

In Vivo Lensless Microscopy Using A Raspberry Pi Camera



Cameron Ray
(RYXCAM002)

Department of Electrical Engineering
University of Cape Town
Rondebosch, Cape Town
South Africa

*Supervisor: Robyn Verrinder
Co-Supervisor: Dr Mandy Mason*

October 2022

EEE4022S final year project report submitted in partial fulfilment of the requirements
for the degree of **Bachelor of Science in Mechatronics** in the Department of
Electrical Engineering at the University of Cape Town

Keywords: microscopy; real-time; lensless

Declaration

I, Cameron Ray, hereby:

1. grant the University of Cape Town free licence to reproduce the above thesis in whole or in part, for the purpose of research;
2. declare that:
 - (a) this thesis is my own unaided work, both in concept and execution, and apart from the normal guidance from my supervisor, I have received no assistance except as stated below;
 - (b) neither the substance nor any part of the above thesis has been submitted in the past, or is being, or is to be submitted for a degree at this University or at any other university, except as stated below.



Cameron Ray

Department of Electrical Engineering
University of Cape Town

Sunday 30th October, 2022

Abstract

This thesis investigates the application of lensless microscopy in the field of Mycobacterium tuberculosis drug research. The design of a lensless microscope and supporting hardware was carried out to yield a real-time system for bacterial colony counting and growth rate estimation. The aim of this research was to determine the feasibility of implementing the Sony IMX477 CMOS Image sensor in a lensless microscope configuration. It is assumed that this camera will be more than suitable for this application. This is due to the large FOV and image sensor area. The image sensor was used on the Raspberry Pi High-Quality camera PCB which was connected to a Raspberry Pi 4B for image processing. The Raspberry Pi is also used for interacting with a real-time sample analysis interface. The tools and interfaces were developed using Python and HTML. They are interacted with via a web portal hosted on a Flask server. Backend image processing and computer vision analysis are accomplished by real-time software which implements the HoloPy and OpenCV libraries. It was found after implementing the Raspberry Pi Camera that it is suitable for the application of lensless microscopy. The project was successful overall although minor issues in some subsystems were discovered. This includes the limitations on image reconstruction and the bacteria samples that did not grow within a 24-hour window.

Acknowledgements

I want to thank my supervisor Robyn Verrinder for their guidance and continuous assistance throughout the project; my co-supervisor, Dr Mandy Mason, for their interest and support in testing as well as for informing me about the exciting research that is undertaken in their lab; Brendon Daniels and Justin Pead for their willingness to ensure that I could 3D print on campus despite difficulties during loadshedding; and lastly, Dr Caron Jacobs and Michael Woodward for their assistance in the IDM laboratory with regards to test-targets, sample preparation and general discussion around microscopy which they are both involved in at UCT.

A special mention must go to Fraser Montandon, for their advice on optics, and Amaan Vally, for their support on specific aspects of this project's hardware development and design.

Contents

Declaration	i
Abstract	ii
Acknowledgements	iii
Contents	vi
List of Figures	ix
List of Tables	x
Glossary	xi
Chapter 1 Introduction	1
1.1 Background	1
1.2 Motivation	2
1.3 Problem statement	2
1.4 Project objectives	2
1.4.1 Research	2
1.4.2 Design	2
1.4.3 Experimentation	3
1.5 Scope, limitations and assumptions	3
1.5.1 Scope	3
1.5.2 Assumptions	3
1.5.3 Limitations	4
1.6 Plan of development	4
Chapter 2 Literature Review	5
2.1 Literature Review	5
2.1.1 Introduction	5
2.1.2 Lighting Techniques	5
2.1.3 Image Sensor Types	7
2.1.4 Sample Placement	8
2.1.5 Reconstruction Algorithm	9
2.2 Suggested Approach	10
2.2.1 Introduction	10
2.2.2 Lighting Techniques	10
2.2.3 Image Sensor	10
2.2.4 Sample Placement on the Sensor	11

2.2.5	Reconstruction Algorithm	11
Chapter 3	Hardware Development	12
3.1	Introduction	12
3.2	Methodology	12
3.2.1	3D Component Modelling	12
3.2.2	3D Printing	14
3.3	Design	14
3.3.1	Sample Slide Mount	14
3.3.2	LED and Guide Housing	16
3.3.3	Raspberry Pi Housing	18
3.4	Summary of Development	20
Chapter 4	Software Development	21
4.1	Introduction	21
4.2	Methodology	21
4.2.1	Initial Development and Testing	21
4.2.2	Using the Raspberry Pi Camera as a webcam	22
4.2.3	Developing the image reconstruction and colony counting software	22
4.2.4	Developing the web monitoring interface	23
4.3	Design	24
4.3.1	Image Reconstruction Software	24
4.3.2	Colony Counting Software	25
4.3.3	Web Monitoring Interface	27
4.4	Summary of Development	30
Chapter 5	Experimental Design and Setup	31
5.1	Introduction	31
5.2	Design	31
5.2.1	Experimental Setup	31
5.2.2	Experiment Objectives	36
5.3	Methodology	37
5.3.1	Preliminary Tests	37
5.3.2	Final Tests	37
5.3.3	Test Target Capture For Calibration	38
5.4	Summary of Experimentation	38
Chapter 6	Results	39
6.1	Introcduction	39
6.2	Subsystem Testing	39
6.2.1	Raspberry Pi High-Quality Camera Testing	39
6.2.2	Image Reconstruction Testing	41
6.2.3	Colony Counting Testing	42
6.2.4	Web Server Live View Testing	44
6.3	Lab Testing	45

6.3.1	18-Hour Growth Test	45
6.3.2	24-Hour Growth Test	46
6.3.3	5-Day Growth Test Using An Incubator	46
6.3.4	Test Target Capture	47
6.4	Summary of Results	48
Chapter 7	Discussion	49
7.1	Hardware	49
7.2	Software	50
7.3	Experiment	51
7.3.1	Bacterial Sample Growth	51
7.3.2	Resolution Analysis and Calculation	51
7.3.3	Objective Analysis	52
Chapter 8	Conclusions	54
Chapter 9	Recommendations and Future Work	55
9.1	In-depth research into reconstruction methods	55
9.2	Investigation into alternative camera solutions	55
9.3	Extended image capture in an incubator	56
9.4	Additional web server features	56
9.5	Scaling the lensless microscope	56
Appendix A		59
A.1	Link To GitHub	59
A.2	Link To Fusion 360 Models	59
A.3	Link To Captured Image Package	59
A.4	Ethics Approval	60

List of Figures

2.1 Light Source Types: (a) An example of an incoherent light source (Bishara, Su, et al. 2010); (b) An example of a partially coherent light source (Luo et al. 2015)	6
2.2 Light Source Configurations: (a) An example of a purely translational light source (Bishara, Su, et al. 2010); (b) An example of a rotational and translational light source (Luo et al. 2015); and (c) An example of a static light source configuration (Bishara, Sikora, et al. 2011)	6
2.3 Microscope Form Factors: (a) An example of the lighting and sensor configuration of a lensless microscope (Bishara, Sikora, et al. 2011); and (b) An example of a normal inverted microscope configuration (Wiklund, Brismar, and Önfelt 2012)	8
3.1 Mechanical drawing of the Raspberry Pi High-Quality Camera's PCB and Lens Mount (Ltd. 2020)	13
3.2 A more detailed mechanical drawing of the Raspberry Pi High-Quality Camera's PCB footprint(Martin and Foundation 2020)	13
3.3 Mechanical drawing of the slide mount part. Dimensions shown are in mm.	15
3.4 3D Overview of the slide mount and its associated parts.	16
3.5 3D Overview of the slide mount and its associated parts.	16
3.6 LED circuit and GPIO pinout configuration	17
3.7 Exploded 3D view of the LED housing	17
3.8 Mechanical drawing of the Raspberry Pi and LED housing. Dimensions shown are in mm.	18
3.9 Exploded view of the Raspberry Pi housing unit. The unit includes 6 side walls and a dummy PCB is placed in the model for sizing and reference.	19
3.10 An overview of the final design of the Raspberry Pi and LED housing unit with all components connected together.	19
4.1 A diagram displaying the functionality encapsulated by the Flask web server.	24
4.2 An example of a background image captured containing debris.	25
4.3 Colony Image Loading: (a) An example of an original colony image; (b) An example of the masked image; (c) An example of the Gaussian blurred image. Original image from Zaburdaev 2021	27

4.4	Colony Image Segmentation and Labelling: (a) Using the masked and blurred image, Canny edge detection is performed and can be seen here as a black and white output; (b) The original image is used to show the colony border overlays at the final output stage.	27
4.5	The landing page of the web interface for the microscope system.	28
4.6	The sample load page of the web interface for the microscope system.	29
4.7	A bacterial sample's associated Sample Class Diagram.	30
5.1	The slide mount is exposed by removing the LED housing. Removing the LED housing is necessary to clean and access the slide mount.	32
5.2	Side view of the experimental setup showing a side wall.	32
5.3	Front view of the experimental setup showing the side wall where USB inputs and an Ethernet port are found.	33
5.4	Side view of the experimental setup showing the side wall where power and video output are found.	34
5.5	Back view of the experimental setup showing the side wall where the Raspberry Pi's SD card can be removed from and where the LED circuit wiring exits the housing.	35
5.6	Diagram showing the distance measurement used to configure the reconstruction algorithm.	36
5.7	Cleaning the protective cover slip before and after the sample is viewed. .	37
5.8	The USAF test target to be used to find the imaging resolution (Optics 2022b)	38
6.1	A background image captured during initial testing of the camera.	40
6.2	An initial test of the camera using liquid culture.	40
6.3	An initial test of the camera using a culture prepared on agar.	41
6.4	A series of images of sample data provided by the HoloPy (Lab 2021a) library. Each image has been reconstructed by propagating a light wave back through them but at varying distances from the image sensor plane.	42
6.5	The colony counting algorithm is outlined in order from figure (a) to figure (e). Original image from Zaburdaev 2021	43
6.6	First Colony Count Test: (a) This baseline image is provided to the software for its colonies to be counted; (b) The software returns an image with the colonies highlighted.	44
6.7	Second Colony Count Test: (a) This baseline image is provided to the software for its colonies to be counted; (b) The software returns an image with the colonies highlighted.	44
6.8	Live viewing pane on the web server's index page.	45
6.9	First Growth Test: (a) An image of the initial state of the bacterial sample is captured; (b) The final state of the sample is captured after 18 hours using the same camera parameters as the first image capture.	45

6.10 Second Growth Test: (a) An image of the initial state of the bacterial sample is captured; (b) The final state of the sample is captured after 24 hours using the same camera parameters as the first image capture.	46
6.11 5-Day Growth Test: (a) Colonies of M. tb visible under 40x magnification; (b) At 100x magnification, the colony is confirmed by the rod shape of the bacterium.	47
6.12 USAF Test Target under white light for better visibility.	47
6.13 Additional test target capture to show the accuracy and resolution of images.	48
6.14 USAF Test Target under red LED light as would be seen under normal operation for the lensless microscope.	48
7.1 Minimum visible line pair group shown. Group 3,3 is the last set of line pairs visible.	53

List of Tables

2.1	Table of image sensors used across various past literature	7
3.1	Table of PrusaSlicer configuration parameters	14
7.1	Raspberry Pi High-Quality Camera Parameters	51

Glossary

CLI A text-based interface which accepts commands from the user to perform tasks on a computer or to run applications without a graphical user interface.. 22

FOV Generally used to describe the extent to which an observer can view the world around them.. 8

Mtb It is a species of pathogenic bacteria and the root cause of Tuberculosis.. 1

RGB A system used to describe a colour using three channels, namely: Red, Green and Blue.. 8

TB A disease caused by the Mycobacterium Tuberculosis bacteria.. 1

Chapter 1

Introduction

1.1 Background

Tuberculosis (TB) research has been rapidly developing and requires newer techniques and technologies to be developed often to combat its aggressive nature and the speed at which it mutates. *Mycobacterium tuberculosis* (Mtb), the bacteria responsible for TB, has proven difficult for researchers to understand fully. Current techniques take up to a week before researchers may begin to count bacterial colonies in the TB field. This is due to the slow growth rate of the bacteria in comparison to other pathogens. Additionally, resource and equipment constraints make it even more difficult to carry out colony counting efficiently.

The technique predominantly used in a lab environment to estimate the size of Mtb colonies through counting colonies and dividing by the dilution factor. This method relies on visual estimates under an optical microscope. The estimation can provide inaccurate results and impact the study for which the bacteria is being analysed. Ideally, researchers would prefer continuous, real-time recording systems that can estimate Mtb bacteria colony sizes with far greater accuracy than what is currently used.

Traditional microscopes have their image sensors placed at a fair distance away from the sample. In between the sensor and the bacterial sample, multiple lenses are used to reach the desired objective magnification. Microscopes such as these are expensive since the hardware is manufactured with high precision to ensure the optics are robust. The answer to lowering the cost and increasing the level of scalability of such a system is to use an alternative microscope design. Through the use of more powerful computers readily available, holographic images of a microscopic sample can be reconstructed to produce high-resolution images.

1.2 Motivation

This project provides an initial study into the viability of lensless microscopy for real-time bacterial growth tracking. The project aims to investigate alternatives to the existing method of manual colony counting found in UCT's Department for Infectious Disease and other research labs. The partnership between this department and the Department of Electrical Engineering is necessary to develop a system which meets the necessary specifications outlined by researchers in the infectious diseases field.

1.3 Problem statement

Current drug studies use low-cost manual observation methods to determine their susceptibility to bacterial cultures. These methods are relatively low cost; however, they are limited by biomass generation, observation and growth timelines, and estimates of colony size and counts. Further, microscopes used to obtain more accurate results are exceedingly expensive and congest the research flow since many researchers may only use the equipment in turn. Microscopy used in such a manner often provides fewer data to the researcher as a result of its periodic observation style in comparison to a real-time capture solution.

1.4 Project objectives

1.4.1 Research

This research will include a literature review of the lensless microscopy field and serves as a basis for understanding the techniques used in the relevant areas of the design. The research will also include a general review of current design techniques used when building such systems. This design research would include 3D printing, software development and UI design.

1.4.2 Design

The chief objective of the project is to design and develop the following:

- A 3D printed housing

- Image reconstruction software
- Bacterial colony counting software
- Flask web framework serving an interface for interacting with the microscope system

The 3D printed housing will host a Raspberry Pi 4B, the Raspberry Pi High-Quality Camera and an LED lighting system. The accompanying software will process the captured image data from the High-Quality Camera and provide an easy-to-use interface with which one can interact with the microscope remotely.

1.4.3 Experimentation

Experimentation will involve testing all the subsystems of the hardware and software. This will include running basic validation on sample imagery and setting up the device to determine if it can be operated remotely.

Once these tests are complete, bacterial samples will be prepared in a safe lab environment and will then be viewed in the microscope's viewing area. This test will consist of taking images over a period of time as well as initial images of test targets for determining the microscope's resolution.

1.5 Scope, limitations and assumptions

1.5.1 Scope

This project will investigate the use of a High-Quality Raspberry Pi camera as a means to count and measure bacterial colonies in real time. The project will aim to deliver a working lensless microscope. Although lab testing is not necessarily the end goal, it would provide beneficial insight into further research applications and the efficacy of the lensless microscope in a lab environment.

1.5.2 Assumptions

Firstly, a non-mechanical approach to the solution will be the most successful methodology. This is due to the vibrations mechanical systems impart on the imaging area and the sample it is capturing. Secondly, components should arrive within their expected lead times due to prior stock availability research that was completed.

1.5.3 Limitations

This project will be limited by time since there is only a 12-week period within which literature review, design and fabrication and reporting need to be delivered. Components may also be another source of limitation due to a widespread silicon shortage. Although this has had very little impact thus far, changes in shipping or supply could vary depending on demand.

1.6 Plan of development

1. Chapter 2: Literature Review

Provides an in-depth investigation into the existing literature in the field of lensless microscopy and holographic image reconstruction.

2. Chapter 3: Hardware Development

The methodology and design of the hardware elements are outlined here to show the process undertaken to manufacture a working prototype.

