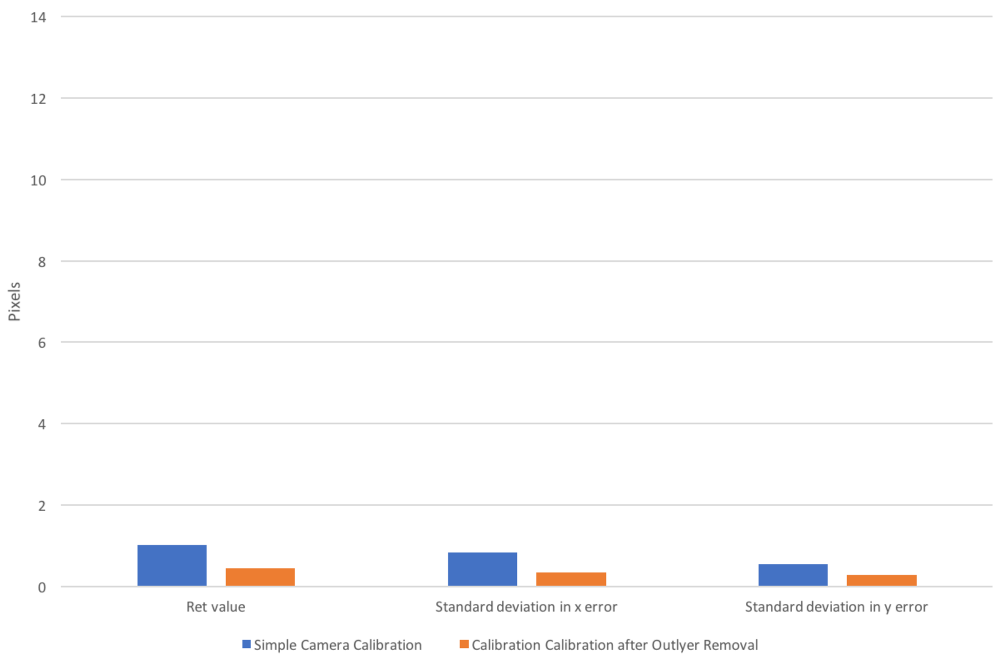


Figure 27: Average calibration results using Circuco blob detector calibration and ten images for calibration

This was likely a problem of over fitting. The algorithms using the Circuco board were calibrated using ten images of the same view of the calibration board. In contrast, the image set for the Charcuco board included six images at one orientation, one image at a second orientation and three at a third orientation. Using the exact same orientation resulted in some image points being detected incorrectly consistently through a number of images. Figure 28 shows an example of this occurring in the Charuco detection algorithm. These three images were taken from the same camera with the same orientation of the Charuco board. Each time the image point was detected marginally to the right of where it actually appears in the image (indicated by the red dot). With multiple images of this orientation used for calibration, this point was weighted by the number of images that agreed with the poorly detected position. The calibration algorithm was therefore more likely to ensure the parameters fit that point (evidence of this is visible in Figure 28 as the re-projected point in green also appears to the right of the actual image point). This would have caused a much worse fit for correctly detected points in other images after calibration. This issue was amplified by the outlier removal, as the parameters were already fit to incorrect points. The incorrect points in this image were therefore not removed as they fit the calibration algorithms. With other points removed, most likely in the images of a different orientation, these incorrect points gained an even stronger weighting, resulting in a worse overall calibration.

This finding suggested that over-fitting was also occurring in the Circuco calibrations as every image (for each camera) was taken with the target at the same orientation. The over-fitting was not evident from the error measures alone however as the calibration algorithm was not tested against any other orientation. It was clear that the two data sets could not be compared directly. Ideally, another data set would have been collected, rotating the target between each image capture to provide a range of view angles for

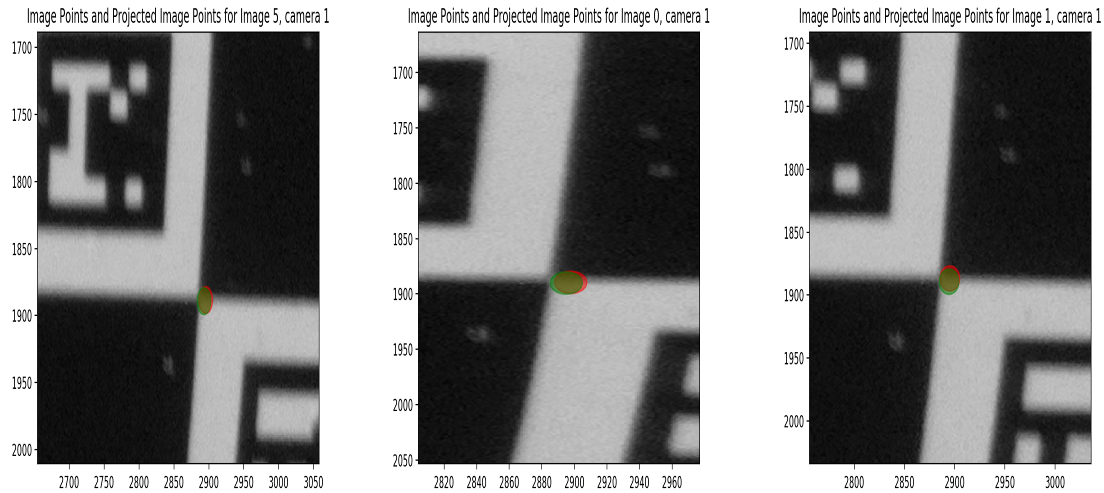


Figure 28: Misidentified point in multiple image from one camera

each camera. Due to access restrictions however, no more data could be gathered. It was therefore decided to continue testing using just one image per camera for calibration comparison.

4.3.4 Calibration using image points from a single image

Figure 29 demonstrates that the Charuco board and blob detector method both considerably outperformed the Hough transform. This was potentially due to the difficulty in ensuring the correct circle was identified out of the initial suggestions given, therefore more image points were identified off centre. The Hough transform was subsequently discarded as a method of calibration as the blob detector provided better results using the same calibration target and images.

Another key result from the calibration tests was whether errors were significantly differently distributed in the x and y direction. The Charuco data provided three angle views of the board.

Figure 30 indicates that there was a slight tendency for a higher error in the y direction resulting from Charuco calibration. This suggests that the distortion parameters calculated during calibration were fit more accurately to the radial distortion in the x direction, and that the distortion in both directions could not necessarily be modelled by the same parameters. More data would be needed to make a more definitive conclusion however these results indicated that there would be a potential gain in calibration accuracy by correcting the x and y distortion separately.

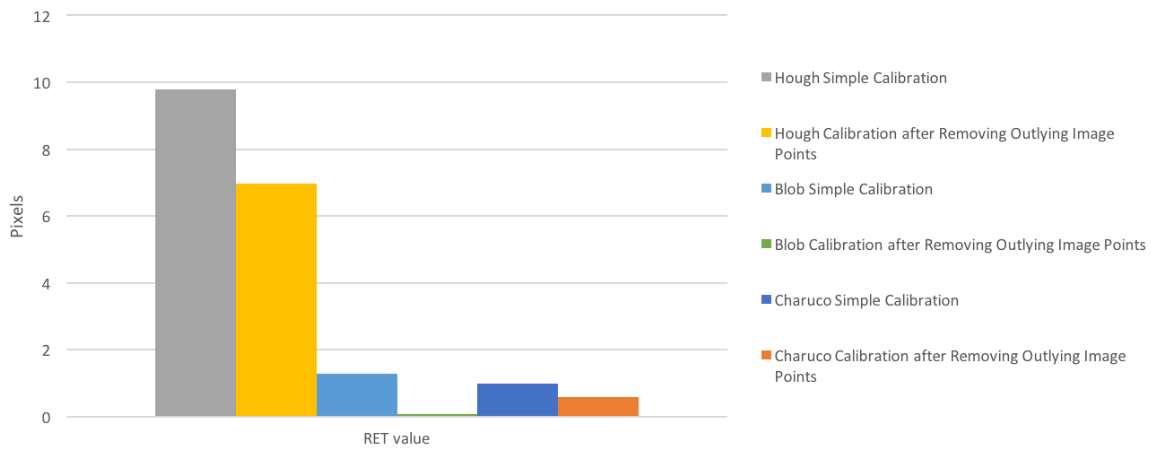


Figure 29: Average calibration RMS error using each calibration method. Using only one image per camera for calibration

