Development of Novel Imaging Modalities for Soft Matter

A thesis submitted in fulfilment of the requirements for the award of the degree of Doctor of Philosophy

Elliot Steele, MEng, ACGI

December 2022

Department of Physics and Astronomy
University of Sheffield

ABSTRACT

In the past 100 years, a plethora of new microscopy and spectroscopy techniques have been developed, allowing biologists to probe the inner workings of cells and organisms in greater detail than ever before. These techniques, many enabled by the application of fluorescent molecules, range from super-resolution, which reveals structures smaller than the physical limit of resolution to be viewed without resorting to electron microscopy, to techniques capable of detecting nanometre changes in the shape of individual molecules and many more. These approaches are not without drawbacks, however. Super-resolution often results in high levels of phototoxicity, presenting issues for live cell imaging and almost all modern techniques place high demands on hardware and software; if the hardware and software is not sufficiently well designed, these complexities are passed on to end users. This thesis outlines developments to the CairnFocal platform, a DMD based confocal microscope, capable of super-resolution and real-time modality switching. The creation of software to support the novel smfBox open-source confocal smFRET microscope is also described.

ACKNOWLEDGEMENTS

I would like to thank all the members of the Cadby and Craggs labs, both past and present, particularly Ash and Tim for their supervisory roles. Additionally, I am greatly appreciative to Cairn Research, the EPSRC and the CDT in Polymers, Soft Matter and Colloids for their generous funding and work in facilitating the PhD.

COVID19 IMPACT STATEMENT

The work presented in this thesis was carried out between October 2018 and December 2022 and as such was heavily impacted by the COVID19 pandemic. In the UK, the initial lockdown began in March 2020, midway through the 2$^{\text{nd}}$ year of the project. During this time, the University of Sheffield decided to restrict access to laboratories, preventing access to the equipment this work relied upon for several months. Even when access to the labs was restored, the anti-COVID protocols that were in place allowed only one person in the lab at once, preventing the majority of experiments that had been planned prior to the pandemic from taking place. By the time restrictions were lifted, several members of the group had left meaning the group no longer possessed the skills required to perform these experiments. Attempts were made to pivot the work towards that which could be completed with the available resources, however, this means that little data was collected on live biological systems.

GLOSSARY

ABI	Application Binary Interface
ALEX	Alternating Laser EXcitation
APD	Avalanche PhotoDiode
API	Application Programming Interface
ASIC	Application Specific Integrated Circuit
AU	Airy Unit
AXI	Advanced eXtensible Interface
BPAE	Bovine Pulmonary Artery Endothelial
BSP	Board Support Package
CCD	Charge-Coupled Device
CDS	Correlated Double Sampling
CFP	Cyan Fluorescent Protein
CLSM	Confocal Laser Scanning Microscopy
CMOS	Complementary Metal-Oxide-Semiconductor
CPU	Central Processing Unit
CSV	Comma Separated Value
DAC	Digital to Analogue Converter
DLL	Dynamic Link Library
DMD	Digital Micro-Mirror Device
EFL	Effective Focal Length
EMCCD	Electron Multiplying CCD
EPFL	École Polytechnique Fédérale de Lausanne

FCCS	Fluorescence Cross-Correlation Spectroscopy
FCS	Fluorescence Correlation Spectroscopy
FLIM	Fluorescence Lifetime Imaging
FOFO	Full-On, Full-Off
FPGA	Field Programmable Gate Array
FRAP	Fluorescence Recovery After Photobleaching
FRET	Förster Resonance Energy Transfer
FWHM	Full Width at Half Maximum
GaAsP	Gallium Arsenide Phosphide
GFP	Green Fluorescent Protein
GPL	GNU General Public License
GSD	Ground State Depletion
GTK	GIMP toolkit
GUI	Graphical User Interface
HDL	Hardware Description Language
IDE	Integrated Development Environment
ISM	Image Scanning Microscopy
ISR	Interrupt Service Routine
JDK	Java Development Kit
JIT	Just-in-Time
JNI	Java Native Interface
JSON	Java Script Object Notation
JVM	Java Virtual Machine
LCD	Liquid Crystal Display
LDI	Laser Diode Illuminator
LED	Light Emitting Diode
LGPL	GNU Lesser General Public License
MDA	Multi-Dimensional Acquisition
MEMS	Microelectromechanical System
MIT	Massachusetts Institute of Technology
MOS	Metal-Oxide-Semiconductor
MOSFET	Metal-Oxide-Semiconductor Field-Effect-Transistor
MSIM	Multi-focal Structured Illumination Microscopy