3. Chapter 4: Software Development

Similar to the hardware chapter, a detailed outline of the software design is given with a brief methodology discussing its development.

4. Chapter 5: Experimental Design and Setup

This chapter details the overall setup of the microscope, in terms of both hardware and software, and the experimental objectives and methodology.

5. Chapter 6: Results

The final experiment results are given in this chapter. The facts outlined here will be discussed in the following chapter.

6. Chapter 7: Discussion

Hardware, Software and Experimental outcomes are discussed and evaluated against the objectives outlined in 1.4

7. Chapter 8: Conclusion

A brief conclusion is provided to evaluate any initial assumptions. Additionally, an overall outcome of the project will be given.

8. Chapter 9: Recommendations and Future Work

Lastly, a list of recommendations and future work on the project is given to allow for the project to continue on and to highlight areas of improvement that may need to be focused on.

Chapter 2

Literature Review

2.1 Literature Review

2.1.1 Introduction

The relevant literature surrounding this topic extends from 2010 until 2020. Previous use cases were identified; however, few researchers' studies were continued. The general methodology of the research papers discussed in this literature review follows the approach of holographic reconstruction of the image sequence captured. The types of sensors used to capture image data vary, but most, if not all, perform the same task while providing varying resolutions. This will be discussed below as well as the techniques used to reconstruct the image of a microscopic sample.

2.1.2 Lighting Techniques

Most studies make use of overhead lighting techniques. This can be done in two ways: either a single lighting source back-illuminates the sample while a mechanical sub-structure displaces the sensor causing the light source to fall at different x-y positions, or a matrix of LEDs connected to a microcontroller provides statically-positioned, sequenced lighting sources.

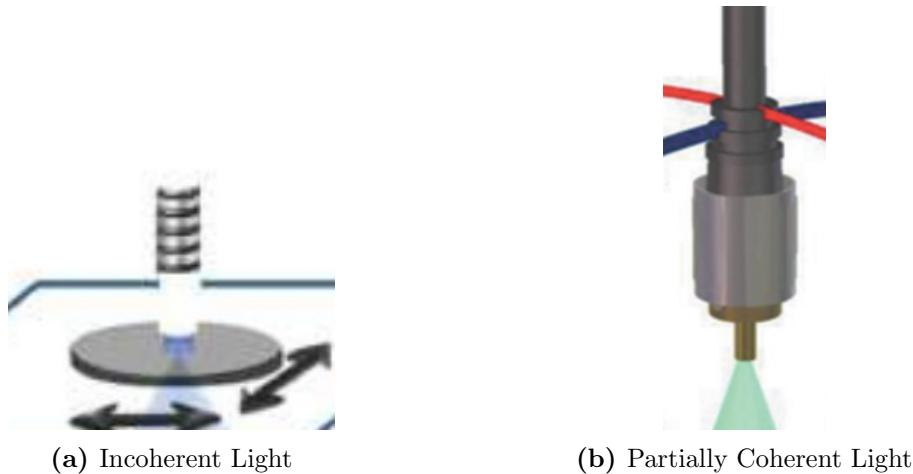


Figure 2.1: Light Source Types: (a) An example of an incoherent light source (Bishara, Su, et al. 2010); (b) An example of a partially coherent light source (Luo et al. 2015)

Partially coherent (Luo et al. 2015) or incoherent light (Bishara, Su, et al. 2010) sources are predominantly used as the light sources at the very top of the lighting system. Incoherent light sources are generated by lighting components with a large emitter surface area. An example of this would be an LED or a Xenon lamp (Bishara, Su, et al. 2010). These sources are then guided using single-mode glass fibre(s). Single-mode fibres allow light to propagate through them without any reflection against the cable's cladding. The light source is separated, using a physical dividing material (Bishara, Sikora, et al. 2011), from the sensor chamber of the experiment. This would most likely be necessary to prevent light leakage into the chamber.

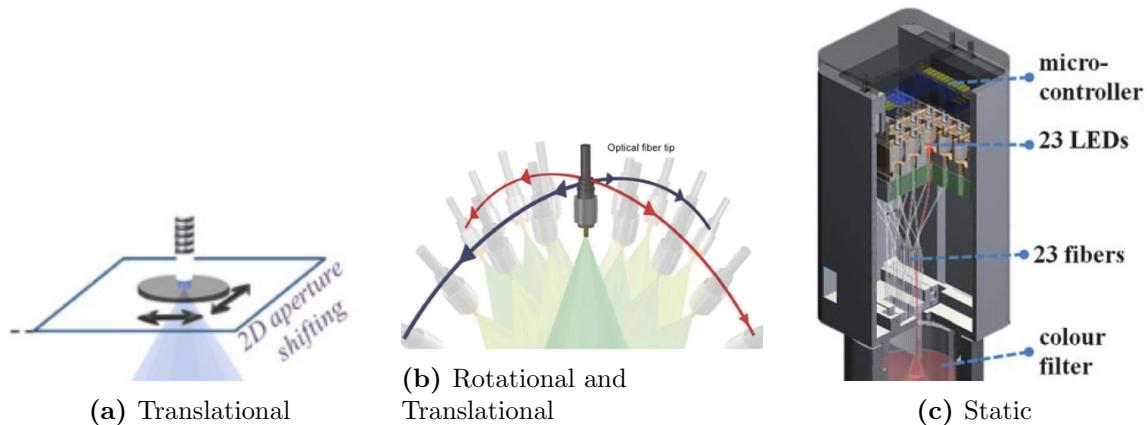


Figure 2.2: Light Source Configurations: (a) An example of a purely translational light source (Bishara, Su, et al. 2010); (b) An example of a rotational and translational light source (Luo et al. 2015); and (c) An example of a static light source configuration (Bishara, Sikora, et al. 2011)

The single light source solution requires a mechanical system (2.2a & 2.2b) to shift it. A popular technique to shift the single light source was observed to be a precise stepper-

motor system that could control the light source in the x-y plane only (Jiang et al. 2020) as seen in Figure 2.2a. This method is more suitable as it has a much simpler mechanical system than a rotational guide arm (Luo et al. 2015) as seen in Figure 2.2b. Rotating the source away from the system's z-axis introduces complexity to the system's design. However, a benefit observed from a rotational lighting arm is a consistent distance between the fibre optic guide's aperture and the objective plane. When a purely translational guide arm is used, the lighting source will introduce a differing aperture-objective distance. This could cause issues during calibration or reconstruction.

Multiple light sources placed statically in a known configuration 2.2c have proven to reduce the size of the overall system (Bishara, Sikora, et al. 2011). This is due to the lack of mechanical features such as stepper motors and their accompanying motor drivers. This lighting technique could be more desirable as it is expected to cause less error than a mechanically displaced system. Since the sources are in fixed positions, the system can be calibrated with higher accuracy. Only one research paper using this methodology was found and examined. It appears to be an under-researched methodology, and its outcomes were positive.

2.1.3 Image Sensor Types

All imaging sensors used in past research were of CMOS type. The CMOS sensors were widely available on component websites, which indicated that they were off-the-shelf components and were not developed by the authors of the literature in which they were used. The following table shows the specifications of image sensors used in the literature that was found:

Table 2.1: Table of image sensors used across various past literature

	Sensor Name	Sensor Size	Pixel Size
Bishara, Sikora, et al., Jung and Lee, Mudanyali et al.	Aptina MT9P031	5.7mm x 4.3mm	$2.2\mu m$
Jiang et al.	ON Semiconductor MT9J003	6.4mm x 4.5mm	$1.67\mu m$
Luo et al.	Sony IMX081PQ	6.52mm Diagonal	$1.12\mu m$

The pixel size of image sensors was considered heavily when choosing the sensor. Since all sensors are of similar dimensions, more pixels can fit onto the sensor die if the pixel area is smaller. This allows for higher-resolution images to be captured. Even though the pixel size is considered large for normal microscopy applications, it is suitable for capturing raw, lens-free images for the purpose of using reconstruction algorithms to boost the image resolution.

All the sensors used are Red-Green-Blue (RGB) type. This means that multiple colour channels are captured when the image sensor is active. This can provide benefits to the researcher if analysis of the image's colour spectrum is necessary but will most often be a drawback to the system. Since the image is processed in a Greyscale format, a conversion is done in most papers to convert the colour channels into a single channel. This is due to the limitations of the reconstruction algorithms and their ability to process multi-channel image data.

Due to the sensor and pixel sizes, a wide Field-Of-View (FOV) is achieved (Bishara, Su, et al. 2010). When compared to microscopes in UCT's labs, the image sensor provides a much larger viewing area but lacks precision, magnification and resolution. The image sensors found in the high-end microscopes are of superior quality and provide the best-in-class imaging. The aim of the literature was not to view microscopic samples at a high resolution however but rather focus was given to the reconstruction software in all cases to produce a better image quality. The load and the cost are then taken off the microscope itself and placed on the limits of the processor used.

2.1.4 Sample Placement

An inverted microscope format is used across all literature to produce the lensless microscope form factor. A comparison of an inverted microscope to a lensless microscope can be seen in the figure below:

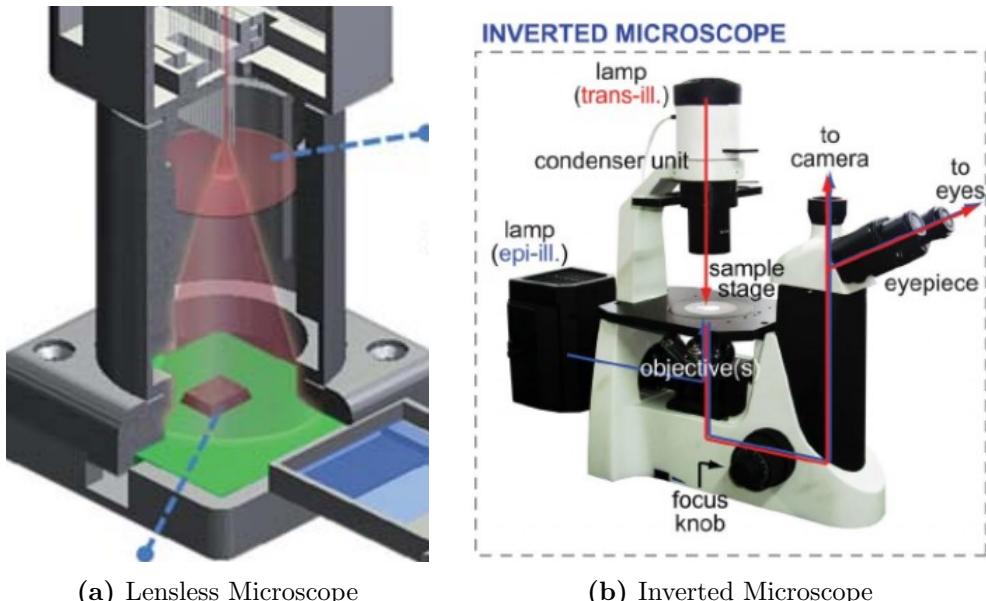


Figure 2.3: Micrsope Form Factors: (a) An example of the lighting and sensor configuration of a lensless microscope (Bishara, Sikora, et al. 2011); and (b) An example of a normal inverted microscope configuration (Wiklund, Brismar, and Önfelt 2012)

Through multiple configurations, one factor remains constant: lighting is provided from an overhead source while the sample is placed directly over the image sensor. The sample should ideally be placed directly on the image sensor, however, the fragility of some of the image sensors means that the sample could damage it permanently. As a compromise, the sample can be placed within a millimetre of the sensor such as in Bishara, Sikora, et al. and Mudanyali et al. On the contrary, Jung and Lee was able to place the sample directly in contact with the sensor. This may be due to the CMOS sensor's properties or protective layers, although, their literature and the corresponding datasheet for the sensor do not specify the protections in place to prevent damage.

In Bishara, Sikora, et al. 2011 a convenient object slide is used to place the sample over the image sensor after preparation. An object slide can be useful as it prevents damage to the sensor and the overall system since the bacterial sample can be prepared in a different location and then placed into the imaging chamber. Literature such as Bishara, Su, et al. 2010 and Luo et al. 2015 do not go into detail on the specifics of the hardware that was implemented to capture images. They merely discuss the techniques used to capture images and the reconstruction process.

2.1.5 Reconstruction Algorithm

Super-Resolution by Sub-Pixel Sweeping/Shifting

(Bishara, Su, et al. 2010) makes use of multiple low-resolution images shifted with respect to each other to generate a higher-resolution image. At the boundary of objects in a hologram, there exists diffraction. This diffraction causes the edges of the objects to appear blurred. By shifting a light source, multiple holograms can be obtained and used in a superposition operation to compute a higher-resolution image. The super-pixel algorithm uses a weighted-superposition formula and a reconstruction cost function. The cost function is minimised based on input parameters and image data.

Kirchoff–Helmholtz Reconstruction Algorithm

(Jericho and Kreuzer 2010) uses the Kirchhoff–Helmholtz transform in its implementation instead of a superposition and cost minimisation. When executing the Kirchoff–Helmholtz reconstruction, a 2D image is used to produce a 3D image at the output. This is done by propagating a wavefront at different z-axis heights from the image sensor level. The 3D image can be separated into individual layers to discover at which slice the highest resolution image is found.

The algorithm is noted to be quite slow and tedious to implement. An alternative imple-

mentation is given in the Appendix of Jericho and Kreuzer 2010 in the form of five steps. These steps include using the Fourier transform, discretizing continuous operations and convolution instead of using the Kirchhoff–Helmholtz transform directly.

2.2 Suggested Approach

2.2.1 Introduction

The suggested approach outlines the key findings of the literature that would be suitable to implement and fall within the scope of the project. Each section suggests an appropriate approach which may or may not change during the course of the project.

2.2.2 Lighting Techniques

The most common lighting technique used was a mechanical system. This creates unnecessary complexity and could introduce vibrations into the system. For the purposes of this project, it is suggested that a static light source be used. This light source may be configured in a few different patterns:

1. Grid of LEDs
2. Straight Line of LEDs
3. A Single, Centered LED

In each case, the LEDs will be placed behind or in a pinhole to create a point source light. To ensure that the point source of light is evenly distributed a thin fibre optic cable will be coupled to an LED to transmit the light through an aperture of known size with low tolerances on the fibre's width. Multi-mode fibre optic cable is suitable as it allows for multiple light rays to be shone at different angles. This will ensure that if the coupling is not a perfect tangency, the LED's light can still be reliably transmitted.

2.2.3 Image Sensor

Most, if not all, of the image sensors used across the various literature, are similar. They vary only slightly from each other in terms of unit cell size, total width and height, and imaging resolution. Since the Raspberry Pi Camera is based on a comparable image

sensor and is readily available off the shelf, it is decided that this Camera should be used in the final build.

2.2.4 Sample Placement on the Sensor

Most literature lacks a standard way of loading and placing a sample into their individual systems. However, Bishara, Sikora, et al. 2011 uses a simple tray system which to load and unload the sample. The tray will slide in a dedicated groove inside the mount. The tray should place the sample directly over the image sensor and should not be easily bumped or disturbed.

The sample cannot be placed directly on the sensor as this could damage it. It is important, however, to place the sample as close to the image plane as possible. A suggested way to solve this is to use a coverslip to protect the image sensor and minimise the air gap as best as possible.

2.2.5 Reconstruction Algorithm

Jericho and Kreuzer 2010 uses the Kirchoff–Helmholtz Reconstruction Algorithm to propagate light back through the images to generate the holograms. This same algorithm was implemented in a Python library called HoloPy (Lab 2021a). The HoloPy library was developed with reference to Jericho and Kreuzer 2010 specifically.

Using HoloPy allows for various functions in its toolset to be called to achieve a superpixel resolution image. These include:

1. Load an image from a file into xArray data type.
2. Load a background image from file.
3. Utilise sample images from the library itself.
4. Background correction functionality to subtract a background image.
5. Plane wave and point source light propagation.
6. Save an image to the file system or show its plot.

Chapter 3

Hardware Development

3.1 Introduction

This chapter outlines the methodology, design and development of the hardware for the project. The hardware involves a 3D-printed housing, GPIO interfacing and a USB connection. The system itself can be operated remotely and the software is hosted on it locally. The Raspberry Pi 4 Model B is used as the processing and data storage management device.