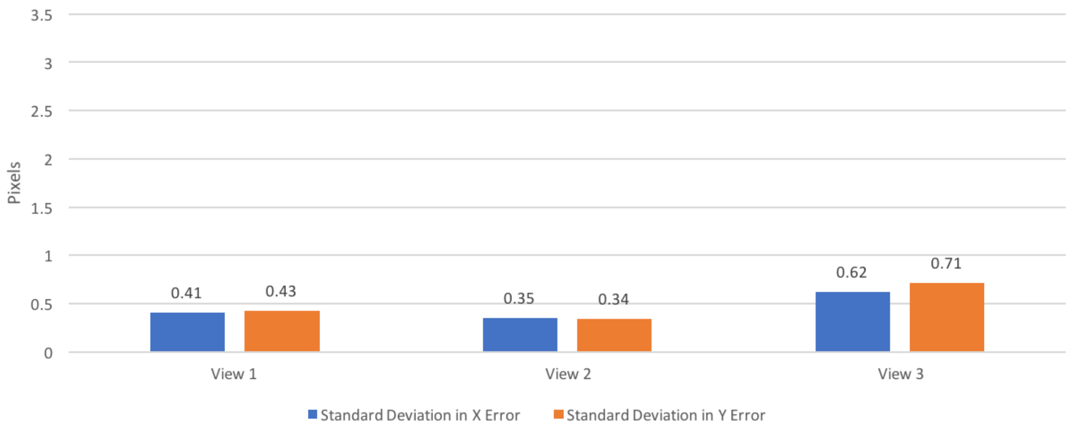


Figure 30: Comparison of error distribution in the x and y direction using Charuco calibration with outlier removal

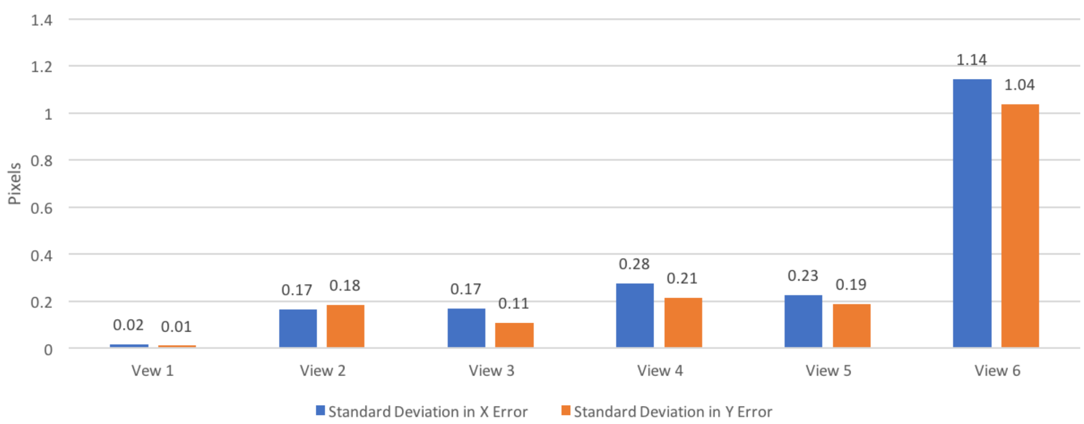


Figure 31: Comparison of error distribution in the x and y direction using blob detector Circuco calibration with outlier removal

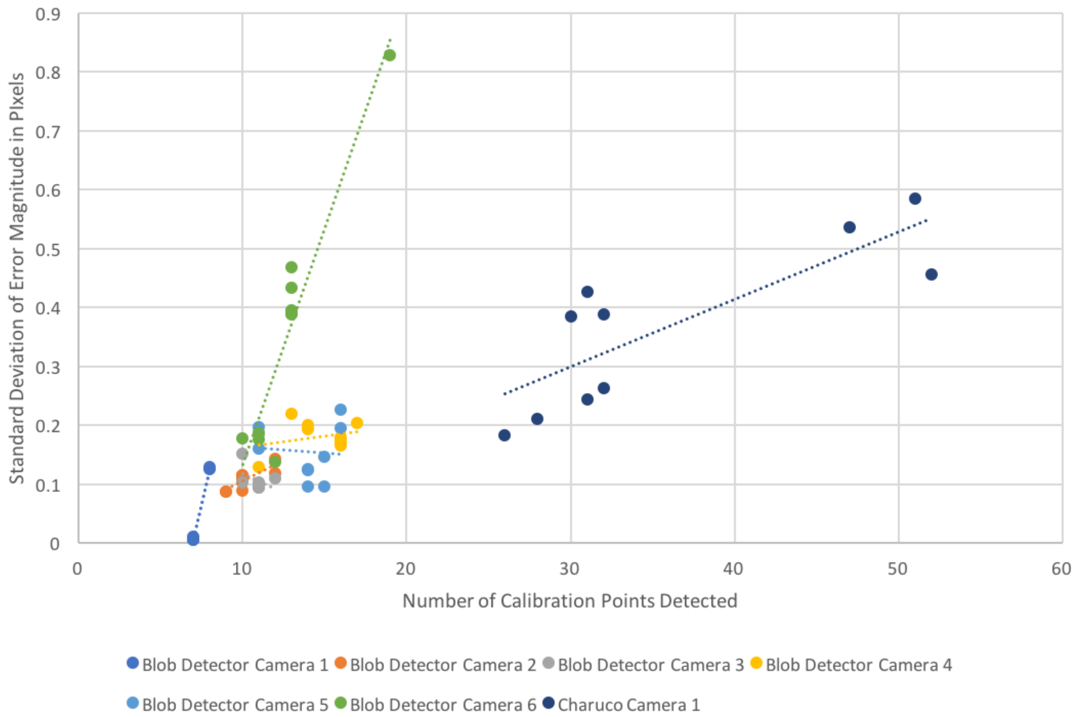


Figure 32: Plot of the standard deviation of error and the number of calibration points used

Figure 31 demonstrates a slight tendency for a higher error in the x direction for Circuco calibration. With Circuco calibration however, the error introduced by circle projection results in an added directional effect beyond refraction. This is due to the variation in the angle between the target surface and the image plane. Although this makes it difficult to determine whether refractive effects were non-negligible, the common differences in the distribution of x and y error once again indicated the potential benefit of correcting x and y distortion separately.

An additional result to come from this data was an correlation between the number of image points detected and the standard deviation of the error. Figure 32 compares the number of calibration points detected and used for calibration in each image with the standard deviation in the magnitude of the calibration error. If the calibration parameters generated were accurate, the standard deviation of error would be expected to decrease with each additional point detected. On the contrary, the standard deviation appeared to increase with the number of points detected for the same camera. Similar result were seen when comparing the RMS error with the number of calibration points. Not only did this suggest that the calibration parameters were not accurate, it indicated that the error measures alone were ineffective at indicating the accuracy of the calibration. This theory is supported by Figures 33 and 34 as the image which provides the worse calibration according to the data clearly produces a more sensible calibration according to the eye. If the calibration had been successful, the vertical lines would appear straight[4].

[4]They may also appear parallel but only if the image plane of the camera was exactly parallel with the vertical axis of the cylindrical target. This was not strictly necessary for calibration or operation of this device.

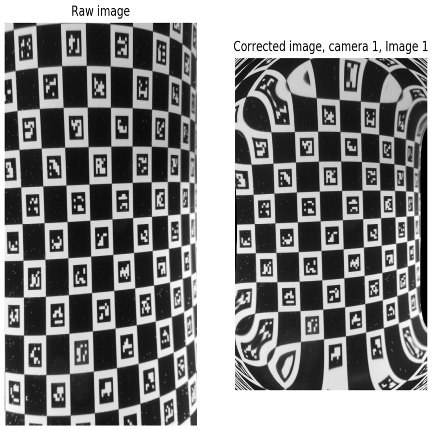


Figure 33: Charuco calibration with 26 points detected. RMS error = 0.372 pixels

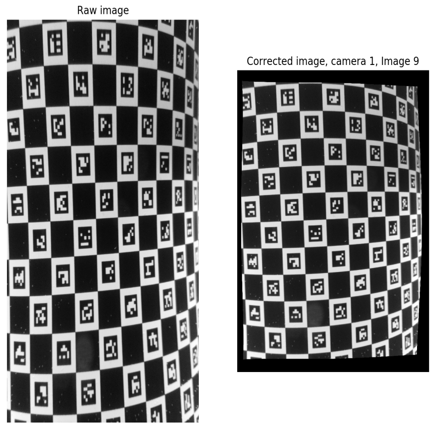


Figure 34: Charuco calibration with 52 points detected. RMS error = 0.827 pixels

It was clear from this analysis that calibration using one image was likely to produce a poor calibration and the associated error would be underestimated. This analysis also suggested that the 'RET value' produced by the OpenCV algorithm is a misleading measure of the accuracy of calibration algorithms when considered alone.