MTF	Modulation Transfer Function
NA	Numerical Aperture
OOP	Object-Oriented Programming
OPT	Optical Projection Tomography
PALM	Photoactivation Localisation Microscopy
PAM	Programmable Array Microscope
PCB	Printed Circuit Board
PFI	Programmable Function Input
PLL	Phase-Locked Loop
PMT	Photomultiplier Tube
PSF	Point Spread Function
QML	Qt Modelling Language
RAII	Resource Acquisition is Initialisation
RAM	Random Access Memory
RESOLFT	Reversible Saturable Optical Fluorescence Transitions
RFP	Red Fluorescent Protein
RISC	Reduced Instruction Set Computer
RMS	Royal Microscopical Society
ROI	Region of Interest
ROM	Read Only Memory
sCMOS	Scientific CMOS
SDCM	Spinning Disk Confocal Microscope
SDK	Software Development Kit
SIM	Structured Illumination Microscopy
SiPM	Silicon Photomultiplier
SLM	Spatial Light Modulator
smFRET	Single Molecule FRET
SMLM	Single Molecule Localisation Microscopy
SMT	Surface Mount
SNR	Signal-to-Noise Ratio
SPEM	Saturated Pattern Excitation Microscopy
spFRET	Single Pair FRET
SPI	Serial Peripheral Interface

SSIM	Saturated SIM
STED	Stimulated Emission Depletion
STORM	Stochastic Optical Reconstruction Microscopy
SVM	Support Vector Machine
TCP/IP	Transmission Control Protocol/Internet Protocol
TDC	Time-to-Digital Converter
TDL	Tapped Delay Line
THT	Through-Hole
TIRF	Total Internal Reflectance Fluorescence
TTL	Transistor to Transistor Logic
TTY	Teletype
UWP	Universal Windows Platform
WPF	Windows Presentation Foundation
YFP	Yellow Fluorescent Protein

CONTENTS

CHAPTER

ONE

INTRODUCTION

1.1. Microscopy

The use of optical devices for magnification of small objects dates back thousands of years, with Roman artisans likely using some form of magnifying glasses for detailed work [1, p.9]. The use of magnifying devices for the investigation of objects on the microscopic scale was not performed until the Renaissance, however. It's unclear exactly when the first compound microscope was built but it was certainly before 1665, when Robert Hooke published his famous book 'Micrographia', which contained a collection of copper-plate illustrations of several microscopic objects he had observed though a compound microscope [2]. Since then, both theoretical understanding and microscope design have improved significantly, allowing the microscope to become a widespread tool in many areas of Science and Engineering.

1.1.1. Structure of a Modern Research Microscope

A classical compound microscope consists, at a minimum, of two lens groups: the objective lens, a short Effective Focal Length (EFL) lens placed close to the sample, and the ocular lens, a longer EFL lens responsible for relaying the intermediate image produced by the microscope to the eye [3]. The ocular lens is placed so as to focus the intermediate image at infinity to allow the eye to view the image in a relaxed state. The magnification of the microscope is determined by a combination of the EFL of the objective and EFL of the ocular. The Royal Microscopical Society (RMS) standard specifies a tube length (the distance between the back focal plane of the objective and the intermediate image plane) of 160 mm and a parfocal distance (the distance between the closest surface of the cover slip to the mounting surface of the objective) of 45 mm. The former standard ensures that any RMS compliant objective will work on any RMS compliant microscope and the latter ensures that the microscope will not need to be refocused when switching magnification, e.g., via a rotating nose-piece. In modern microscopy, the eye is often replaced by a digital camera, in which case the intermediate image may be focused directly onto the camera sensor.