3.2 Methodology

3.2.1 3D Component Modelling

3D Components will be developed in Autodesk Fusion 360. Fusion 360 is a CAD program that allows for the modelling and design of parts that can be manufactured in industrial environments. It is useful to use this software as it is cloud-based and all work is backed up in case of issues with the computer. To use Fusion 360, download it from the Autodesk Website at <https://www.autodesk.com/products/fusion-360/overview>. Once the download and installation are complete, configure the workspace to the desired units or requirements for the task at hand.

Fusion 360 handles the parts of a design in a format called a component. Components will be developed by creating a sketch on a defined plane and extruding it in the desired fashion for the part that is being designed. Once the sketch is extruded, the component contains a new body object. This body object is a 3D structure which can be exported using Fusion 360's 3D print file generation tool. The body, once in an STL or similar file

format, can be used in the 3D printing process.

When designing the body needed it is important to refer to the necessary mechanical drawings of existing parts which will be incorporated into the system. Below are the two reference mechanical drawings (3.1 and 3.2) of the Raspberry Pi High-Quality Camera. These drawings will be used to accurately construct mounting holes or other structures which will be placed around the High-Quality Camera.

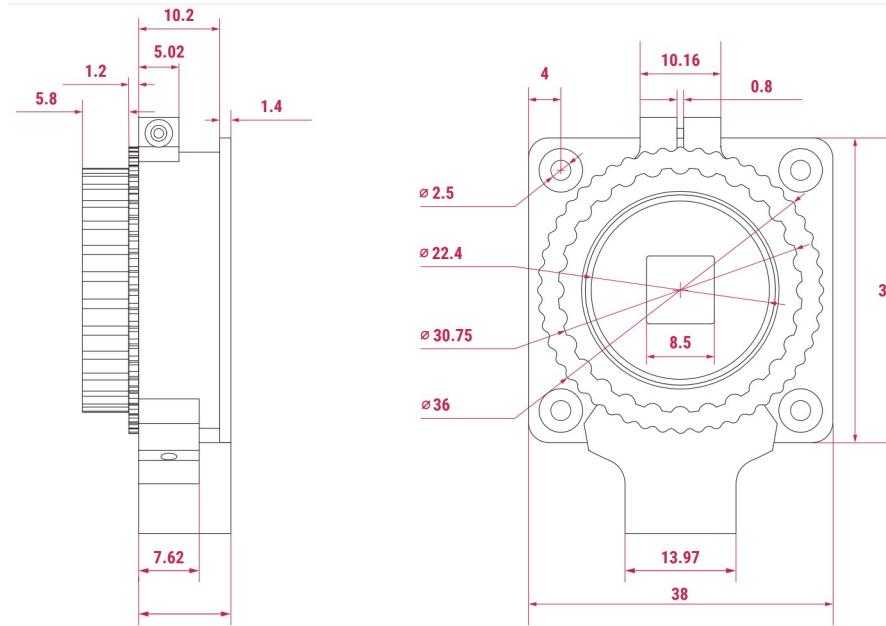


Figure 3.1: Mechanical drawing of the Raspberry Pi High-Quality Camera's PCB and Lens Mount (Ltd. 2020)

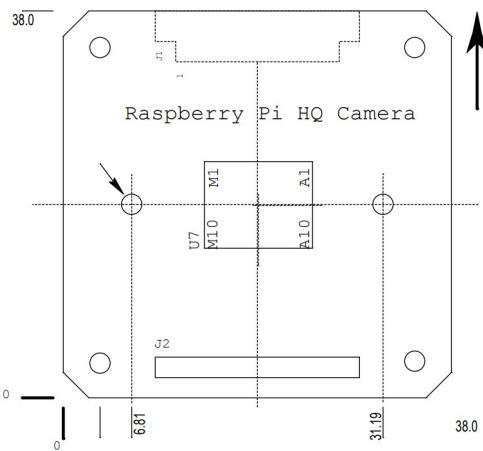


Figure 3.2: A more detailed mechanical drawing of the Raspberry Pi High-Quality Camera's PCB footprint (Martin and Foundation 2020)

The CSI to USB adapter (Arducam 2020) has the same footprint as the High-Quality Camera and thus it is sufficient to only use the High-Quality Camera's drawings as a reference as both parts will line up exactly.

3.2.2 3D Printing

The 3D Printing technology used is very simplistic and offers an inexpensive way to manufacture and scale parts in the future. The UCT Electrical Engineering Department has a set of Original Prusa MINI+ printers set up for use. The Prusa platform has been streamlined for easy user interaction and efficient fabrication. To print a 3D model generated by a CAD package on a Prusa 3D printer, the PrusaSlicer software must be downloaded from their website: https://www.prusa3d.com/page/prusaslicer_424/.

PrusaSlicer offers a 3D print configuration tool and a print preview display. The configuration has different levels of control ranging from Simple to Advanced. This allows first-time users to start printing almost immediately which makes it useful if parts need to be printed regularly for the lensless microscope. The main configuration settings needed to 3D print the lensless microscope STL files are the following:

Table 3.1: Table of PrusaSlicer configuration parameters

Parameter Name	Parameter Value
Filament Type	Pusament PLA
Support Inclusion	Everywhere (For parts with an overhanging section) or None (For all other parts)
Infill	15% (For parts that are not interacted with often) or 60% (For parts that are interacted with often)
Orientation	Set the object with the largest and flattest face pointing downwards

Further configuration can be done if necessary, however, the configuration listed in the table above (3.1) assists with printing the objects faster and more effectively. It is important to divide up the body into smaller parts to ensure that the builds are more successful. Larger components take longer to print and have a high chance of failure. A faster build means that print fails are not as costly.

3.3 Design

3.3.1 Sample Slide Mount

The sample slide mount was developed to replace the lens mount that is attached to the camera PCB off the shelf. This lens mount is useful for applications where lenses are necessary. Since the aim of the project was to develop a lensless microscope, it was

important to replace this part with something more useful. Based on a similar slide design to Bishara, Sikora, et al., a slide mount was developed.

The slide mount consists of three parts, namely: the slide mount body, the tray, and the limiter. The slide mount took a number of iterations to develop since the Raspberry Pi Camera mechanical drawings did not specify the exact sizing of the CMOS Sensor die. A final design was created around the approximate size of the CMOS die.

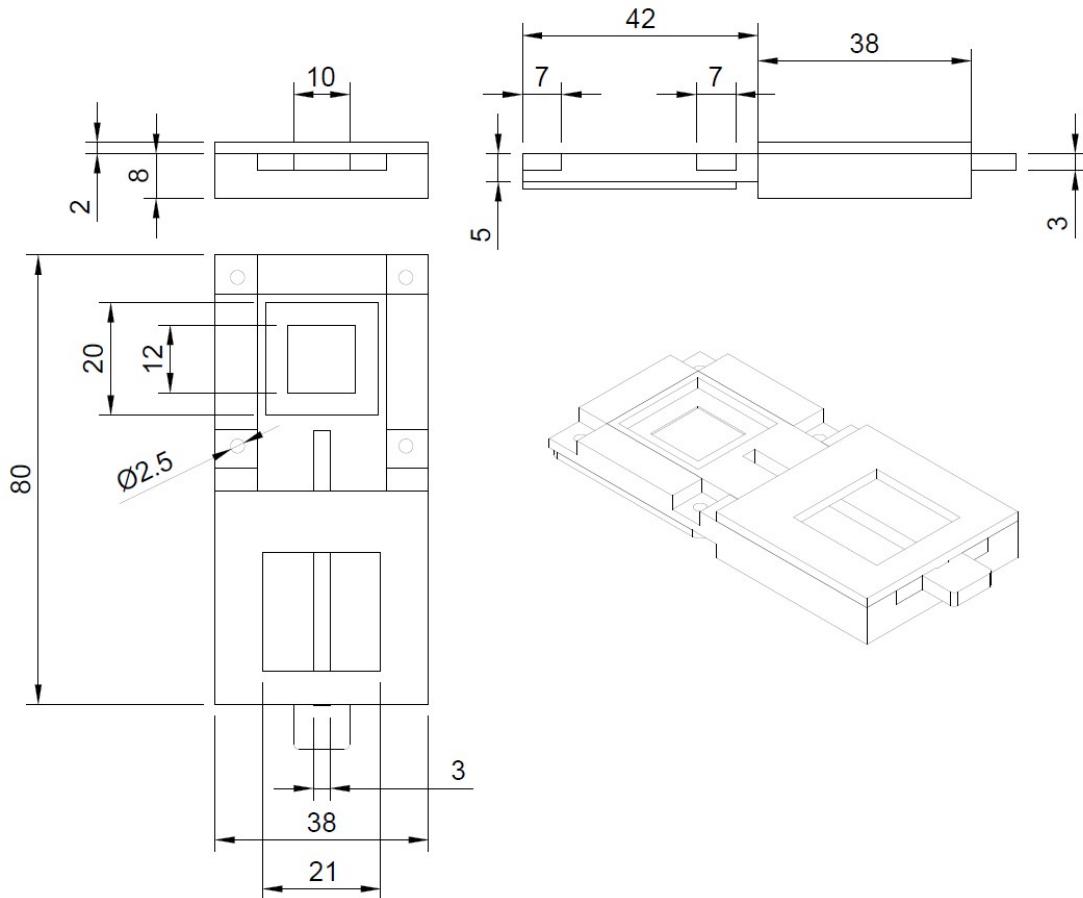


Figure 3.3: Mechanical drawing of the slide mount part. Dimensions shown are in mm.

In the slide mount exploded view below, part 1 is the limiter. This part has a stopper which prevents the tray from extending out too far while the limiter is in place. It also helps guide the tray smoothly along the slide mount body. Part 2 in the figure is the tray. It has a sample placement window 20mm x 20mm in width and height. This size window supports an 18mm x 18mm coverslip or a slightly larger gene frame. Part 3 is the main slide mount body and it holds all the parts together. It has a shallow groove where a protective cover slip is glued in place to protect the CMOS sensor and prevent debris from falling onto the sensor. Finally, part 4 shown here is a dummy PCB modelled to the same dimensions as the mechanical drawings provided by Raspberry Pi.

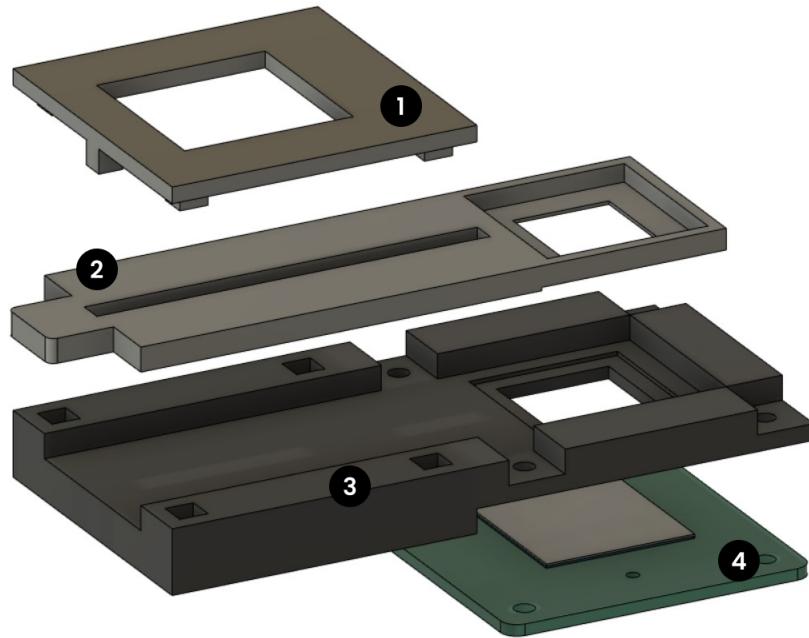


Figure 3.4: 3D Overview of the slide mount and its associated parts.

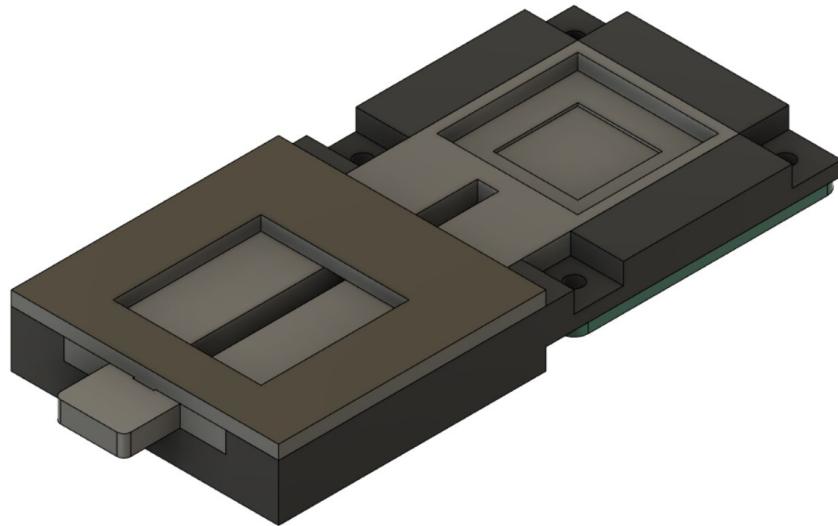


Figure 3.5: 3D Overview of the slide mount and its associated parts.

3.3.2 LED and Guide Housing

The LED system features a very simple LED-resistor combination to act as the point source of light in the system. The following circuit diagram shows the connections made to the Raspberry Pi and the GPIO pins used:

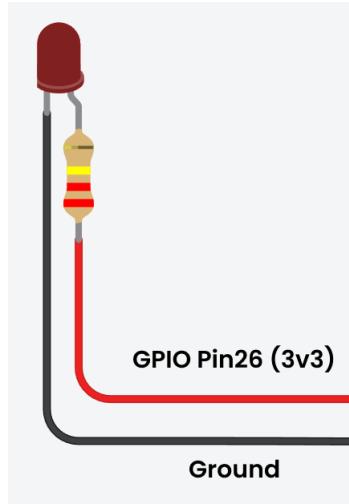


Figure 3.6: LED circuit and GPIO pinout configuration

The housing has an upper wall which has a large hole for the LED wiring to pass through. This wall, as seen in the figure below, prevents light leakage into the chamber. The LED fits into another hole inside the chamber. This hole is part of a shelf-like component visible in the figure below and within the 5 walls of the housing.

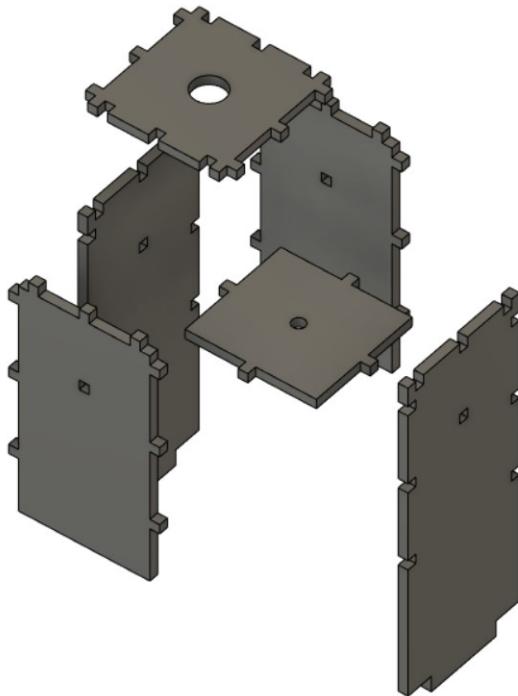


Figure 3.7: Exploded 3D view of the LED housing

3.3.3 Raspberry Pi Housing

The Raspberry Pi housing is a rectangular box which houses the Raspberry Pi and its cooling fan. The side walls with the larger cut-outs are placed on the sides where peripheral port access is necessary. During operation, the Raspberry Pi must be powered by its USB-C port and the CSI to USB adapter is plugged into the USB 3.0 ports.