4.3.5 Calibration using image points from images of three separate angles

For a more accurate calculation of the error, the Charuco data was used for further development as this was the data set with multiple views available for the same camera. The Circuco data, although providing reasonable results at first glance, presented multiple issues in calibration that would have likely extended into further calibration testing. With significantly fewer calibration points present in each image, it was possible that the success of the Circuco calibration was merely a result of the effects demonstrated in Figure 32. In addition, all errors measured so far had yet to take into account the error introduced by circle projection. Appendix B includes calculations of the expected additional error. Although the error caused could be reduced considerably by using smaller circles on the calibration board, the size used for this Circuco board (radius of 2.5mm) resulted in an additional re-projection error of at least two pixels which could not be refined easily during image processing. With the accuracy already achieved using the Charuco boards, this was too large for the scales being considered. No further optimisation was carried out with the calibration using the Circuco board.

Continuing with the Charuco board, to obtain the best calibration with a more accurate measurement of the error, all image/object points from three images of three separate angles were used for calibration. The error was calculated against all of these points. Three angles of the Charuco board were available for the first camera. A number of data sets were created, using a random selection of the available images at those orientations.

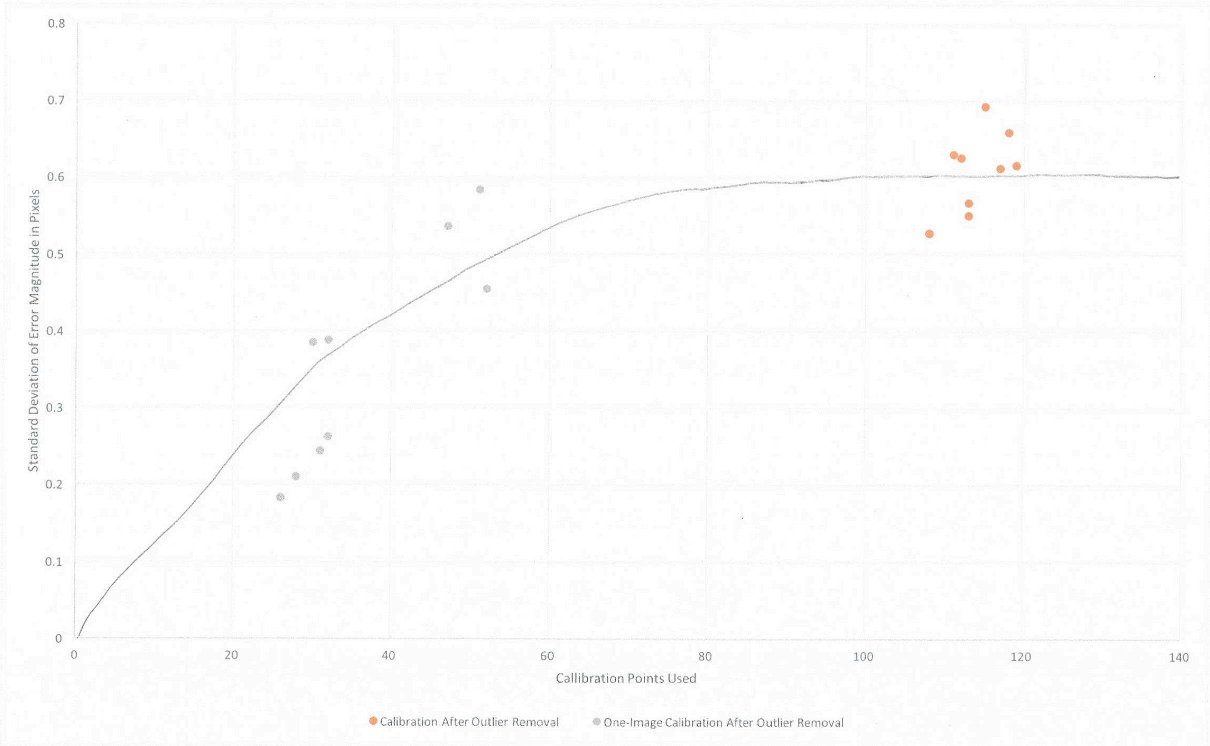


Figure 35: Plot of the standard deviation of error against number of calibration points

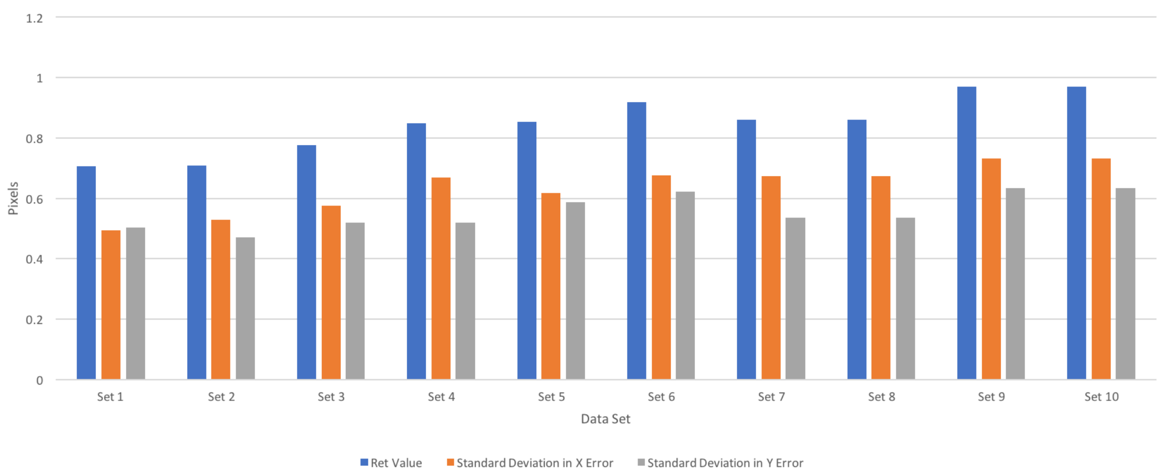


Figure 36: Error from three-angle calibration

Figure 35 demonstrates that, by using three separate angles for calibration, the effect of the number of identified calibration points on the error standard deviation has been limited. The measure of error calculated for these calibrations therefore provides a better indication of the accuracy of the calibration parameters. Due to the shape of the data, it suggested it would be reasonable to take the RMS error (the RET value) as the 'correct' error measure for the calibration algorithm.

Figure 36 shows the error ascociated with the accumulated data sets. All error measures were taken after outlier removal. The important result to note is the difference between the standard deviation in the x direction and the y direction, with the x direction consistently having a higher standard deviation than the y direction. This could be a result of similar data being used for calibration or it could have been due to additional warping in the x direction due to cell curvature.

4.3.6 Calibration with separate x and y warping

The OpenCV algorithms use a limited number of distortion parameters. More parameters would require more detected calibration points in order to calibrate. In this instance, all calibration images detected significantly more than six points; the required minimum number of calibration points. Therefore, for this calibration system, it was possible to introduce an additional set of parameters which correct the x and y distortion separately.

To calibrate the parameters, the re-projected object points and image points were compared. A perfect calibration algorithm would be able to project object points to the exact position of the image points that they correspond to. Discrepancies would indicate a process of image distortion that cannot be accurately modelled by OpenCV's calibration procedure. If an additional set of parameters could be determined to map these re-projected points to the detected image points, separately for x and y, a better description of the full projection process could be made. These equations could subsequently be used in reverse to calibrate the images. For calibration, the inverse of the equations mapping re-projected points to image points would be applied to the raw image, followed by the standard OpenCV calibration procedure using the distortion parameters and camera matrix calculated for the newly warped image. The error could then be calculated as the distance between image points detected in the pre-warped image and the re-projected object points. This was the method adopted to further enhance the calibration procedure.