The compound microscope described above utilises a finite focal length objective, whereby the objective lens forms the image at the intermediate image plane. This configuration has fallen out of favour in research microscopes, primarily due to the difficulties associated with introducing components such as filters, beam splitters, prisms, etc., without introducing aberrations. Most modern microscopes use infinity corrected objective lenses, which form an image at infinity rather than at a finite position. A second lens, the tube lens, is required to form the image at the intermediate image plane. In an infinity corrected system, the EFLs of the objective and tube lenses determine the magnification, with the oculars only required to relay the image to the eye at unity magnification. This configuration allows for ancillary optical components to be placed in the infinity space between the objective and tube lenses with less stringent flatness requirements. The drawback is that some microscope manufacturers choose to perform some aberration correction in the tube lens rather than the objective, meaning that infinity corrected objectives from different manufacturers are typically no longer interchangeable. Another advantage of this setup is that tube lenses with differing EFLs can be used to increase or decrease magnification (within reason), without changing the location of the intermediate image plane.

1.1.2. Köhler and Critical Illumination

Research microscopes also typically come with the ability to provide well controlled illumination to the sample for transmitted light (diascopic) imaging. This diascopic illumination scheme consists of a light source (e.g., a filament lamp), a collector lens, which collects and collimates the light from the source and a condenser lens which refocuses the collimated light onto the sample. Two diaphragms are placed between the collector lens and the condenser lens to allow the operator to control the Numerical Aperture (NA) of the collector, in order to adjust contrast and resolution, and to control the region illuminated in the sample.

Appropriate alignment of the condenser system is paramount to achieving good imaging results. There are two main schemes for condenser alignment, Köhler Illumination and Critical Illumination. In Köhler illumination, the image of the source is brought into conjugate focus with the microscope's aperture stop (usually located on the back aperture of the objective). This ensures that the image of the fourier transform of the source is conjugate with the sample, ensuring a flat and even illumination, even for sources with inherent structure like filament lamps. Critical illumination is the main alternative to Köhler illumination, where the source is imaged onto the sample. The advantage of Critical illumination over Köhler is that light is transferred from the source to the sample plane more efficiently, however, any structure and non-uniformities in the source will appear in focus in the final image so a highly uniform source is required. In the case of both Köhler and critical illumination, the technique is generally referred to as brightfield microscopy, owing to the fact that the sample appears dark on a bright background field.

1.1.3. Alternative Illumination Schemes

One difficulty with the illumination schemes described above is that the contrast of the final image (i.e., the difference in the measured intensity between structures of interest and the image background) is determined by the absorption of the illumination light as it passes through the sample. As remarked in [4], many samples of biological interest are almost entirely transparent, with different structures within the sample demarcated by changes in refractive index, causing the sample to change the phase of incident light waves but not their amplitudes. The consequence of this is that many biological samples appear low contrast when imaged using the diascopic illumination schemes described in the previous section.

In 1942, Zernike outlined a novel method for improving the contrast in such situations, Phase Contrast Microscopy [4]. The basic concept is based on the observation that the light scattered by most biological samples will have a phase shift of $\sim 90°$ with respect to the light passing directly through the sample. By advancing or delaying the phase of the direct light by 90° while leaving the scattered light unaltered with the introduction of a phase plate, the scattered and direct light will constructively or destructively interfere at the image plane, producing an image where contrast is determined by small phase variations introduced by the sample. Typically, this is achieved by introducing an annular aperture into the condenser assembly, allowing a hollow cone of light to illuminate the sample. The system is aligned such that the illumination cone passes through a phase ring that can be placed either within or outwith the objective. Sensitivity to phase differences can be further improved by introducing some attenuation of the direct illumination path in the phase ring [4].

Zernike also described the mathematics behind a prior technique that also uses an annular aperture in the condenser, darkfield microscopy [4]. In darkfield, the direct illumination light is removed entirely by ensuring that the illumination cone impinges on the objective at such an oblique angle that it is rejected by the objective's aperture stop. In this case, only the light scattered by the sample will be collected by the objective and used to form the final image. In contrast to brightfield imaging, regions of the field that contain no sample will appear dark (hence the name darkfield), with sample appearing light. Since scattering will occur due to differences in refractive index even when no absorption is present, the contrast of transparent biological samples will once again be improved. While a potentially useful technique in some circumstances, darkfield has several drawbacks. Firstly, since the direct illumination light doesn't participate in image formation and the intensity of the scattered light is usually much less than that of the direct light, darkfield requires higher illumination intensities to achieve comparable Signal-to-Noise Ratio (SNR) to brightfield or phase contrast. Additionally, since the angle illumination cone must be sufficiently oblique that it is rejected by the objective aperture, this necessitates the NA of the condenser being higher than the NA of the objective. This generally requires complex and expensive condenser assemblies to accommodate a large range of objectives [5] and prohibits the use with immersion objectives with NA > 1 since condenser assemblies are usually immersed in air. Figure 1.1 shows a comparison between the three techniques discussed.