The overall height of the system is 146mm which is smaller in comparison to most microscope form factors but is comparable in size to the device developed in Bishara, Sikora, et al. 2011. The housing uses a clip system to connect the printed components together. Each clip measures 3mm (L) x 3mm (W) x 3mm (H). The clips are securely placed into the matching grooves.

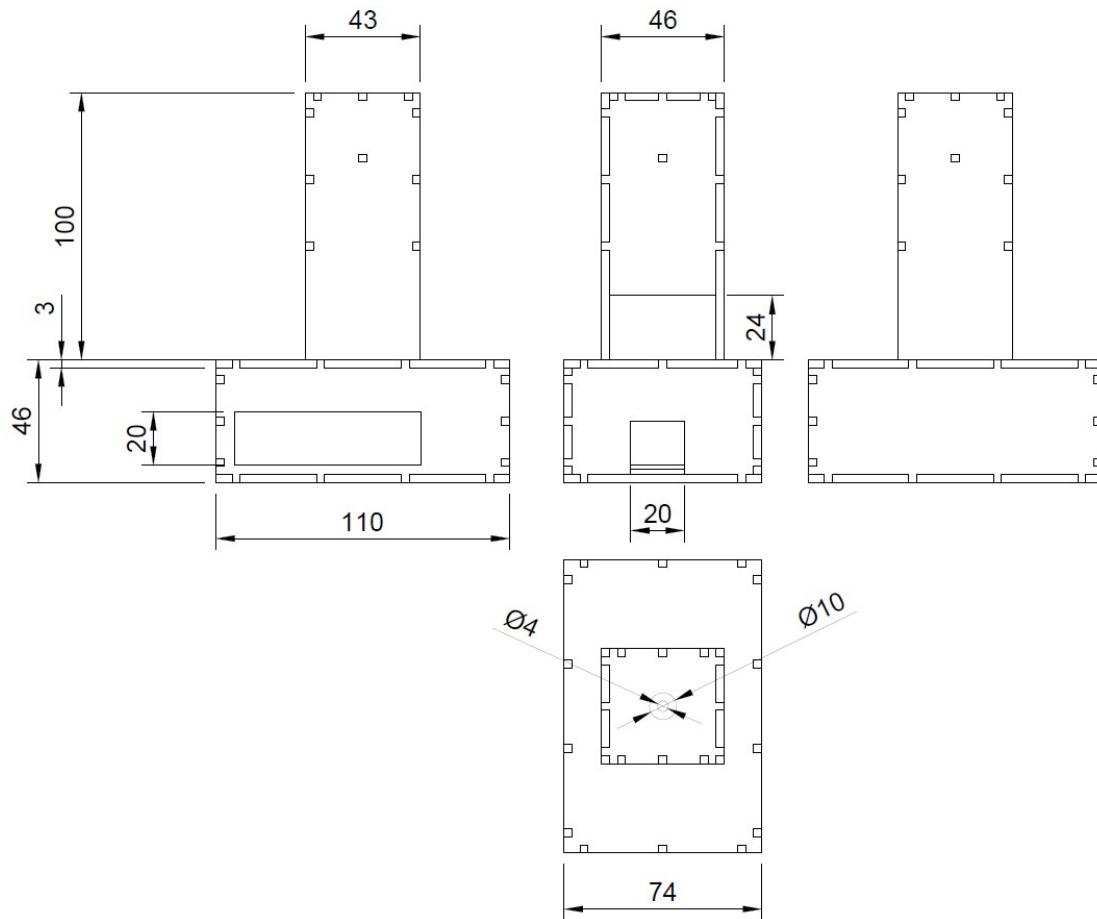


Figure 3.8: Mechanical drawing of the Raspberry Pi and LED housing. Dimensions shown are in mm.

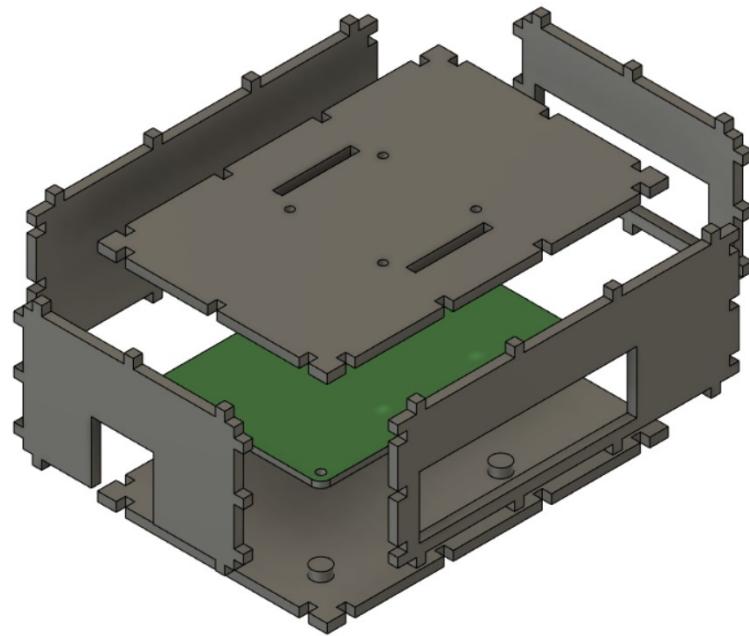


Figure 3.9: Exploded view of the Raspberry Pi housing unit. The unit includes 6 side walls and a dummy PCB is placed in the model for sizing and reference.

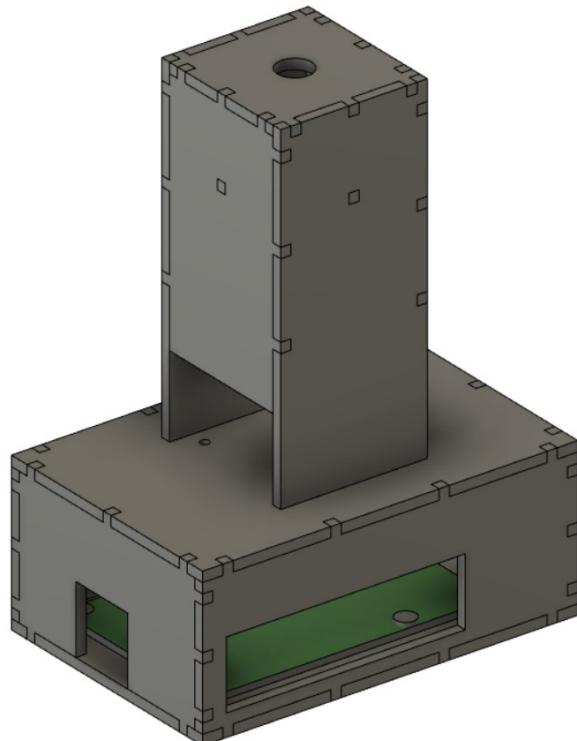


Figure 3.10: An overview of the final design of the Raspberry Pi and LED housing unit with all components connected together.

3.4 Summary of Development

The hardware developed in this project includes the LED housing, Raspberry Pi housing and slide mount. The Raspberry Pi housing serves as the base for the hardware of the system. The LED housing uses two extruded skegs to hold itself firmly in place while also allowing easy access to the sample chamber if necessary. The slide mount screws onto the Raspberry Pi housing using 32mm cheese head screws and nuts along with the CSI to USB adapter. The hardware components are tested with the software and bacterial samples in the experiment and the test's outcomes are outlined in the discussion chapter.

Chapter 4

Software Development

4.1 Introduction

This chapter outlines the methodology, design and development of the software component for the project. The software involves using a three-step process to present the microscopic image to a user. An image is first captured from the Raspberry High-Quality Camera, undergoes image reconstruction as well as colony counting and finally, is served to the user via an interactive web interface.

4.2 Methodology

4.2.1 Initial Development and Testing

Initial development of the software required some research and testing of the Raspberry Pi High-Quality Camera. The camera has a Sony IMX477 CMOS image sensor and interfaces with the Raspberry Pi 4 Model B via a MIPI CSI connector. Some initial packages must be installed to interface with the camera. This can be done by running the following bash command:

```
$ git clone https://git.libcamera.org/libcamera/libcamera.git  
$ cd libcamera  
$ meson build  
$ ninja -C build install
```

Listing 4.1: Bash commands to install the libcamera application

Once interfaced with the Raspberry Pi, bash terminal commands can be used to capture still images from the camera:

```
$ libcamera-jpeg -o test.jpg
```

Listing 4.2: Bash commands to capture an image with the libcamera application

This command captures an image and displays it. This helps provide an initial view of the camera's default configuration.

4.2.2 Using the Raspberry Pi Camera as a webcam

The process of interfacing with the camera within Python or C++ programs is made simpler by using a CSI to USB Adapter. This ensures that the camera can interface more seamlessly and allows for a higher level of control over the device, ultimately simplifying the software to avoid unnecessary processing. With the adapter in use, the `fswebcam` command-line interface (CLI) application can be set up and called to capture images and once again test the setup:

```
$ sudo apt-get update
$ sudo apt-get install fswebcam
$ fswebcam --device /dev/video0 -r 4032x3040 --png 0 output.png
```

Listing 4.3: Bash commands to install fswebcam and capture a still image

The commands above will update the system's package lists, install the `fswebcam` CLI application and finally capture an image from the USB device connected to the video0 bus at a resolution of 4032 pixels wide by 3040 pixels high.

4.2.3 Developing the image reconstruction and colony counting software

The software will be developed in Python to ensure that it can be easily replicated and can be installed on most Linux-based systems. The initial setup involves installing the library for reconstructing called HoloPy (Lab 2021a) by executing the following commands:

```
$ clone https://github.com/manoharan-lab/holopy
$ cd holopy
$ python setup.py develop
```

Listing 4.4: Bash commands to install the holopy Python library

For these commands to work, some additional packages may need to be installed. This is dependent on your system and you will have to follow the prompts to install the correct ones using the package manager. Once the holopy library is installed, the crucial OpenCV library must also be installed for Python via the following command:

```
$ sudo pip install opencv-contrib-python
```

Listing 4.5: Bash command to install OpenCV Python libraries

These packages may run successfully after the first attempt. Although, depending on your system version and installation type, additional packages may need to be installed. These packages will be identified by the Linux package manager and can easily be installed by following the prompts.

4.2.4 Developing the web monitoring interface

The web monitoring interface will be served to the user via a Flask web server. Flask is a python based web server and it is a powerful tool since it can interact with any python library or code. The required packages to install and run the microscopy software can be obtained from the project's GitHub page. This can be done by executing the following bash commands:

```
$ clone https://github.com/Cameron-Ray/lensless-microscopy.git  
$ cd lensless-microscopy/microscope-webview  
$ sudo pip install -r requirements.txt
```

Listing 4.6: Bash command to install OpenCV Python libraries

To ensure that the camera can function once the web server is ready to be deployed, the Debug flag in the Flask application must be set to False. The web server combines all aspects of the software into a singular running package as shown in the diagram below:

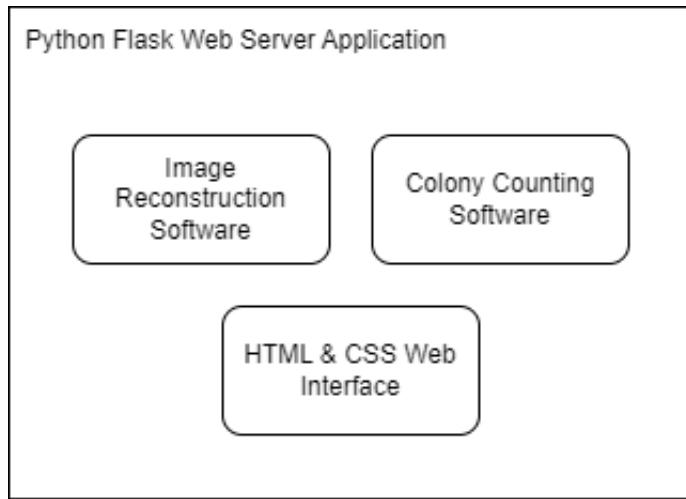


Figure 4.1: A diagram displaying the functionality encapsulated by the Flask web server.

4.3 Design

4.3.1 Image Reconstruction Software

The image reconstruction software uses the HoloPy library to reconstruct captured images into a higher resolution. This library features the following key functions:

1. `load_image()`

This function loads in an image using a path string which points to the location of the relevant image.

2. `bg_correct()`

This function uses the image data in question and an image of the background (before the sample is loaded) and performs a subtraction between the two images.

3. `center_of_mass()`

This function locates the centre of the beam of light.

4. `detector_grid()`

This function defines an output array which is populated during the propagation process.

5. `ps_propagate()`

This function performs a point source propagation of light back through the image to one or more distances from the image centre. The propagation distance is used to focus the object.

The background image is loaded first in either JPEG or PNG format. A background image may contain debris or other inconsistencies which need to be removed before propagation may occur. A background image may appear as the following:

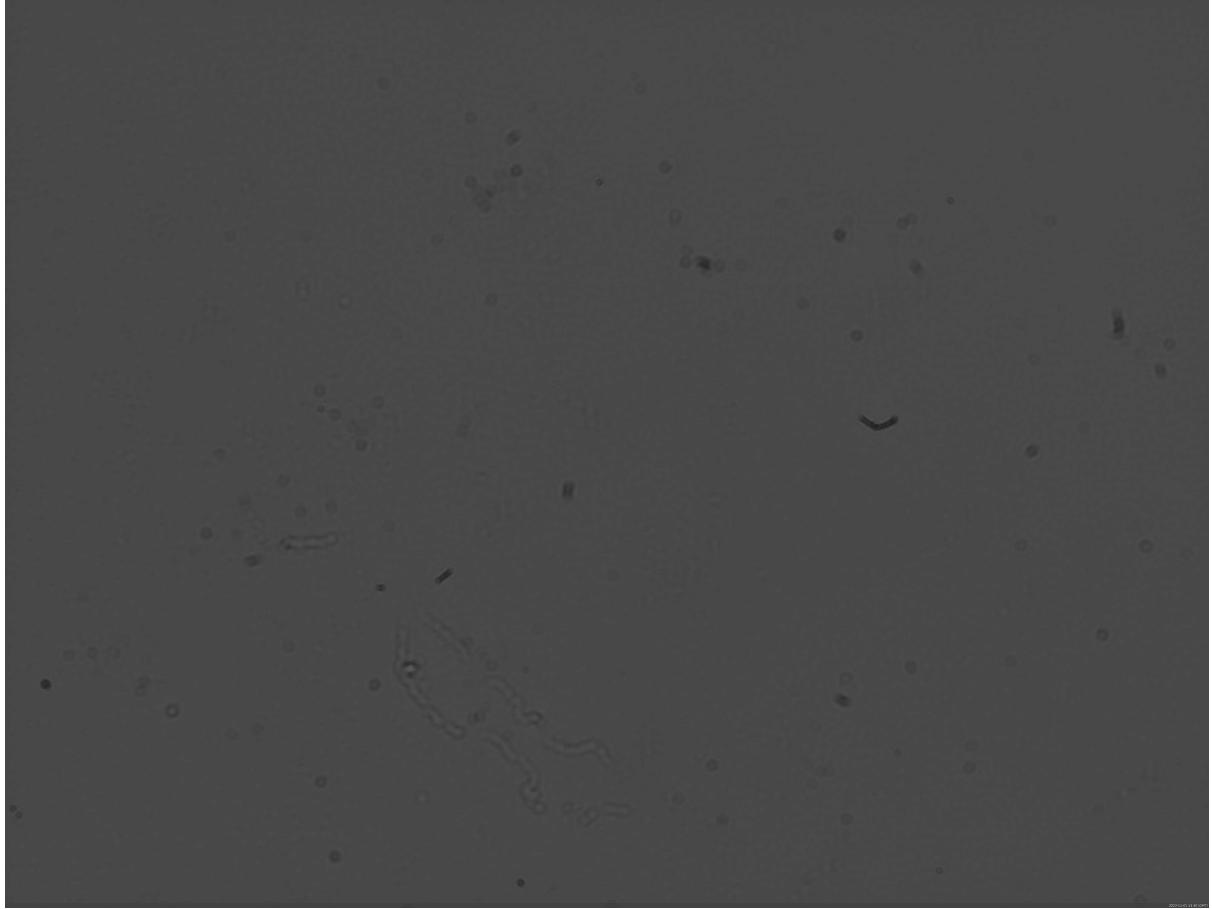


Figure 4.2: An example of a background image captured containing debris.