To produce the initial warp equations, the position of each image point was described in terms of the position of its corresponding re-projection (Equations 6 and 7). The number of parameters was set to a limit of five in each direction as each image had detected calibration points at least five well-separated y values. Adding more parameters would have introduced the possibility that a warp equation could be found, matching the image points and re-projected object points but causing incorrect warping between the points.

$$u_{imagepoint} = r_{u1} + r_{u2}u_{re-projection} + r_{u3}u^2_{re-projection} + r_{u4}u^3_{re-projection} + r_{u5}u^4_{re-projection} \quad (6)$$

$$v_{imagepoint} = r_{v1} + r_{v2}v_{re-projection} + r_{v3}v^2_{re-projection} + r_{v4}v^3_{re-projection} + r_{v5}v^4_{re-projection} \quad (7)$$

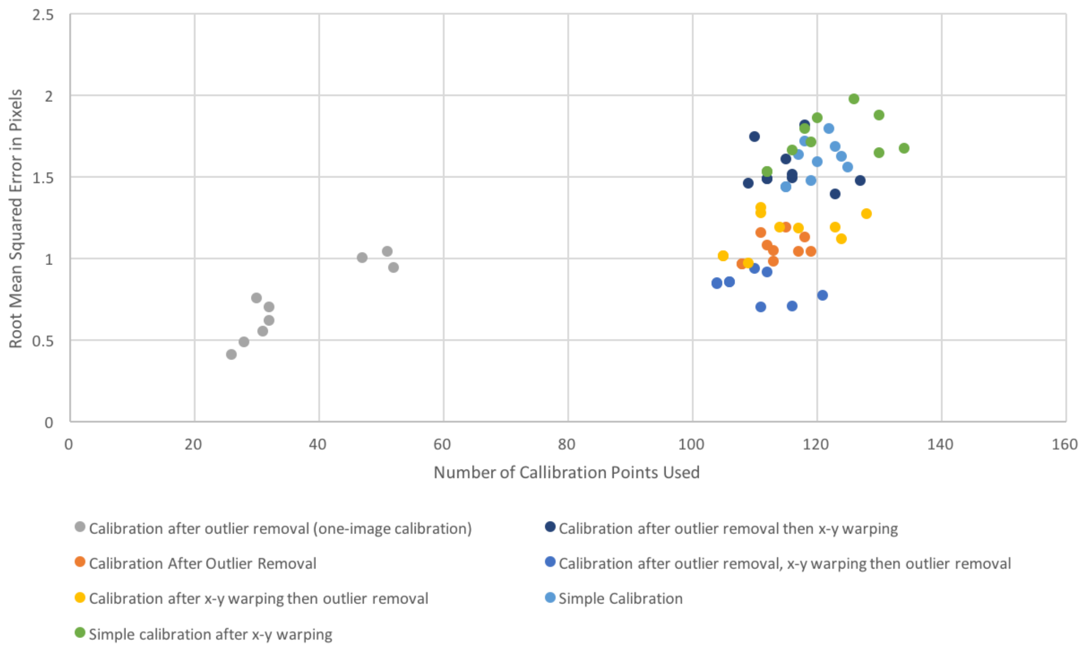


Figure 37: RMS errors from multiple calibration procedures using the Charuco Calibration board

The warp coefficients were determined by substituting in the calibration point pairs, forming a set of simultaneous equations and solving using a least squares approach. Implementing this directional warping resulted in a reduction of the RMS error for most data sets (Figure 37)[5].

Once parameters had been calculated, to calibrate subsequent images, the directional warp would be applied to each image followed by the standard OpenCV calibration procedure. To apply the warp to the initial image, a new image was created of the same size. The value of each pixel was determined by calculating the corresponding coordinate in the raw image using Equations 6 and 7, then interpolating between neighbouring pixels to find the pixel value.

As the calibration procedure was designed to be automatic, each camera could be calibrated using whichever of the methods presented in Figure 37 produced to lowest RMS error. The limits of the final, refined method of calibration are presented in Table 4.

[5]x-y warping resulted in any points discarded during outlier removal to be re-added to the calibration algorithm.

Data Set	**Minimum achieved RMS error (pixels)**	**Corresponding largest error magnitude (pixels)**	**Calibration procedure**
1	0.77556	2.31153	Outlier removal, x-y warp, outlier removal
2	0.91835	3.16989	Outlier removal, x-y warp, outlier removal
3	0.85979	3.36529	Outlier removal, x-y warp, outlier removal
4	1.05006	2.99194	Outlier removal
5	0.847810853	2.69601	Outlier removal, x-y warp, outlier removal
6	0.70762	2.42564	Outlier removal, x-y warp, outlier removal
7	0.70530	2.14195	Outlier removal, x-y warp, outlier removal
8	0.85979	3.36529	Outlier removal, x-y warp, outlier removal
9	0.93890	4.12329	Outlier removal, x-y warp, outlier removal
10	0.85310	2.70094	Outlier removal, x-y warp, outlier removal
Average	0.85205	2.95509	

Table 4: Minimum achievable RMS error for each of the Charuco data sets

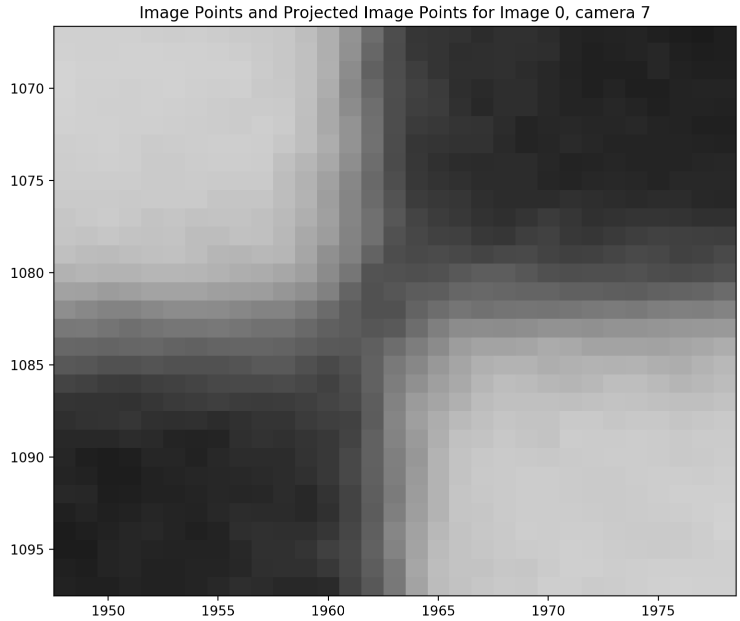


Figure 38: Magnified image of corners visible in Charuco board image

Calibration using the Charuco method was ultimately limited by the quality of the photos taken using the system. Imperfect focus caused discrepancy in the exact location of the chessboard corners (Figure 38). This was down to the focus of the camera. Better focusing of the camera would likely have resulted in a better calibration however, with RPi cameras needing to be focused by hand having been taken out of their mounts, it would be difficult to achieve/ensure a better focus than that achieved here.

This issue was somewhat anticipated and was the reason behind the creation of the Circuco board as blob detection does not require good focus for accuracy. The issues caused by centre projection introduced more error than the lack of focus however. This error could potentially be accounted for with computation however this is not a simple problem due to the non-flat surface of the target and the refraction index dependencies. For this reason, the results achieved using the refined Charuco method were deemed the best achievable calibration results using the 3D target method.

5 3D reconstruction

5.1 Theory

There are a number of steps to be completed in order to finish the build of the testing device. These will include linking the image capture and calibration software to 3D reconstruction and data analysis software. 3D reconstruction requires both feature matching and epipolar geometry in order to locate image points in 3D space. Whilst these are still to be completed, it was nevertheless possible to make an estimation of the precision that the device could achieve and determine whether this was appropriate for strain analysis and shear band identification.

Strain measurements require tracking the movement of a number of points on a test sample over time. To provide reference points for computer vision algorithms, seeded or spray-painted samples will be used. Feature points would be set initially by defining a 3D grid covering the sample surface (Figure 39). This grid would identify a number of points that can be tracked during testing. Their location in the images of neighbouring cameras (post calibration) would be identified using the full camera model equation (Equation 3).

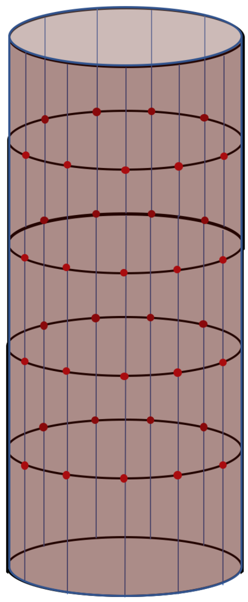


Figure 39: Grid of points on sample surface (red) before refinement assuming the sample is perfectly cylindrical. Point locations can then be refined by feature matching and subsequently tracked through time.