A recently developed alternative to an annular diaphragm and condenser setup is described in [5]. The author describes the use of a ring of Light Emitting Diodes (LEDs) in place of the traditional condenser setup. By positioning the LED ring in an appropriate position above the objective, the LED ring can provide suitable illumination for performing phase contrast microscopy or, by brining the LED much closer to the objective, darkfield. When performing phase contrast, the exact positioning of the LED ring is not overly critical, as long as the image of the LED ring falls within the image of the system's phase ring at the back focal plane of the objective (which can be viewed with the use of a Bertrand lens commonly available in microscope oculars). Since the image of the ring will appear smaller at the back aperture as the LED ring is moved further from the objective due to perspective effects, by choosing suitably sized LED rings, in arbitrary illumination working distances can be achieved, without the need to design long working distance condenser lenses. The main drawback of this approach is that since the size of the image of the LED ring is dependent on the magnification of the objective, so adjusting the position of the ring is likely to be required when swapping objective lenses. The paper describes the use of concentric, individually addressable LED rings to help alleviate this issue.

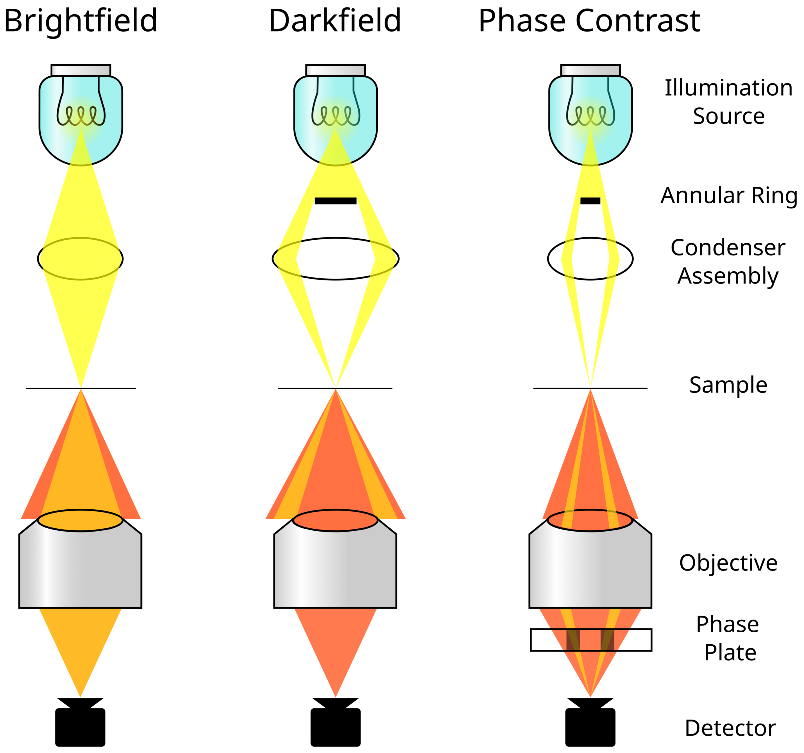


Figure 1.1.: Comparison between the setups used to perform brightfield, darkfield and phase contrast microscopy. Illumination light is shown in yellow. Light scattered from the sample is shown in red. Regions where both illumination and scattered light are present are shown in orange. The dark regions in the phase plate present in the phase contrast scheme correspond to the regions that phase delay the light passing through them by ±90°.

1.1.4. Fluorescence Microscopy

The process of fluorescence was first discovered in the mid 1800s, with George Stokes coining the term in his 1852 paper [6]. When light of particular wavelengths is incident on a fluorescent material, some of the incident light is absorbed and re-emitted at a longer wavelength. It wasn't until the early 1900s that a suitable explanation of the causes of fluorescence at an atomic level was developed [7]. In short, fluorescence occurs when a susceptible molecule absorbs an incident photon, promoting an electron to a higher energy state. The excited electron will tend to decay to the ground state after a short time (typically of the order of nanoseconds), emitting a photon as it does so. The general shift towards longer wavelengths noted by Stokes (referred to as the Stokes Shift) is caused by the excited electron losing energy to non-radiative processes before relaxation into the ground state.