The point source propagation function is a wrapper for a plane wave propagation function and is limited to refractive indices of 1 which will be used for experimental purposes but may have to be refined later. The propagation uses the Kirchoff–Helmholtz Reconstruction Algorithm (Jericho and Kreuzer 2010) derived from the Kirchoff–Helmholtz transform:

$$K(r) = \int_S d^2\xi I(\xi) \times \exp(ik\xi \cdot \frac{r}{\xi}) \quad (4.1)$$

4.3.2 Colony Counting Software

The colony counting software segments the image data from the image reconstruction process and counts the number of bacterial colonies using edge detection. The software

is also written in Python and uses OpenCV 2 to process the image data. In the code the following key functions are used:

1. `imread()`

This function loads in an image using a path string which points to the location of the relevant image.

2. `inRange()`

This function uses the image data, a lower bound and an upper to create a mask which highlights the colonies.

3. `bitwise_and()`

This function masks the original image to capture the portions of interest highlighted by the mask.

4. `GaussianBlur()`

This function performs a Gaussian blur on the image to prepare it for edge detection. It smoothes the edges of the bacteria (or other objects) in the image.

5. `Canny()`

This function performs the crucial Canny edge detection to identify the boundaries of the colonies in the image.

6. `dilate()` and `erode()`

These functions are used together to close the space between the identified edges so that they become more pronounced.

7. `findContours()`

This function finds all the contours given by the Canny edge detection.

8. `contourArea()` and `convexHull()`

These functions are used to find a contour's area and then create a vector of points which outline the contour.

9. `drawContours()`

This function draws the contours onto the original image using the hull vectors.

10. `imwrite()`

This function saves the final output image with the count written on the image.

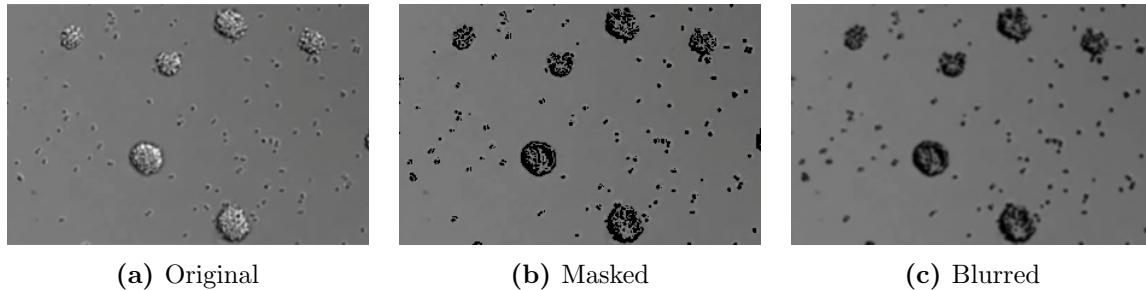


Figure 4.3: Colony Image Loading: (a) An example of an original colony image; (b) An example of the masked image; (c) An example of the Gaussian blurred image. Original image from Zaburdaev 2021

The beginning of the colony counting algorithm features the loading (4.3a) and masking phases (4.3b). The image is also highlighted (4.3c) based on colour boundaries so that edge detection can be performed with greater accuracy later.

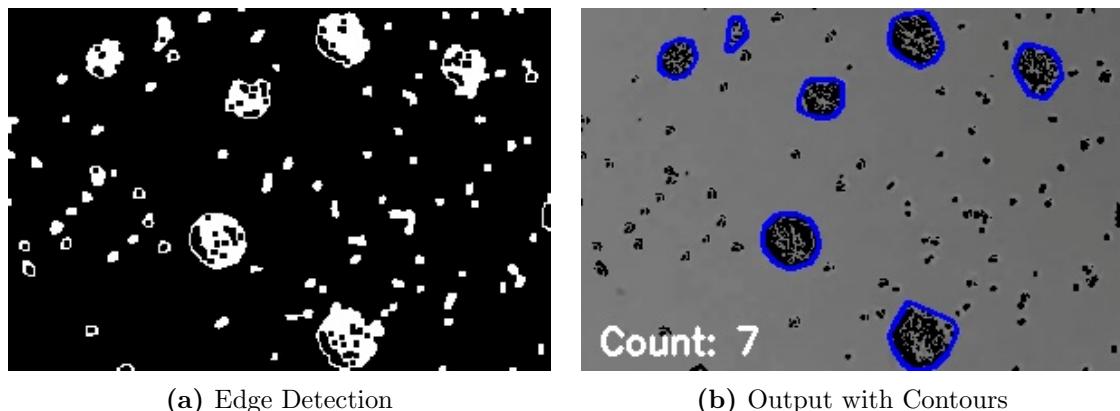


Figure 4.4: Colony Image Segmentation and Labelling: (a) Using the masked and blurred image, Canny edge detection is performed and can be seen here as a black and white output; (b) The original image is used to show the colony border overlays at the final output stage.

In the final phases of the colony counting process, the image is subjected to a Canny Edge Detection algorithm (4.4a) and then the colony boundaries are drawn on the original image (4.4b) to show which areas contain a valid colony.

4.3.3 Web Monitoring Interface

The web monitoring interface features a landing page with a live view of the microscope's camera. Additionally, the front preview shows live statistics on colony count and growth rate. The Raspberry Pi's diagnostics are also shown on this page briefly. The interface is displayed to a user on a local network as follows:

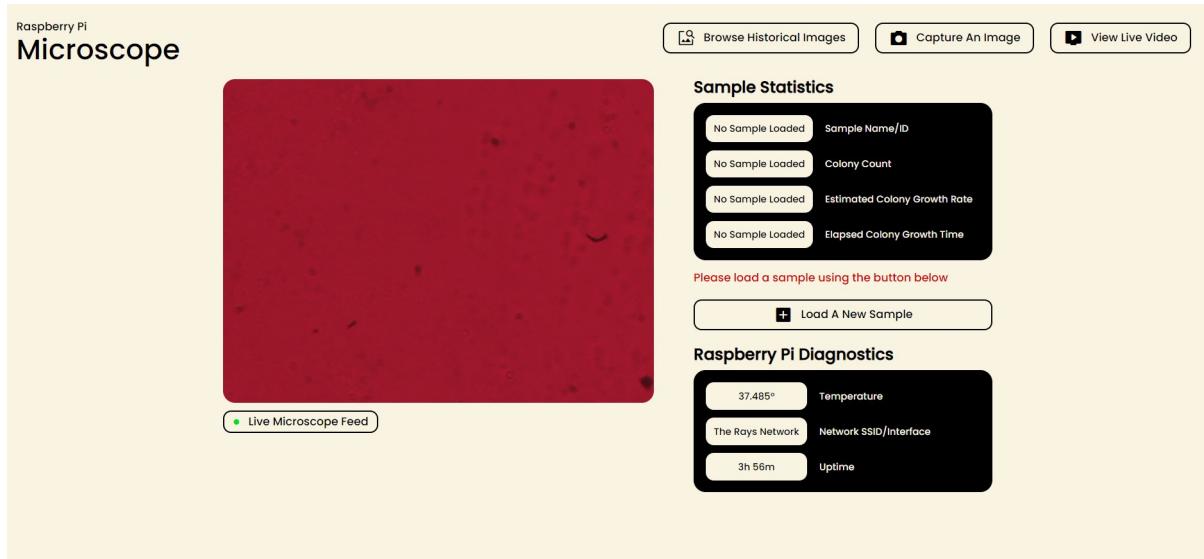


Figure 4.5: The landing page of the web interface for the microscope system.

The web server is hosted using a flask application and contains the following page routes:

1. index

This is the landing page of the web server. It displays the current colony status and Raspberry Pi Statistics to the user.

2. historical-images

This page route displays a list of historical images captured from the microscope and allows users to view and download the images to their local device.

3. image-capture

This route performs an image capture of the live microscope display. It does not show a page but rather redirects back to the index once it has completed the capture.

4. load-sample

This page route shows a form where the user may enter the details for a new sample that is to be loaded. This page also allows for background capture to be invoked so that the user can save the background before the sample is loaded.

5. background-capture

This route is similar to the image-capture route, it only captures a background image and redirects back to the load-sample route when complete.

6. video-feed

This route is different to the other routes in that it does not render an HTML template but rather returns a video stream Response object which is used to populate an image tag in the HTML template.

Loading a new sample is completed via the "Load New Sample" option on the index page. This page is designed as follows:

The screenshot shows a web interface for a Raspberry Pi Microscope. At the top left, it says "Raspberry Pi" and "Microscope". Below that, there are four main sections: "Step 1: Sample Information", "Step 2: Capture Background Image & Initialise Sample", "Step 3: Review Changes", and "Step 4: Track Live Changes".

- Step 1: Sample Information**: A black rectangular input field containing three white input boxes. The first box is labeled "Sample Name/ID", the second "Colony Count (Optional)", and the third "Colony Growth Data (Optional)".
- Step 2: Capture Background Image & Initialise Sample**: A section with a single button labeled "Capture & Initialise".
- Step 3: Review Changes**: A section containing two radio buttons. The first is labeled "Sample Name/ID" and the second is labeled "Colony Count".
- Step 4: Track Live Changes**: A section with a button labeled "Back to Live View" featuring a video camera icon.

Figure 4.6: The sample load page of the web interface for the microscope system.

A sample is represented by the **Sample** class developed specifically for a bacterial sample. It contains globally accessible fields to store the sample's related information. The following Class Diagram outlines the class and its methods:

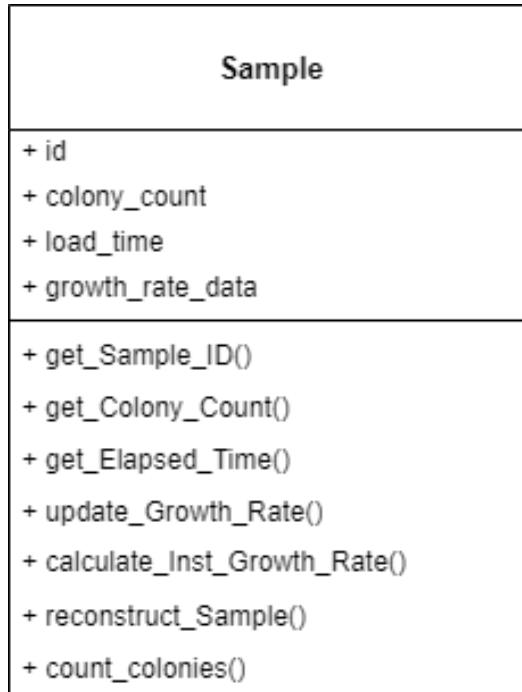


Figure 4.7: A bacterial sample's associated Sample Class Diagram.

The **Sample** class can be interacted with via its methods to obtain current data pertaining to the sample's behaviour or to update its fields using methods such as `count_Colonies()` or `reconstruct_Sample()`. These methods will run the functionality discussed in the image reconstruction (4.3.1) and colony counting software development (4.3.2) sections. Once a clear image has been reconstructed from a snapshot of the live sample view, it is processed by the colony counting software to update the field in the bacterial sample object. This object is discarded and replaced when a new sample is loaded.

4.4 Summary of Development

The software developed in this project included image reconstruction, colony counting and a web server interface. The web server ties all aspects of the software together into a neater package so that users do not have to run the three aspects separately. The web server is very flexible and has the ability to include many more functions in the future. Its purpose was to allow remote interfacing with the microscope and to provide a user-friendly experience while interacting with complicated subsystems. The image reconstruction and colony counting functionality are tested alongside the web server in the experiment and the test's outcomes are outlined in the discussion chapter.

Chapter 5

Experimental Design and Setup

5.1 Introduction

This chapter outlines the experimental methodology and setup for the project. The experimental setup brings all the relevant subsystems together and defines a set of experimental objectives. Additionally, a collection of experimental steps is outlined and will be carried out in the lab.

5.2 Design

5.2.1 Experimental Setup

The experimental setup involves bringing the software and hardware together to create a working microscopy device. The Raspberry Pi was placed on the base plate and secured with screws and standoffs. Once the Pi was in place, the side walls were secured in the clip slots. Before the top wall was put in place, the CPU fan was installed. Then the camera adapter, camera and slide mount were coupled together and secured to the housing's top wall. Once the last components were secured onto the top wall of the housing, the LED circuit was connected to GPIO Pin 26 and then the top wall was clipped in place. In the figure below, the exposed slide mount is visible:

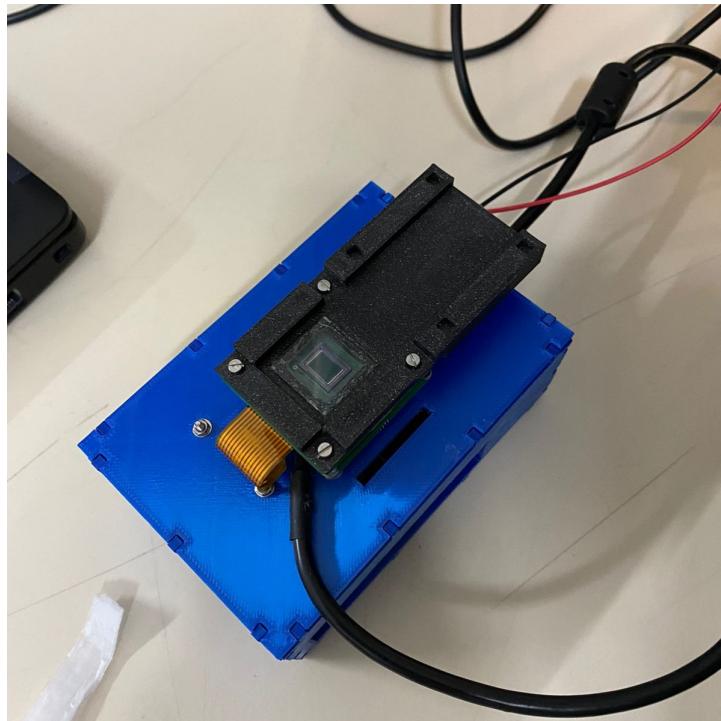


Figure 5.1: The slide mount is exposed by removing the LED housing. Removing the LED housing is necessary to clean and access the slide mount.

The final step was to set up the LED housing. This required that the LED be placed into its pinhole inside the LED housing shelf. Lastly, The LED housing is secured in the long clip slots present on the top wall of the Pi housing. The following images show the final configuration from four different angles:



Figure 5.2: Side view of the experimental setup showing a side wall.



Figure 5.3: Front view of the experimental setup showing the side wall where USB inputs and an Ethernet port are found.

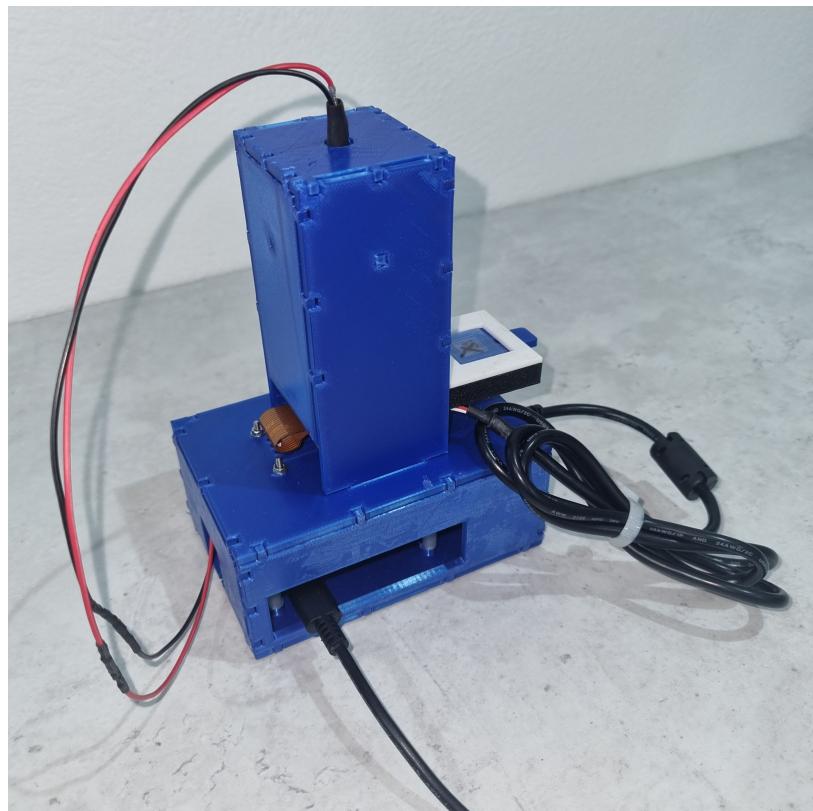


Figure 5.4: Side view of the experimental setup showing the side wall where power and video output are found.