The initial grid would be generated assuming that the sample is perfectly cylindrical. This assumption may be not correct and it may be that the grid point locations identified in two neighbouring cameras may not identify the same surface point (Figure 40).

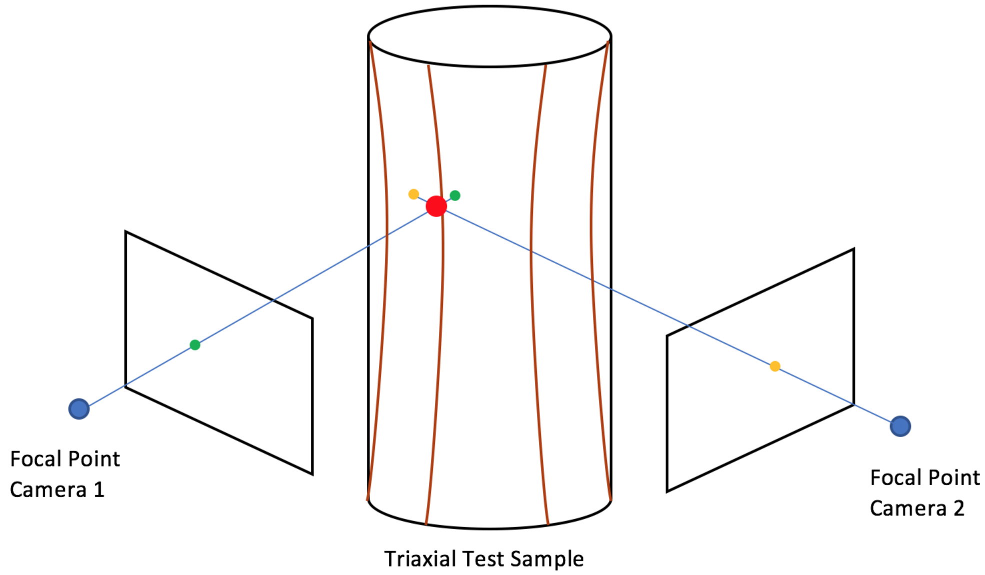


Figure 40: Red grid point projected to the image plane of two neighbouring cameras. As the sample is not perfectly cylindrical, the pixel coordinates calculated for each camera do not reference the same surface point.

With one camera in a pair labelled as the master camera and the other as the slave, the coordinates in the slave camera would then be updated to correctly match the point identified by the master camera. The location could be refined by creating a descriptor for the feature location in one image (SIFT or similar) and searching a small area in the second image for the best matching patch (Figure 41). The exact 3D location of the identified point would then be updated (see Appendix A) and a 3D software model of the test sample surface would be built.

Similar to previous versions of digital image-based strain measurement systems, PIV software would then be used to track the location of these points in each 2D image between time steps. The 3D location of each point would be updated and a new 3D model built for each time step. Strains and their distribution could then be determined from the displacement of the surface points. Using this theory it is possible to estimate the precision of the device due to its design so far.

5.1.1 Synchronisation error

Synchronisation error introduces the possibility that a point may have moved between being captured by one camera and being captured by its neighbour. From synchronisation testing of the device using a 2.4GHz wireless connection, the expected range in image capture was 0.768s. The extent of the movement over this time will depend on the strain rate chosen for the triaxial test.

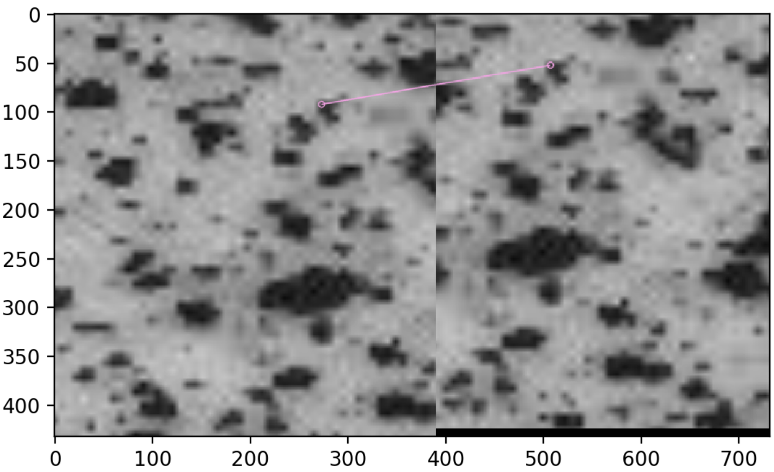


Figure 41: Point identified in master camera (left) matched to corresponding point in slave camera (right) using the OpenCV orb 'detectAndCompute' feature matching function

The error introduced was computed for three strain rates (all used for testing similar imaging systems [3][19]). The strain rate and capture time range were then used to calculate the maximum displacement of a point between capture time and mean capture time (Table 5).

Strain rate (%/min)	**Initial compression rate (mm/min)**	**Maximum point displacement from mean capture time (±mm)**
0.712	1	0.0064
0.0357	0.05	0.00032
0.00357	0.005	0.000032

Table 5: Maximum distance a feature point may be from its position at the mean image capture time

5.1.2 Calibration error

Geometric analysis will be conducted on calibrated versions of all the images taken by each camera. Using Charuco boards and independent x-y correction, the remaining error caused by image distortion is expected to be 0.852 pixels. With approximately a 65mm gap between the camera and sample surface and taking into account the refraction caused by the cell water, this corresponds to ± 0.0144mm between detected and actual location of a point on the sample surface. The uncertainty caused by imperfect synchronisation will add an additional error in 2D point location. Synchronisation error manifests as a difference in the 'correct' particle position in the image view of one camera (position at mean capture time) and detected particle position (position at actual capture time). This error can therefore be accounted for by summing with the calibration error. Considering both causes for error, an image of the surface will locate a particle in 2D with an error margin as shown in Table 6 dependant on the strain-rate of the test.

5.1.3 2D point location error projected to 3D

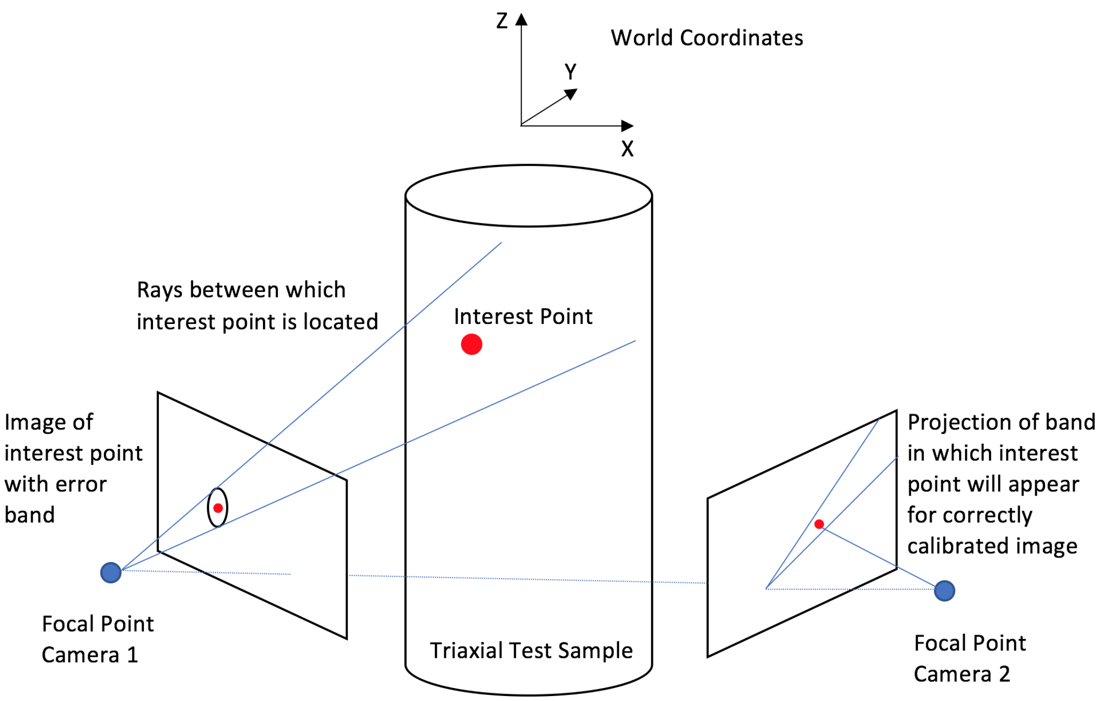


Figure 42: Point identified in master camera (left) matched to corresponding point in slave camera (right)

Locating a point in the master camera image defines a ray along which the point must lie in 3D space. This line can be projected onto the image plane of the slave camera to narrow down the point's possible location. With potential error present in this ray, the ray becomes a band with width (at the point it meets the surface of the sample) of two times the 2D error (Figure 42). Creating this band of possible locations demonstrates how accumulative errors can be avoided; a point appearing outside this band effectively indicates the additional calibration error, allowing it to be corrected for (Figure 43).