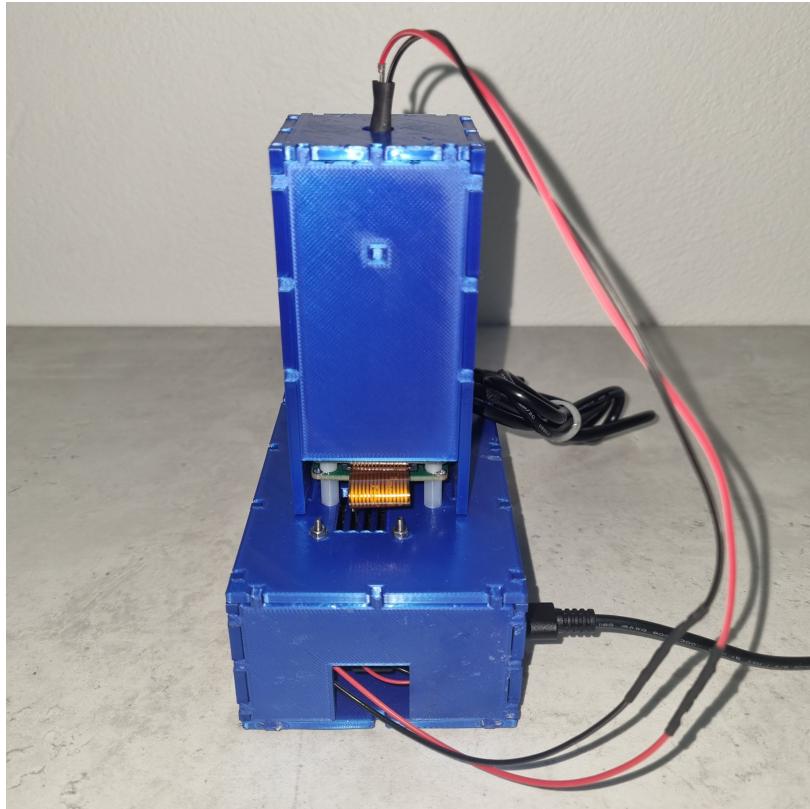


Figure 5.5: Back view of the experimental setup showing the side wall where the Raspberry Pi’s SD card can be removed from and where the LED circuit wiring exits the housing.

The Raspberry Pi is then connected to power via a USB-C cable. It can either be powered from a 5V DC power supply or a portable power bank with a 5V rating. Before the experiment begins, the distance between the LED light source and the sample plane is measured and recorded as $z_1 = 58mm$ and an approximate recording for the distance between the sample plane and the image sensor is taken as $z_2 = 0.75mm$. These distances are used in the initial condition setup for the image recognition software if the system was tested using the lensless microscope. The distances that were recorded can be observed in the figure below:

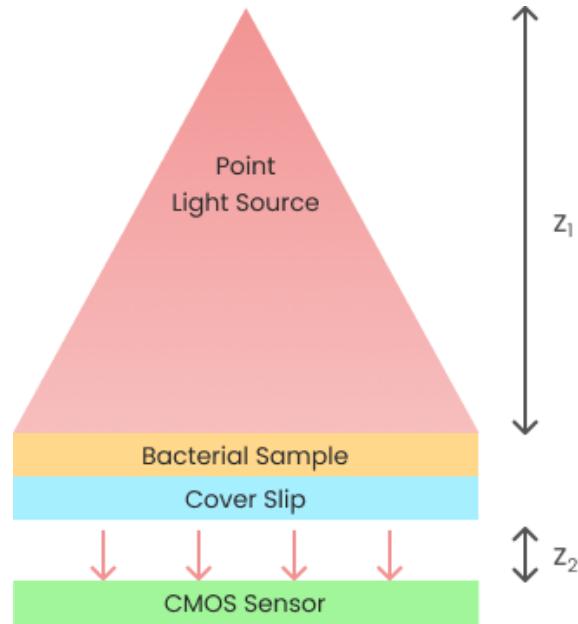


Figure 5.6: Diagram showing the distance measurement used to configure the reconstruction algorithm.

5.2.2 Experiment Objectives

This experiment aims to test the subsystems of both the hardware and software components. Additionally, sample preparation and recording procedures are tested to discover if they are appropriate for this system and if they produce viable images.

The hardware objectives to test include:

1. Does the housing support regular use or do the components deteriorate?
2. Is the system easy to assemble?
3. Does the camera interface well over USB compared to CSI?

The software objectives to test include:

1. Does the web server provide a clear live image of the sample?
2. Does the image reconstruction work on image data supplied from both example images and recorded images?
3. Does the colony counting software accurately detect colonies and provide an accurate count?

And lastly, preparation and detection techniques to test include:

1. Is the gene frame suitable as a sample viewing plane?
2. Do colonies grow under room temperature or incubator environments better?
3. Are colonies easily visible in the resulting images?

5.3 Methodology

5.3.1 Preliminary Tests

Testing the software and hardware involves setting up the Pi and the housing unit in the lab. Once the Pi is set up and connected to a WiFi network, background images will be recorded using the command line to verify the camera setup.

Initially, samples will be prepared using an exposed coverslip and a gene frame to determine which will work best. Next images will be recorded of the liquid and the agar samples. These will provide a baseline sample for the remaining tests.

5.3.2 Final Tests

During experimentation it is important to clean the equipment, this can be seen in the figure below:

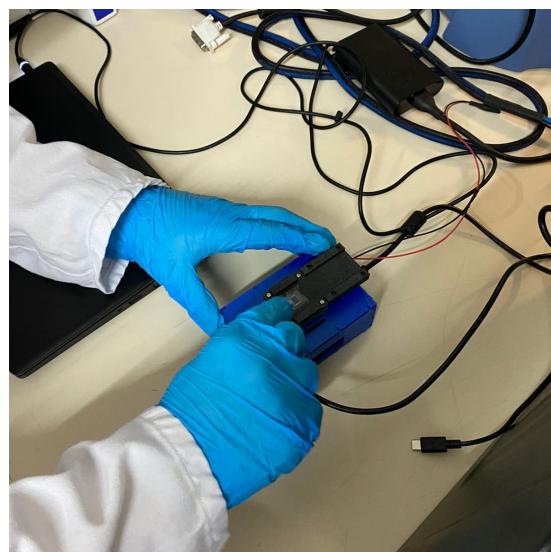


Figure 5.7: Cleaning the protective cover slip before and after the sample is viewed.

Ethanol will be used to clean the cover slip before and after the experimentation. For the final testing procedure, there will be two time periods within which sample images will be recorded periodically. The first test will cover an 18-hour window and the second test will be a 24-hour window. To compare the result of the lensless microscope, three samples will be prepared and stored in an incubator. These samples will then be viewed under a normal microscope after 5 days.

5.3.3 Test Target Capture For Calibration

A USAF test target 5.8 will be used to test and find the resolution of the lensless microscope. The relevant resolution formulae will be used in the results chapter.

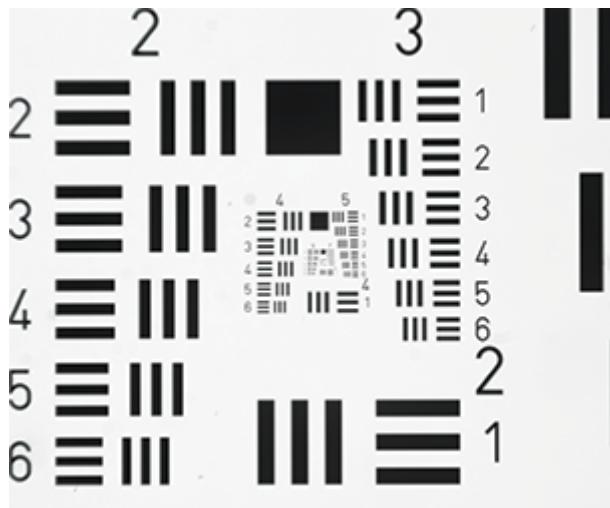


Figure 5.8: The USAF test target to be used to find the imaging resolution (Optics 2022b)

5.4 Summary of Experimentation

Experimentation will encompass hardware, software and techniques. This will provide insight into the superior and inferior aspects of the system. The tests will be carried out at the Institute for Infectious Disease and Molecular Medicine. Their lab has a powerful microscope, an incubator and other lab equipment necessary to perform the tests.

Chapter 6

Results

6.1 Introduction

The results recorded for the experiment involved both testing the software on sample data as well as capturing new data in a lab environment. This chapter will provide all the results captured from all the tests that were performed and highlight observations in the images captured. Higher resolution results data can be obtained from the links in the A.3

6.2 Subsystem Testing

6.2.1 Raspberry Pi High-Quality Camera Testing

Initial testing was completed using the Raspberry Pi Camera. This allowed for simultaneous testing of the camera and sample plating techniques. The first image captured was of the microscope's background (6.1), it shows debris and inconsistencies which will be subtracted during the reconstruction process.



Figure 6.1: A background image captured during initial testing of the camera.

Secondly, a liquid culture was prepared (6.2) and placed into the sample chamber. This culture provided a second opportunity to test the camera. It produced a clear image and demonstrated the wide FOV of the camera.

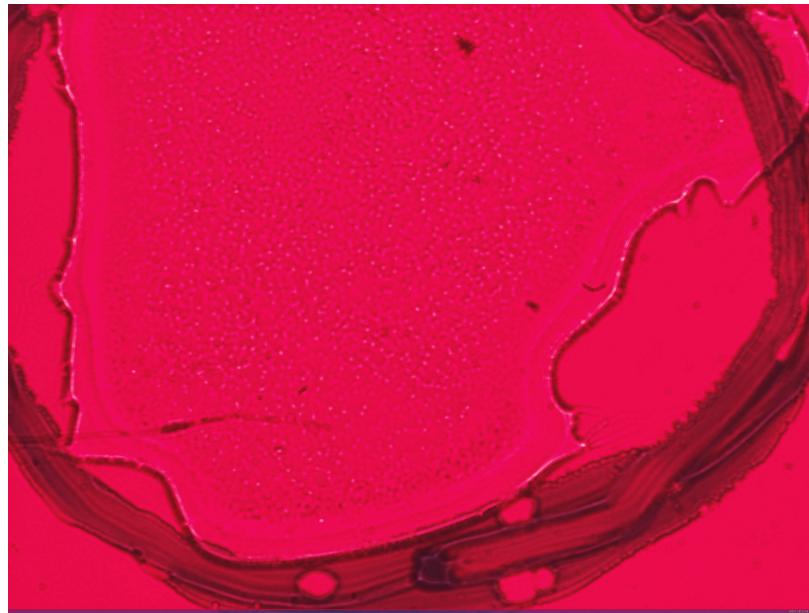


Figure 6.2: An initial test of the camera using liquid culture.

Lastly, a sample was prepared using agar (6.3) and placed inside a gene frame. This resulted in a smoother sample with less debris and fewer air bubbles. This technique involved using two coverslips and a gene frame and pressing them together to form an airtight sample.



Figure 6.3: An initial test of the camera using a culture prepared on agar.

6.2.2 Image Reconstruction Testing

Since images of colonies on the lensless microscope could not be captured in a short window, the image reconstruction subsystem was tested using sample image data provided by the HoloPy library. The output images below (6.4) show the same sample but with light propagated back through them to increasing distances from the image sensor from (a) to (f).

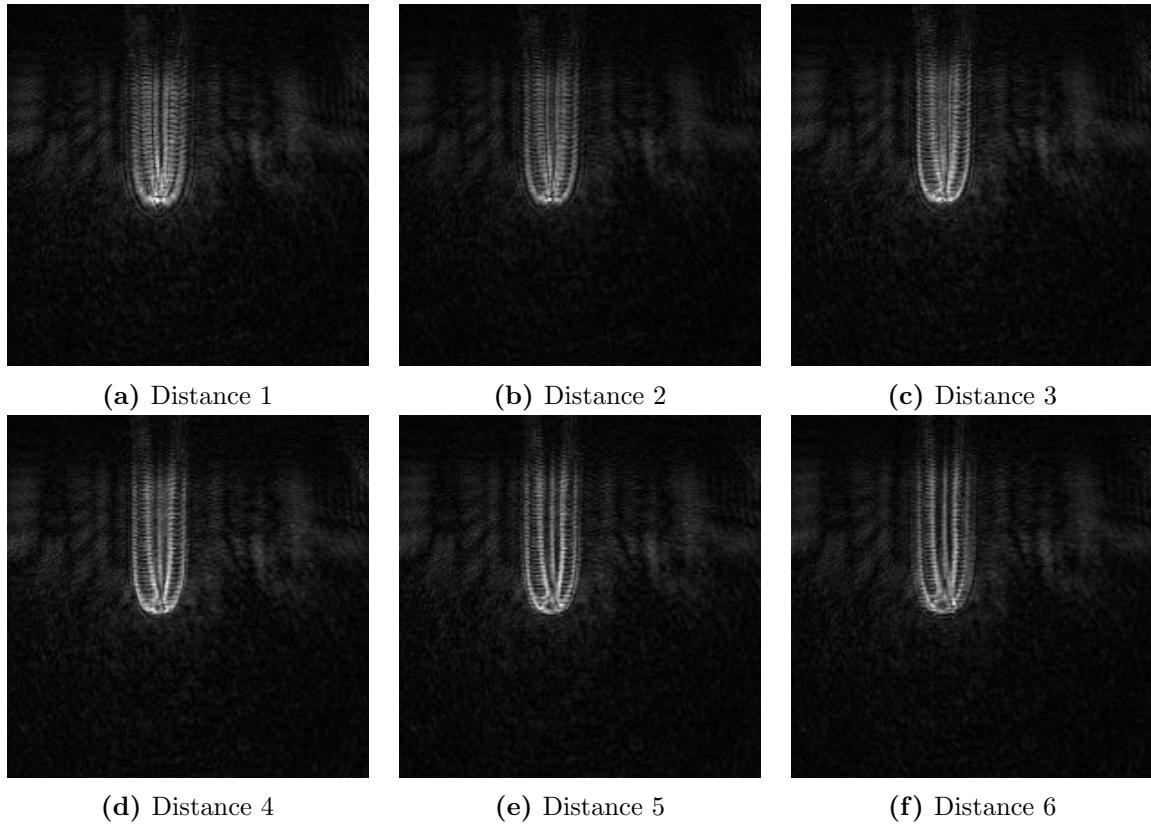


Figure 6.4: A series of images of sample data provided by the HoloPy (Lab 2021a) library. Each image has been reconstructed by propagating a light wave back through them but at varying distances from the image sensor plane.

6.2.3 Colony Counting Testing

Colony counting was tested as its own subsystem, similar to the image reconstruction subsystem. This was once again due to the lack of colonies present in the captured image data. Two sample images were subjected to the colony counting algorithm.

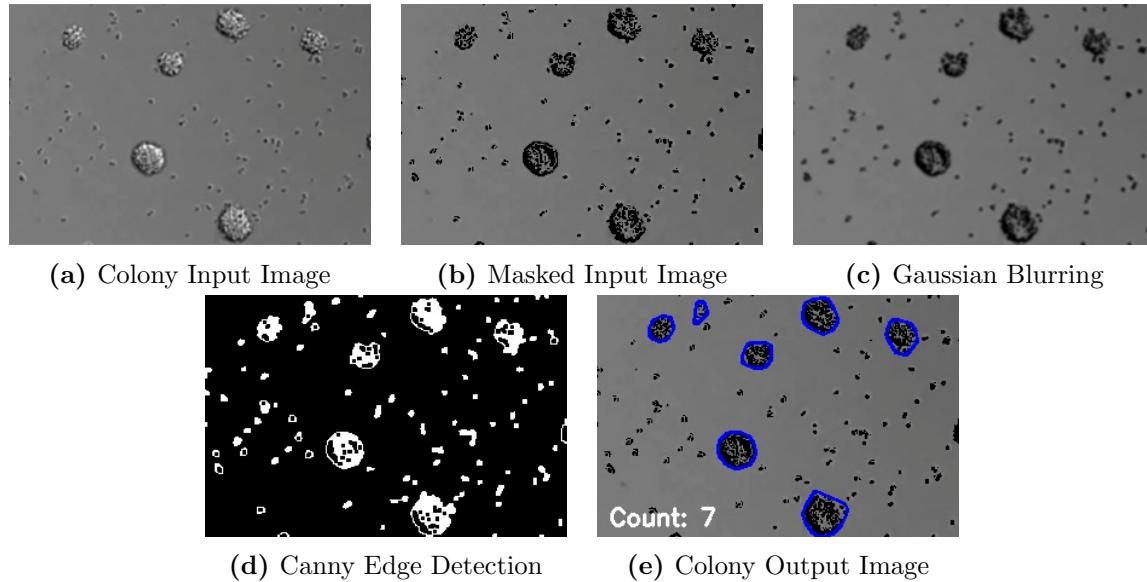


Figure 6.5: The colony counting algorithm is outlined in order from figure (a) to figure (e). Original image from Zaburdaev 2021

The first step in the algorithm (6.5a) is the image input stage. The image seen in this figure is loaded into the Python software. The second step (6.5b) masks the input image boosting the image highlights. The third step (6.5c) blurs the image to reduce the prominence of smaller flecks visible in the sample. The fourth step (6.5d) utilises Canny Edge Detection to find the colony contours. These contours are drawn on the input image and saved at the last step (6.5e).

The first test image had medium-sized colonies with large spaces between them. This image was also not converted to grayscale as its colour range was not too varying. The output image highlights the colonies with a blue outline and counts the number of colony boundaries detected. Finally, a colony count is placed on the image for reference.

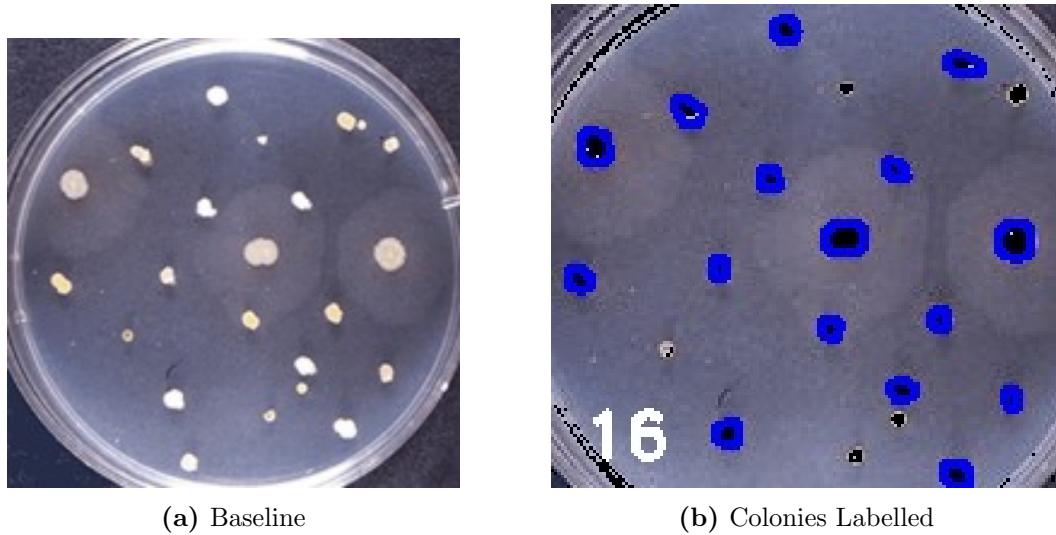


Figure 6.6: First Colony Count Test: (a) This baseline image is provided to the software for its colonies to be counted; (b) The software returns an image with the colonies highlighted.

The second test image had colonies with less spacing and had to be converted to grayscale first before processing. Additionally, some edges were cropped to avoid false detections. The colony boundaries were once again computed and highlighted with a green border in this test.

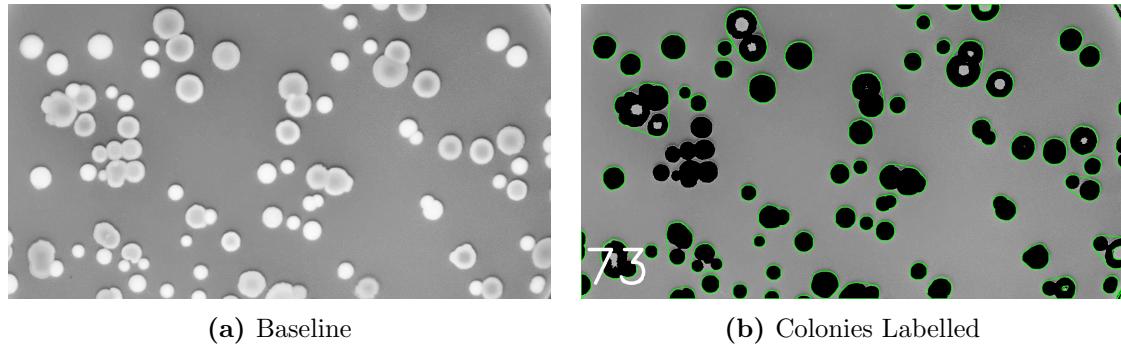


Figure 6.7: Second Colony Count Test: (a) This baseline image is provided to the software for its colonies to be counted; (b) The software returns an image with the colonies highlighted.

6.2.4 Web Server Live View Testing

The web server was hosted on the Raspberry Pi and accessed via the Raspberry Pi's IP address on the local network. The figure below shows a screen capture of the live viewing pane:

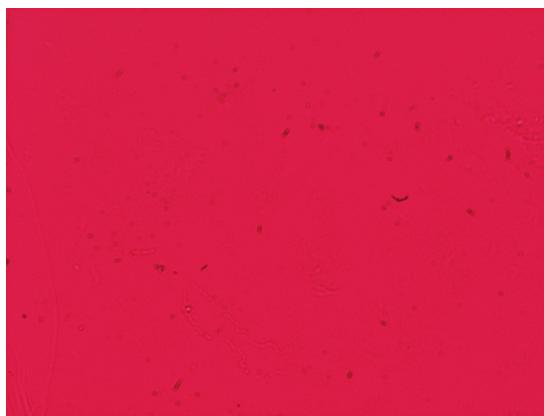


Figure 6.8: Live viewing pane on the web server's index page.

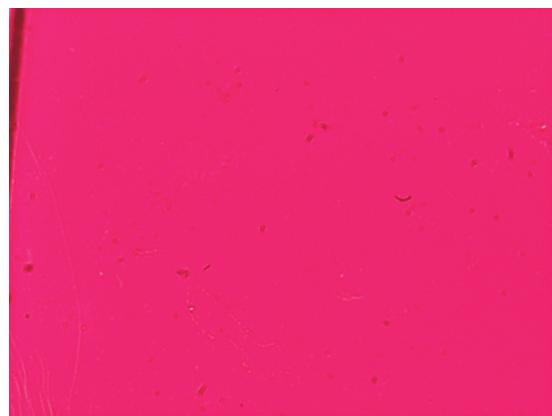
6.3 Lab Testing

6.3.1 18-Hour Growth Test

These images (6.9) were taken in the first long-run test. Image 6.9a is the first recorded image and 6.9b is the last recorded image. The images show little to no growth at room temperature here.



(a) Initial Sample Capture



(b) Final Sample Capture

Figure 6.9: First Growth Test: (a) An image of the initial state of the bacterial sample is captured; (b) The final state of the sample is captured after 18 hours using the same camera parameters as the first image capture.

6.3.2 24-Hour Growth Test

These images (6.10) were taken in the second long-run test. Image 6.10a is the first recorded image and 6.10b is the last recorded image. The images show little to no growth at room temperature here once again.

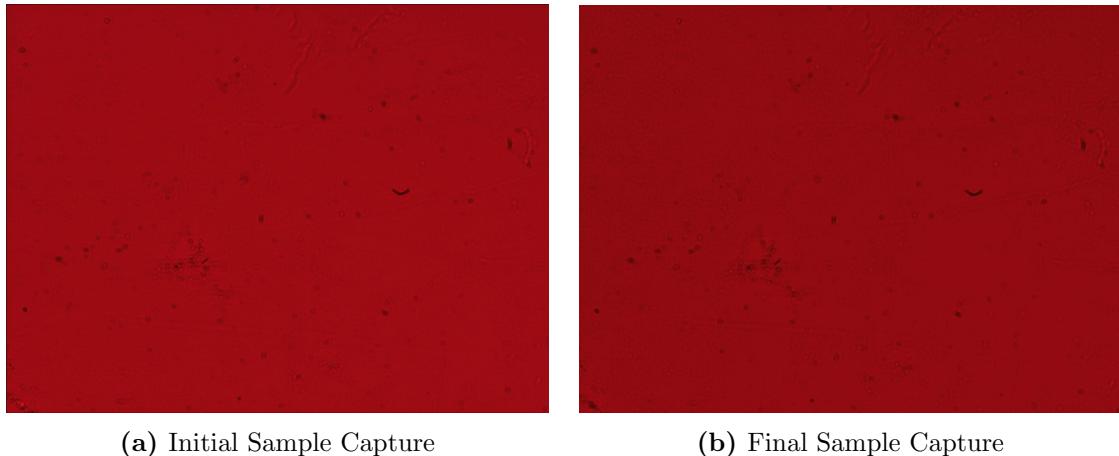


Figure 6.10: Second Growth Test: (a) An image of the initial state of the bacterial sample is captured; (b) The final state of the sample is captured after 24 hours using the same camera parameters as the first image capture.

6.3.3 5-Day Growth Test Using An Incubator

After little success with the shorter incubation periods, a longer trial was run. The samples were prepared and placed in an incubator for 5 days and then placed under a normal microscope. Two images were captured of each sample. These images were captured at 40x and 100x magnification and show clearly the colony growth.

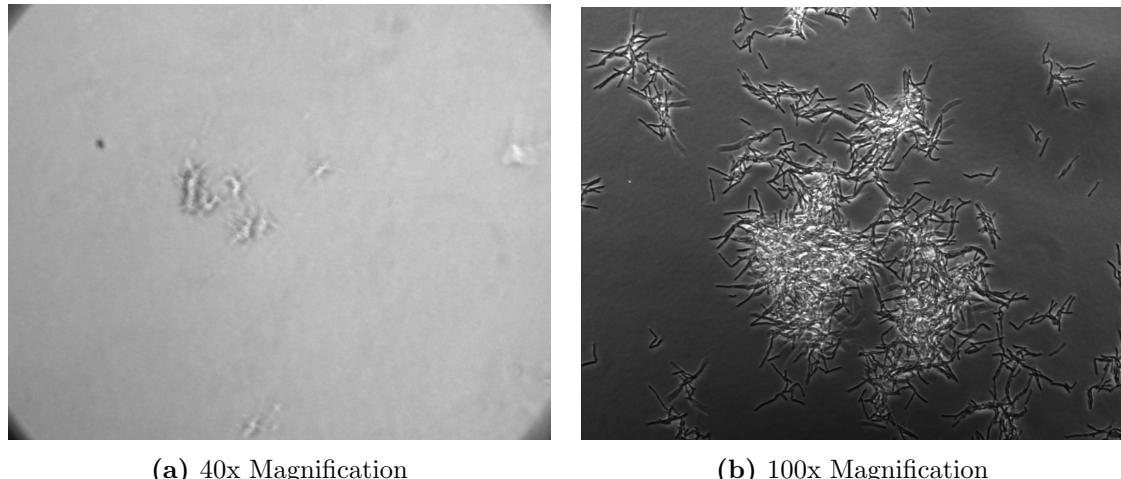


Figure 6.11: 5-Day Growth Test: (a) Colonies of M. tb visible under 40x magnification; (b) At 100x magnification, the colony is confirmed by the rod shape of the bacterium.

Image 6.11a shows the 40x magnification image that was captured. In its centre, a colony has grown over the 5-day period.

Image 6.11b shows the 100x magnification image that was captured. In its centre, a closer look at the colony is visible. The rod-shaped bacteria are visible in great detail.

6.3.4 Test Target Capture

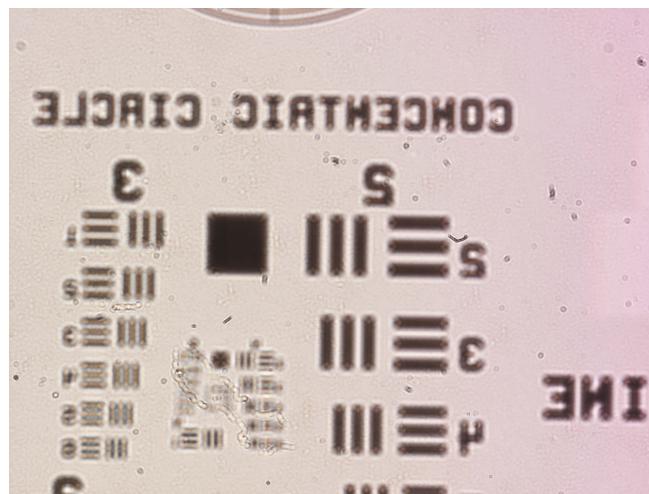


Figure 6.12: USAF Test Target under white light for better visibility.

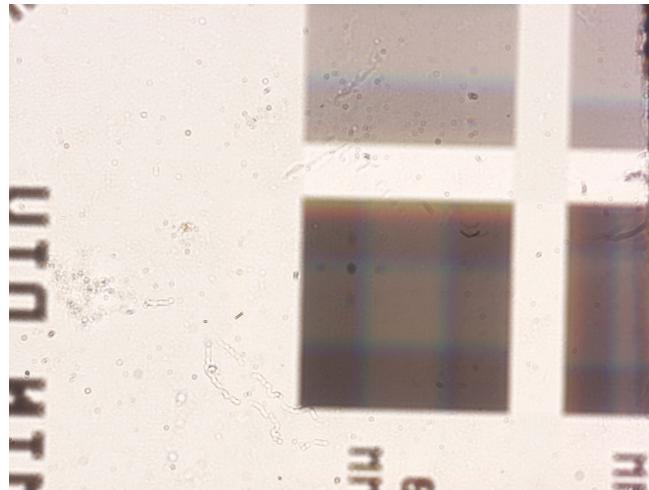


Figure 6.13: Additional test target capture to show the accuracy and resolution of images.

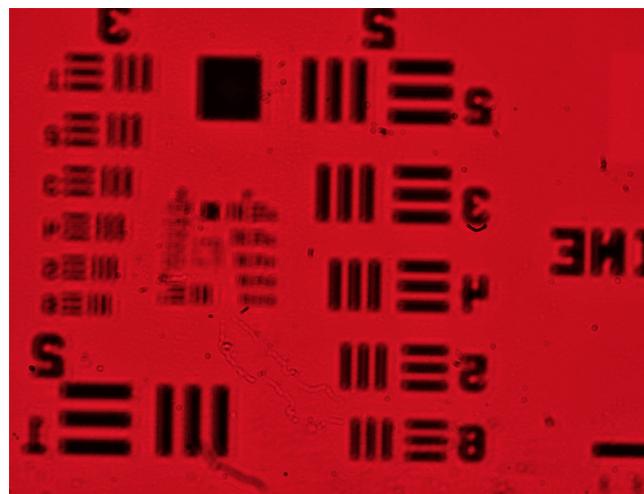


Figure 6.14: USAF Test Target under red LED light as would be seen under normal operation for the lensless microscope.