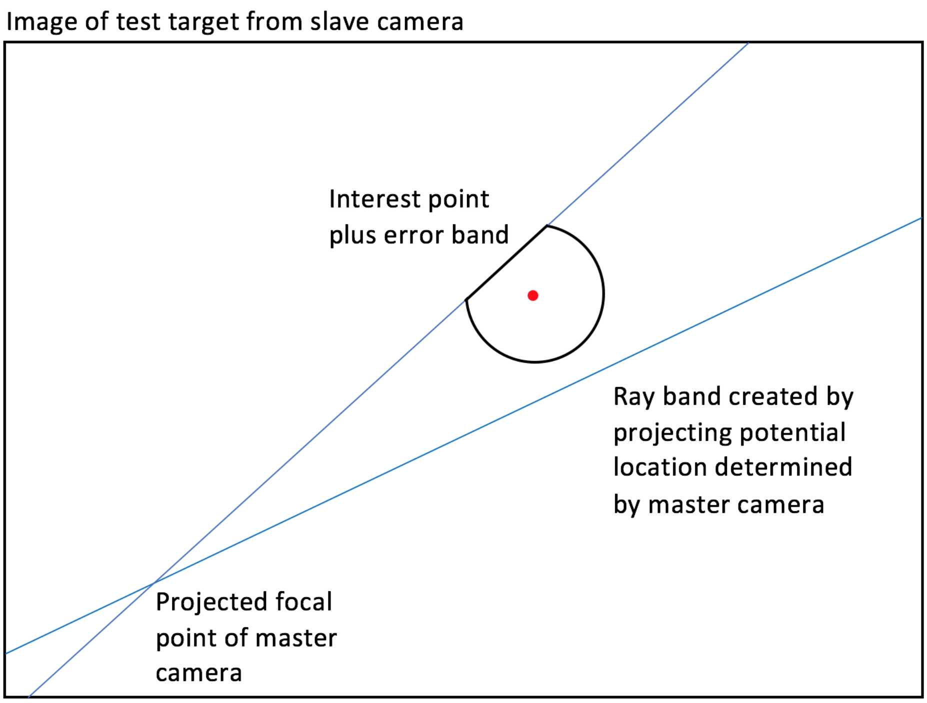


Figure 43: Point identified in master camera (left) matched to corresponding point in slave camera (right)

The minimum values of local strain detectable depend on the size of region over which they are measured (Table 7). The precision of the local strain that the device will be able to measure will therefore depend on the strain-rate and size of the defined local area.

5.2 Discussion

The aim of this project was to create a device that could measure local strains in order to detect the onset of shear-banding in triaxial tests. If possible, use of the device may have been extended to measure local small-strains.

Strain rate (%/min)	**Error in identified 3D location (±mm)**	**Minimum distance between distinguishable positions (±mm)**
0.712	0.0208	0.0416
0.0357	0.0147	0.0294
0.00357	0.0144	0.0289

Table 6: Minimum accurately distinguishable surface point displacement

	l = 40mm	l = 20mm	l = 10mm
r = 0.712(%/min)	0.104%	0.208%	0.416%
r = 0.0357(%/min)	0.074%	0.147%	0.294%
r = 0.00357(%/min)	0.072%	0.144%	0.289%

Table 7: Minimum local strain measurable at strain rates (r) and section lengths (l)

According to S. Yimsiri's 2002 paper [22], "for accurate measurement of stress-strain behaviour of soils, instrumentation capable of measuring strains to an accuracy of 10^{-3}% is required." Table 7 shows the minimum measurable local strain values when taken over a range of section lengths and strain rates. Even using a section length of 40mm (just under a quarter of the total sample length), the device is not capable of measuring strains at this order of magnitude.

Lower strain rates could be used in order to further minimise the impact of synchronisation error. This would not be ideal as the device would still be unreliable for higher strain-rate testing. The connections could instead be hardwired and the control system adapted to reduce synchronisation error, however this would result in an increasingly complicated device and a very difficult assembly process. Despite this, the limiting factor is the error due to calibration. Even if the synchronisation error could be reduced to approximately zero, the calibration procedure would have to provide calibration with maximum error of approximately ± 0.002 pixels. Improved camera focus would be necessary although at this point, any imperfections in the surface of the cell would potentially cause problems. In summary, this design of device is not suitable for measuring small strains.

In terms of shear band identification, the requirement of the device is to recognise differences in local strain across the sample surface between time steps. Triaxial test runs on Hostun HN31 sand (composed of angular quartz) suggest that, at formation, deformation within shear bands can account for up to 20% of a total axial strain of 1% [8]. This corresponds to a local axial shortening of 0.28mm in a 140mm length sample; easily detectable by the device at all analysed strain rates (Table 6). Under plane strain tests, soft clays have shown to have approximately 3-5% of 1% axial strain occurring within shear bands [21]. This corresponds to local axial shortening of 0.042mm - 0.070mm; just within sensitivity of the device at all strain-rates.

The device as currently configured is predicted to be appropriate for detecting the onset of shear band formation in triaxial tests [6].

[6] based on the sample test rig configuration using a 140mm by 70mm sample and a 200mm diameter cell

6 Conclusion

This project began the design and development of a commercially viable device for providing a holistic view of surface strains in triaxial testing. The purpose of the device was to identify the onset of shear bands during testing. At this point, soil samples begin to deform non-uniformly and strain values measured by the standard test no longer accurately represent the mechanisms within the test sample. It was also desired to test whether the same device would be suitable for small-strain analysis. The device in its final form should be easy to set up and operate such that the test can be run and results collected without specialist geotechnical knowledge. It should also be low cost.

The complete device design requires 36 Raspberry Pi Zero W cameras and V2 cameras, Raspberry Pi Zero camera cables, 'Adafruit' camera cases for each camera, power cables and USB hubs, 18 3D prints of a camera mount, a calibration target, a calibration board pattern and a wireless router. All are standard components apart from the camera mounts, the calibration target and the calibration board patterns. Camera mounts can be created by downloading the CAD pattern and 3D-printing the design. The required calibration target is a metal cylinder matching the shape of the intended test sample with a screw attachment suited to the triaxial apparatus. This is a simple component that can be milled in any workshop with milling equipment. Workshops do exist where this work can be outsourced. The calibration board is a Charuco board pattern printed onto sticky-back vinyl. The pattern can be downloaded and sent to a commercial printing company for printing. This design has no high cost components. The only specialised knowledge/equipment required would be for the calibration target.

The mount is a modular design in the form of a CAD file that can be 3D printed. Printing at 100 micron resolution using PLA filament results in a sufficient quality product. The camera mount is designed to hold the arm of an 'Adafruit Raspberry Pi case'. To minimise the slip in the camera focus, the tape between the PCB and the camera module should be removed. The mounts provide sufficient stability to prevent slipping when not touched. The design ensured the minimum distance between the camera lens' and the triaxial cell wall, minimising image distortion. Due to the design, achieving perfect focus of the cameras was difficult and this limited the precision of the calibration process. Soft power cables should be used and it is recommended not to use cases for the Raspberry Pi boards themselves (specifically the 'Adafruit' design) to minimise potential damage.

Control of the device is managed through a control laptop using a 2.4GHz LAN. Beyond the initial set-up of the device, this is the only control mechanism required. The control system enables connection to all set-up Raspberry Pis, sends instructions for images to be captured (both for calibration and during testing) and collects data. Synchronisation testing of the device suggested a range of approximately 0.768s in capture time for one set of images. This was largely due to a small number of outlying points caused by overcrowding on the network and could potentially be avoided by operating the wireless router on the 5GHz frequency band.