6.4 Summary of Results

This section presents an overview of the data that was captured during experimentation and comments briefly on the behaviour observed in the subsystem tests and lab tests. It is clear that all subsystems operate as expected. Lab testing will need to be discussed in detail as unhelpful outputs came from short testing durations.

Chapter 7

Discussion

7.1 Hardware

3D printing was a suitable choice of fabrication method for all the housing components. The PLA used to fabricate the components melts at 215°. This means that the components can withstand suitably high temperatures before failure. The components are designed with an easy clip-in system. This greatly improved fabrication time and reduced the number of failed builds. A higher number of iterations of each component was achieved which meant that parts could be rapidly changed if necessary.

The CSI to USB adapter from Arducam (Arducam 2020) integrated seamlessly with the hardware. Initially, this part was not needed until it was realised that USB webcam interfacing in Python is much easier. It was able to slot between the Raspberry Pi Camera PCB and the top wall of the Raspberry Pi housing. The only adjustment needed was to insert some standoffs to increase the height above the top wall. This was due to a connector which protrudes from the bottom of the adapter's PCB. Overall, it was very effective

In reflection on the hardware experiment objectives: The housing is robust and supports regular use while still being cost-effective so that components may be easily replaced, the system is extremely easy to assemble with components that clip together, and the USB camera interfaces in a much better way than the CSI camera. The longer USB cable is the only drawback to this setup.

7.2 Software

The software subsystems include image reconstruction, colony counting and web server programs. Each functioned as expected when unit testing was performed. The web server combined the image reconstruction and colony counting software into its Sample class successfully. This integration allows for continuous image capture, image processing and quantification of colony count and growth.

The image reconstruction subsystem was limited by many factors and could only produce images in specialised cases. The HoloPy Python library that was used, does not have a wide enough support base to have new features released often. This means that there are features which are yet to be improved. The first limitation observed was the refractive index limit of 1. This refractive index was suitable for the test image data supplied by HoloPy but it meant that the lensless microscope could not reproduce images with the same accuracy as with the test images. Secondly, there is a memory limitation when processing the image. Lab 2021b specifies that there is a convolution method outlined in Jericho and Kreuzer 2010 which could solve this memory limitation. Finally, the image processing pipeline can only operate with grayscale images. This means that there is either extra overhead to convert images to grayscale before processing or colour images cannot be displayed at the output if desired.

The image reconstruction tests shown in Figure 6.4 are of the same sample with different light propagation heights. At each height, the image quality can be assessed. At distance 5.6.4e there is a high level of detail observed. This propagation height is deemed to provide the best for that type of sample and image format.

Colony counting proved to be remarkably successful. Even though this subsystem was tested on its own, in the same way as the reconstruction subsystem, it had a much better capability in handling varying input parameters. The colony count observed in 6.6b is extremely accurate. This is attributed to the fact that the colonies are spaced far apart and none of them is irregular in shape. The second count observed in 6.7b is less accurate in comparison to the first count. This is due to the input image quality and spacing of colonies on the medium.

In reflection on the software experiment objectives: The web server does produce an easily accessible and highly visible live viewing pane, the image reconstruction only works reliably on sample data due to limitations in memory size, and the colony counting software accurately counts the number of colonies in a given image.

7.3 Experiment

7.3.1 Bacterial Sample Growth

The growth of the samples in the first 6.3.1 and second 6.3.2 tests was not visible in the recorded images. This was most likely due to factors such as the sample preparation procedure, total growth time or lack of image processing. Since these images were recorded manually as part of the experimental process, they were not passed through the image reconstruction software. This potentially allowed higher-resolution images to be missed.

The 18-hour growth test saw little difference between its first 6.9a and last capture 6.9b. There were minor differences in background noise. This noise could have been caused by an external source such as ambient lighting or debris landing on the sample plane.

The 24-hour growth test used a better-quality sample which resulted in a clearer view of the agar at the initial stage 6.10a. Although there are a few areas in the final sample capture 6.10b which could be possible areas for growth, they were deemed to be inconsistent with the M. tb rod shape.

The 5-day growth test with images captured on a powerful Zeiss microscope confirmed that the sample preparation was most suitable for this application. The resulting colony growth points to a lack of growth time in the first and second tests. The 40x magnification capture 6.11a shows a clear, high-level view of the sample and the colonies grown on it. The 100x magnification 6.11b validates the sample's preparation technique again as it shows clear growth of the bacterium. The rod-shaped M. tb is highly-visible in the image.

7.3.2 Resolution Analysis and Calculation

To calculate the resolution of the camera, the formulae outlined in Optics 2022a are used:

The parameters of the Raspberry Pi Camera are used to determine the resolution:

Table 7.1: Raspberry Pi High-Quality Camera Parameters

Parameter	Value
Pixel Size	$1.55\mu m \times 1.55\mu m$
Number of Pixels	4056 x 3040
Desired FOV (Horizontal)	200mm

First, it is necessary to calculate the limiting sensor resolution 7.1:

$$\begin{aligned}\xi_{ImageSpace} &= \left(\frac{1}{2 \times s}\right) \times \left(\frac{1000\mu m}{1mm}\right) \\ \xi_{ImageSpace} &= \left(\frac{1}{2 \times 1.55\mu m}\right) \times \left(\frac{1000\mu m}{1mm}\right) \\ &\approx 323 \frac{lp}{mm}\end{aligned}\tag{7.1}$$

Second, the sensor dimensions are found 7.2:

$$\begin{aligned}H_{Horizontal} &= 1.55\mu m \times 4056 \times \left(\frac{1mm}{1000\mu m}\right) = 6.29mm \\ H_{Vertical} &= 1.55\mu m \times 3040 \times \left(\frac{1mm}{1000\mu m}\right) = 4.71mm\end{aligned}\tag{7.2}$$

Third, the desired magnification is calculated 7.3:

$$m = \frac{6.29mm}{200mm} = 0.03145X\tag{7.3}$$

Lastly, the resolution of the system is calculated 7.4:

$$\xi_{ObjectSpace}[\mu m] = 323 \frac{lp}{mm} \times 0.03145 = 10.16 \frac{lp}{mm} \approx 9.84\mu m\tag{7.4}$$

It can be seen that this amounts to a resolution that is bigger than the pixel size of the image sensor. The object resolution indicates the minimum resolvable width viewable on a captured image. This minimum resolvable width can be compared to the test target recorded in the results chapter in 6.12. The test target is a USAF Resolving Power Test Target 1951. In the figure below 7.1, it is observed that the group and element pair which is most visible is 3,3. This corresponds to a resolution of $10.08 \frac{lp}{mm}$. This agrees with the calculations above.

7.3.3 Objective Analysis

In reflection on the lab experiment objectives: The gene frame sample preparation method is appropriate for the lensless microscope, the colonies grew well in the incubator at 37°C which is M. tb's incubation temperature, and the colonies are easily visible at the output of the colony counting software.

The test targets provided much-needed insight into the resolution of the system. The calculations completed in 7.3.2 aligned with the minimum visible USAF test target.



Figure 7.1: Minimum visible line pair group shown. Group 3,3 is the last set of line pairs visible.

Chapter 8

Conclusions

The research objectives of the project were achieved through a thorough literature review and the suggested approach to solve the problem statement. The suggested approach drew on key elements from the literature to outline a suitable hardware, software and experimentation solution within the specified scope.

The hardware design objectives were all met as the 3D printed housing was robust and performed well during experimentation. The 3D-printed components were printed multiple times with each iteration improving on the previous. The software design objectives were mostly met. Development of a robust image reconstruction program was difficult as there were found to be restrictions on libraries used. Overall, the software subsystems worked in their individual testing. Some subsystems had to be fed sample data since experimentation did not yield the desired data.

The experimentation objectives were well met but yielded unexpected results. Micro-colonies were not visible over a period of 24 hours during the image capture process using the lensless microscope. This could be due to a number of factors including the environment, the sample preparation or the lack of magnification in the imaging process. The longer experimentation period and observation under the normal microscope yielded a successful image of bacterial colonies.

Overall this project has met its goals and a working lensless microscope system was delivered. The microscope worked well in a lab environment and its effectiveness means that it could be developed further with additional testing. The imaging ability of the camera was investigated and yielded a benchmark of its ability to capture images of microcolonies, however, this investigation may need to be carried out further to understand the camera's limits.

Chapter 9

Recommendations and Future Work

9.1 In-depth research into reconstruction methods

This project is scoped to develop both hardware and software aspects. This made it difficult to focus on one subsystem entirely. It would be beneficial to adapt the image reconstruction software to a wider variety of bacteria and to improve its ability to produce higher-resolution images. Currently, the software uses a pre-built Python library called HoloPy. This library has many limitations and it is not being maintained as often as it used to be. Since only a few functions from the library are used, albeit they are quite complex, they could be reproduced or developed into the system. This means that reliance on the library and its dependencies can be reduced.

Other reconstruction methods outlined in the literature that was reviewed are also suitable. There are image reconstruction and manipulation tools which are written in Java for ImageJ. Wrapper classes could be used to interface with the Java applet, however, this adds more processing and computation to the software.

9.2 Investigation into alternative camera solutions

The IMX477 image sensor was used in the Raspberry Pi High-Quality Camera package. Alternative cameras or image sensors could be substituted in for the camera that is in use. If the image quality is suitable with earlier, smaller versions of the Raspberry Pi Camera, it might be worthwhile implementing this. The smaller Raspberry Pi cameras lack the resolution of the High-Quality camera but come at a cheaper cost (PiShop 2022).

9.3 Extended image capture in an incubator

As suggested by co-supervisor, Dr Mandy Mason, future testing of the bacteria in a lab incubator for extended periods would provide better insight into the colony growth. The *M. tb* cells have a long growth period of over 18 hours. It would be beneficial to capture images of the bacterial growth over one to three weeks. Colony counting methods used in the lab are only carried out after one week when colonies are visible to the naked eye. With such a large FOV, the camera may be able to detect colonies before one week, but testing was only done for 24 hours and yielded little to no visible growth in the medium.

9.4 Additional web server features

The web server that serves the HTML-based website to users on the front end lacks some features which could not be implemented. These features include:

1. Multiple sample viewing capabilities
2. Camera configuration parameters adjustment
3. Full remote access via the internet

Only one sample can be viewed at a time currently. This functionality would need to be expanded if the system is scaled to utilise multiple cameras. Each camera's configuration would need to be adjusted based on the sample type and general cell size. This functionality is lacking and has to be set manually in the back end. These features can be tied together with a secure web server that can authenticate users and allow remote access over the internet.

9.5 Scaling the lensless microscope

The Raspberry Pi 4 Model B has a set of 4 USB ports available. This would allow 4 cameras to connect to it with the possibility of more if the cameras can operate through a USB hub. The benefit of having a scaled solution allows researchers to test different drug susceptibility on the same sample or samples of varying dilution ratios. The hardware design would also need to be updated to support multiple LED housing chambers and slide mounts.

Bibliography

- Arducam (2020). *Arducam IMX477 UVC Camera Adapter Board for 12MP Raspberry Pi HQ Camera*.
- Bishara, Sikora, et al. (2011). “Holographic pixel super-resolution in portable lensless on-chip microscopy using a fiber-optic array”. In: *Lab on a Chip* 11, pp. 1276–1279. DOI: [10.1039/C0LC00684J](https://doi.org/10.1039/C0LC00684J).
- Bishara, Su, et al. (2010). “Lensfree on-chip microscopy over a wide field-of-view using pixel super-resolution”. In: *Optics Express* 18, pp. 11181–11191. DOI: [10.1364/oe.18.011181](https://doi.org/10.1364/oe.18.011181).
- Jericho and Jürgen Kreuzer (2010). “Point Source Digital In-Line Holographic Microscopy”. In: ed. by Ferraro, Wax, and Zalevsky. *Coherent Light Microscopy: Imaging and Quantitative Phase Analysis*. Springer Berlin Heidelberg, pp. 3–30. DOI: [10.1007/978-3-642-15813-1_1](https://doi.org/10.1007/978-3-642-15813-1_1).
- Jiang et al. (2020). “Wide-field, high-resolution lensless on-chip microscopy via near-field blind ptychographic modulation”. In: *Lab on a Chip* 20, pp. 1058–1065. DOI: [10.1039/c9lc01027k](https://doi.org/10.1039/c9lc01027k).
- Jung and Lee (2016). “Real-time bacterial microcolony counting using on-chip microscopy”. In: *Scientific Reports* 6. DOI: [10.1038/srep21473](https://doi.org/10.1038/srep21473).
- Lab, Manoharan (2021a). *Holography and Light Scattering in Python — HoloPy 3.5.0 documentation*. HoloPy Documentation.
- (2021b). *Reconstructing Point Source Holograms*. HoloPy Documentation.
- Ltd., Raspberry Pi Trading (2020). *Raspberry Pi High Quality Camera*.
- Luo et al. (2015). “Synthetic aperture-based on-chip microscopy”. In: *Light: Science & Applications* 4, e261–e261. DOI: [10.1038/lsa.2015.34](https://doi.org/10.1038/lsa.2015.34).
- Martin and Raspberry Pi Foundation (2020). *Raspberry Pi HQ camera board*.

- Mudanyali et al. (2010). “Compact, light-weight and cost-effective microscope based on lensless incoherent holography for telemedicine applications”. In: *Lab on a Chip* 10, pp. 1417–1428. DOI: 10.1039/c000453g.
- Optics, Edmund (2022a). *Resolution*.
- (2022b). *Testing and Targets*.
- PiShop (2022). *Raspberry Pi Camera Board V2 (8MP, 1080p)*. www.pishop.co.za.
- Wiklund, Brismar, and Önfelt (2012). *Acoustofluidics 18: Microscopy for acoustofluidic micro-devices*. Research Gate.
- Zaburdaev (2021). *New mathematical model: How dangerous bacteria form colonies Faculty of Sciences*. Friedrich Alexander University of Erlangen-Nürnberg.

Appendix A

A.1 Link To GitHub

<https://github.com/Cameron-Ray/lensless-microscopy>

A.2 Link To Fusion 360 Models

Microscope Pi Housing
Microscope Slide Mount

A.3 Link To Captured Image Package

All Capture Data
First Capture Series
Second Capture Series
Third Capture Series
Test Target Captures

A.4 Ethics Approval

Application for Approval of Ethics in Research (EiR) Projects
Faculty of Engineering and the Built Environment, University of Cape Town

ETHICS APPLICATION FORM

Please Note:

Any person planning to undertake research in the Faculty of Engineering and the Built Environment (EBE) at the University of Cape Town is required to complete this form **before** collecting or analysing data. The objective of submitting this application *prior* to embarking on research is to ensure that the highest ethical standards in research, conducted under the auspices of the EBE Faculty, are met. Please ensure that you have read, and understood the **EBE Ethics in Research Handbook** (available from the UCT EBE, Research Ethics website) prior to completing this application form: <http://www.ebe.uct.ac.za/ebe/research/ethics1>

APPLICANT'S DETAILS		
Name of principal researcher, student or external applicant	Cameron Luke Ray	
Department	Department of Electrical Engineering	
Preferred email address of applicant:	ryxcam002@myuct.ac.za	
If Student	Your Degree: e.g., MSc, PhD, etc.	BSc (Eng) Mechatronics
	Credit Value of Research: e.g., 60/120/180/360 etc.	40
	Name of Supervisor (if supervised):	Robyn Verrinder (Co-Supervisor: Dr Mandy Mason)
If this is a research contract, indicate the source of funding/sponsorship	N/A	
Project Title	In vivo lensless microscopy using a Raspberry Pi Camera	

I hereby undertake to carry out my research in such a way that:

- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism.

APPLICATION BY	Full name	Signature	Date
Principal Researcher/ Student/External applicant	Cameron Luke Ray		18/08/2022
SUPPORTED BY	Full name	Signature	Date
Supervisor (where applicable)	R.A. Verrinder		2022-08-18
APPROVED BY	Full name	Signature	Date
HOD (or delegated nominee) Final authority for all applicants who have answered NO to all questions in Section 1; and for all Undergraduate research (Including Honours).			
Chair: Faculty EIR Committee For applicants other than undergraduate students who have answered YES to any of the questions in Section 1.			