The precision of calibration with Charuco boards was limited by the physical difficulty in focusing the cameras. Although calibration with Circuco boards may have minimised the lack of precision in point detection that this caused, the error introduced by projecting circle centres onto oblique planes was larger than the improvements they might have made. Calibration using one angle view was determined to be an inappropriate method as occasional misidentification of image points may lead to poor calibration. This effect could also pass undetected due to the method used by OpenCV to measure calibration error. Instead, multiple images of different views of the Charuco board should be used to calibrate one camera. After removing outlying image points, a small but consistent difference was present in the distribution of remaining error in the x direction and y direction. This was potentially a result of the cell surface curvature. By applying an independent warp in x and y based off one run of the calibration algorithm, this effect could be minimised.

The optimal calibration method for this device is a Charuco calibration procedure with optional independent x-y warping and outlier removal. The calibration method proposed would consist of a run of ten images taken by all cameras whilst mounted. The calibration target would be mounted and the cell filled with water. The calibration target can then be twisted between image sets by the arm protruding from the top of the triaxial cell. This process is automatic and does not require specialist knowledge to run.

In order to measure small-strains, both calibration and synchronisation would need to be significantly improved. The calibration error would need to be reduced by at least two orders of magnitude. Even then, using a wireless connection, it is highly unlikely that the synchronisation could be improved by the necessary amount to refine strain calculations. Wired connections would be required, making the device much less viable for wide-scale use. The design of this device is therefore unlikely to be appropriate for small-strain analysis.

Using the methods of control and calibration recommended, this device is expected to be sufficient to identify variations in strain on a triaxial sample surface and subsequently detect the formation of shear bands. The device is low cost and requires minimal specialist knowledge or skill to set up and run. This would allow geotechnical laboratories to integrate this device into standard triaxial tests.

7 Further work

A number of steps are required in order to bring the device to full functionality:

- Code for 3D reconstruction based off calibrated images is required in order to map out a 3D set of points defining the sample surface.
- PIV software should be applied to calibrated images, tracking the displacement trajectories of the surface points over time.
- Code should be written to map the measured 2D trajectories to 3D in order to complete a 3D plot of the sample surface distortion over time.
- The 3D model can subsequently be used to derive axial and radial strains over the surface of the sample. Ultimately, the strain variation map will identify the point at which shear bands form as strains become non-uniform.

References

[1] Actiontec. 2.4ghz vs 5ghz wifi, 2020 (accessed 06/05/2020). https://www.actiontec.com/blog/wifi/wifi-wednesday-2-4ghz-vs-5ghz-wifi/.

[2] Hight D. Baldi, G. and G. Thomas. A reevaluation of convential triaxial test methods. advanced triaxial testing of rock and soil. *Advanced Triaxial Testing of Soil and Rock*, pages 219–263, 1988.

[3] Athma Bhandari, William Powrie, and Richard Harkness. A digital image-based deformation measurement system for triaxial tests. *Geotechnical Testing Journal*, 35:209–226, 03 2012.

[4] S. F. Brown and M. S. Snaith. The measurement of recoverable and irrecoverable deformations in the repeated load triaxial test. *Geotechnique*, 24(2):255–259, 1974.

[5] P. Bésuelle. Evolution of strain localisation with stress in a sandstone: Brittle and semi-brittle regimes. *Laboratoire de Geologie, Ecole Normale Suptrieure, Paris, France*, 26(1-2):101–106, 2000.

[6] R. Cipolla. Module 4f12: Computer vision. handout 3: Projection. *CUED University of Cambridge*, 2019.

[7] Octavio Coronado, Bernardo Caicedo, Said Taibi, Antonio Correia, Hanène Souli, and jean-marie Fleureau. Effect of water content on the resilient behavior of non standard unbound granular materials. *Transportation Geotechnics*, 7:7, 04 2016.

[8] Andò E. Mevoli F. A. Debove L. Desrues, J. and G. Viggiani. How does strain localise in standard triaxial tests on sand: Revisiting the mechanisms 20 years on. *Mechanics Research Communications*, 92:142–146, 09 2018.

[9] J. Desrues and R. Chambon. Shear band analysis and shear moduli caibration. *International Journal of Solids and Structures*, 39:3757–3776, 02 2002.

[10] Raspberry Pi Foundation. Noobs, 2020 (accessed 05/05/2020). https://www.raspberrypi.org/downloads/noobs/.

[11] S. Haigh. 3d2 geotechnical engineering ii. *Cambridge University Engineering Department*, page 28, 2018.

[12] C. Hird and P. Yung. The use of proximity transducers for local strain meassurements in triaxial tests. *Geotechnical Testing Journal ASTM*, 12:292–296, 1989.

[13] R. J. Jardine, M. J. Symes, and J. B. Burland. The measurement of soil stiffness in the triaxial apparatus. *Geotechnique*, 34(3):323–340, 1984.

[14] K. Mordvintsev, A. Abid. Harris corner detection, 2013 (accessed 29/04/2020). https://opencv-python-tutroals.readthedocs.io.

[15] Ibsen L. B. Peri, E. and B. N. Nielsen. Influence of sample slenderness and boundary conditions in triaxial test - a review. *E3S Web of Conferences*, 92(02009), 2019.

[16] A. Rechenmacher and Z. Medina-Cetina. Calibration of soil constitutive models with spatially varying parameters. *Journal of Geotechnical and Geoenvironmental Engineering - J GEOTECH GEOENVIRON ENG*, 133, 12 2007.

[17] H. Rhody. Lecture 10: Hough circle transform. *Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology*, 2005.

[18] Frost J. Lo Presti D. Scholey, G. and M. Jamiolkowski. A review of instrumentation for measuring small strains during triaxial testing of soil specimens. *Geotechnincal Testing Journal, ASTM*, 18(2):137–156, 1995.

[19] P. Shapally. An optical method for strain measurement in triaxial tests. *University of Western Australia*, 2017.

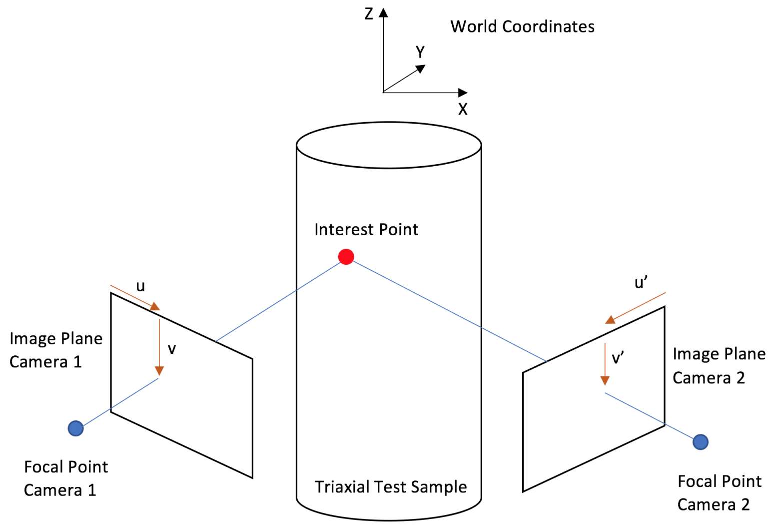
[20] OpenCV Dev Team. *Camera Calibration and 3D Reconstruction*, 2014 (accessed 28/04/2020). https://docs.opencv.org.

[21] V. Thakur. Experimentally observed shear bands in a scandinavian soft clay subjected to an undrained shearing under the plane strain condition. *Department for Engineering and Natural Sciences, Aalesund University College, Aalesund, Norway*, 09 2014.

[22] Siam Yimsiri and K. Soga. A review of local strain measurement systems for triaxial testing of soils. *Geotechnical Engineering, SEAGS*, 33:42–52, 04 2002.

Appendices

A Calculating real world coordinates from two sets of image coordinates



Camera matrices and rotation translation matrices are known for both cameras as a bi-product of the Charuco calibration method. Image point locations (u,v) and (u',v') are known.

$$s\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}f_x & 0 & c_x\\0 & f_y & c_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_1\\r_{21} & r_{22} & r_{23} & t_2\\r_{31} & r_{32} & r_{33} & r_3\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

$$s\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}r_{11}f_x + r_{31}c_x & r_{12}f_x + r_{32}c_x & r_{13}f_x + r_{33}c_x & t_1f_x + t_3\\r_{21}f_y + r_{31}c_y & r_{22}f_y + r_{32}c_y & r_{23}f_y + r_{33}c_y & t_2f_y + t_3\\r_{31} & r_{32} & r_{33} & t_3\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

$$s\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}p_{11} & p_{12} & p_{13} & p_{14}\\p_{21} & p_{22} & p_{23} & p_{24}\\p_{31} & p_{32} & p_{33} & p_{34}\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

$$u = \frac{su}{s} = \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}$$

$$v = \frac{sv}{s} = \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}$$

$$0 = X(p_{11} - p_{31}u) + Y(p_{12} - p_{32}u) + Z(p_{13} - p_{33}u) + (p_{14} - p34u)$$

$$0 = X(p_{21} - p_{31}v) + Y(p_{22} - p_{32}v) + Z(p_{23} - p_{33}v) + (p_{24} - p34v)$$

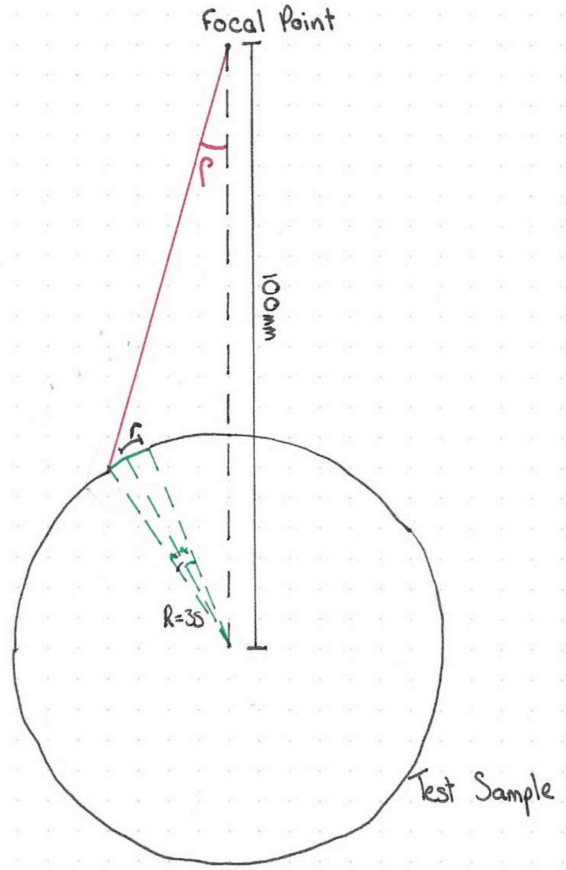
$$0 = X(p'_{11} - p'_{31}u') + Y(p'_{12} - p'_{32}u') + Z(p'_{13} - p'_{33}u') + (p'_{14} - p34'u')$$

$$0 = X(p'_{21} - p'_{31}v') + Y(p'_{22} - p'_{32}v') + Z(p'_{23} - p'_{33}v') + (p'_{24} - p34'v')$$

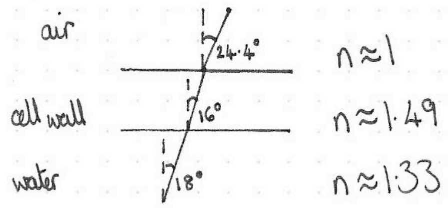
$$\begin{bmatrix} p_{11} - p_{31}u & p_{12} - p_{32}u & p_{13} - p_{33}u & p_{14} - p34u \\ p_{21} - p_{31}v & p_{22} - p_{32}v & p_{23} - p_{33}v & p_{24} - p34v \\ p'_{11} - p'_{31}u' & p'_{12} - p'_{32}u' & p'_{13} - p'_{33}u' & p'_{14} - p34'u' \\ p'_{21} - p'_{31}v' & p'_{22} - p'_{32}v' & p'_{23} - p'_{33}v' & p'_{24} - p34'v' \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0$$

These equations can be solved for X, Y and Z using a least squares approach.

B Calculation of error caused by circle projection

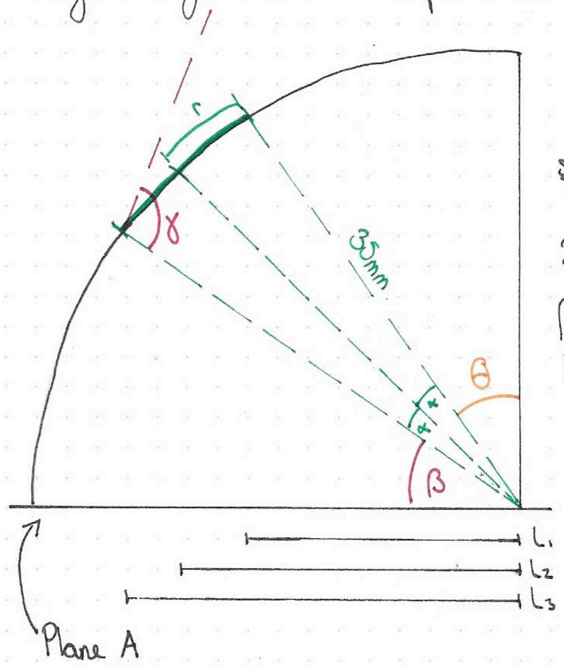


Ignoring the cell wall thickness and any distance between the cell wall and focal point



Assuming the cell wall has refractive index close to that of perspex, and as no partial blobs were identified in calibration, we shall assume $\rho_{max} = 18° = \frac{\pi}{10}$ rad

Project surface circle to a plane at the sample centre



$r = \frac{2\pi R}{2\pi} \times \alpha \Rightarrow \alpha = \frac{r}{R} = \frac{r}{35}$

$\frac{\sin(\gamma)}{100} = \frac{\sin(\rho)}{35} \Rightarrow \gamma = \pi - \sin^{-1}\left(\frac{100}{35}\sin(\rho)\right)$

$2\alpha + \theta = \pi - (\gamma + \rho)$

$\beta = \frac{\pi}{2} - (2\alpha + \theta) = \gamma + \rho - \frac{\pi}{2} = \rho + \frac{\pi}{2} - \sin^{-1}\left(\frac{100}{35}\sin(\rho)\right)$

$L_1 = 35\cos(2\alpha + \beta)$

$L_2 = 35\cos(\alpha + \beta)$

$L_3 = 35\cos(\beta)$

Projected length of LHS $= L_3 - L_2$

Projected length of RHS $= L_1 - L_2$

Distance on Plane A between centre of projected circle and ellipse:

$$= \frac{1}{2}(L_3 - L_1) - (L_3 - L_2) = D$$

Distance converted to pixels (assuming sample curvature negligable here)

$$= D \div \text{length of plane visible} \times \text{number of pixels across plane}$$

$$= D \div (2 \times 100 \times \tan(\rho_{max})) \times 2454$$

$$= 37{\cdot}763D$$

	r = 2.5mm	r = 1mm	r = 0.5mm
$\rho = 9$	0.78426	0.14789	0.03882
$\rho = 10.8$	1.01216	0.18390	0.04778
$\rho = 12.6$	1.25503	0.22214	0.05729
$\rho = 14.4$	1.51888	0.26352	0.06756
$\rho = 16.2$	1.81398	0.30955	0.07896
$\rho = 18$	2.16197	0.36345	0.09228

Table 8: Pixel difference between centre of ellipse in image and centre of projected circle

Appendix C: Covid-19 disruption

Due to Covid 19 I was not able to increase the size of my image set for calibration. Ideally this would have taken place over the Easter holidays, potentially between the 18th March – 1st April. This could not be done due to access restrictions and having returned home.

The images I had taken were initially collected solely for creating and improving the calibration algorithm rather than for testing its performance. Therefore, the realisation during analysis that one-image calibration would not be suitable could not be addressed. This was especially true of the data collected and used for the Circuco boards as only one view-angle was available for each camera. If it had been possible, ten images from each camera would have been taken of the calibration target, all at different angles.

It would also have been ideal to have acquired an updated Circuco board with smaller circles and test the calibration results. It was discovered during early analysis this would have reduced the inherent circle-projection error to the point that it would have been comparable to Charuco calibration. An expected result from this may have been that the errors introduced to Charuco calibration by lack of camera focus were not present in Circuco calibration. With small circles, the expected minimum achievable error may have been smaller than that achieved with the Charuco method.

This realisation occurred early April and a board could have been ordered any time from then onward. The work involved in this would be minimal however required access to the NCEB and labs which was impossible at the time. Due to the delay in delivery times of the previous board prints however, it is not possible to say whether this would have been achievable in the time or not.

The focus of the calibration section therefore moved more towards analysing the best measure of calibration error rather than minimising the error. The project also extended and covered the potential performance of the device given how it was performing at the point of leaving Cambridge.

I also only have access to my logbook for Michaelmas term. The papers I used for Lent term are still in Cambridge and I do not have access to them.