Fluorescence was first encountered in the context of microscopy in the early part of the 20th century, when scientists were attempting to improve the resolution of microscopes by making use of ultraviolet illumination [8]. It was initially considered a nuisance, as autofluorescence of samples acted to degrade images, however, it was quickly realised that if the excitation light could be prevented from reaching the detector, then detection of autofluorescence would could provide significant contrast improvements over brightfield, with the first fluorescence microscopes being developed in the early 1910s. In 1941, Albert Coons developed the first immunofluorescent labelling techniques [9], allowing fluorescent molecules to be attached to specific targets within cells and by extension, allowing for high contrast imaging of specific cellular structures.

Starting in the 1960s, attempts to isolate the molecules responsible for bioluminescence in Aequorea jellyfish resulted in the development of two new fluorophores and and associated microscopical techniques [10]. One of the two molecules, aequorin, is calcium sensitive, only capable of fluorescing in the presence of Ca^{2+} ions. In [11], the authors describe microinjecting aequorin into the muscle cell of an Acorn Barnacle in order to observe calcium transients via fluorescence. This encouraged the development of several other calcium sensitive fluorescent dyes, with a particularly important contribution made by Roger Tsien in 1980 [12] of a Ca^{2+} sensitive dye which was relatively insensitive to Mg^{2+} or pH. Subsequent work resulted in a range of dyes sensitive to various intracellular environmental conditions, e.g., ion concentrations, pH, voltage, etc. The ability to detect intracellular changes and even image using these dyes has revolutionised fields such as physiology and neuroscience. The second fluorophore isolated from Aequorea was Green Fluorescent Protein (GFP). As well as it being possible to use as the fluorophore in immunofluorescent labelling, providing both excitation and emission in the visible spectrum, it was demonstrated in 1994 by Martin Chalfie *et al.*, that genes could be introduced into the DNA of organisms in order to have them express GFP. This could be used as an indicator of gene expression, allowing experimenters to verify that their DNA manipulations have been successful. It also offers a highly specific labelling technique that can be used to fluorescently label the particular proteins that make up structures of interest within the organism [13].

Fluorescence microscopy is performed using setups that are for the most part similar to the arrangements described in the previous sections. Illumination may be provided through the diascopic illumination pathway described above and pictured in Figure 1.1 but is more commonly provided through the episcopic pathway (i.e., through the objective lens). This is possible since fluorophores usually emit in all directions, irrespective of the direction of illumination and there is therefore no particular advantage to illuminating diascopically in most cases. Since fluorescent emission is of a significantly lower intensity than the excitation light required, emission and excitation light must be separated before fluorescence can be detected. In modern research microscopes, this is performed by exploiting the Stokes Shift of the fluorophores. The Stokes shift ensures that the emission wavelength is longer than the excitation wavelengths and this differential allows the removal of the excitation light from the light collected by the objective. Typically, the sample is illuminated by a small band of wavelengths, matched to the excitation spectrum of the fluorophore, either by utilising narrow bandwidth sources (e.g., lasers) or by filtering a broadband source (e.g., Xenon arc lamps) with a suitable excitation filter. Light collected by the objective is similarly filtered before it reaches the detector, using an emission filter which absorbs the excitation wavelengths while allowing the emission wavelengths to pass. When using the episcopic illumination pathway, a dichroic mirror is used to allow the excitation and emission pathways to converge before the objective lens.

1.1.5. Detectors

Early microscopes were designed to work primarily with the human eye, with scientists often drawing their observations by hand. Microscopes were sometimes adapted to use photographic plates and eventually film, however, with the advent of modern electronic photodetectors most modern research microscopes use digital photodetectors, with the collected data automatically transferred to computer. Modern detectors fall roughly into two distinct categories, area detectors and point detectors.

Area detectors usually refers to cameras, of which there are two primary topologies, Charge-Coupled Device (CCD) and Complementary Metal-Oxide-Semiconductor

(CMOS). When using a camera as a detector, the oculars are not used and the tube lens focuses the image of the sample directly onto the camera sensor. The sensor is comprised of an array of pixels, which each sample the irradiance of the image; when read out to a computer, the samples give a digital representation of the image formed by the microscope. In most cases, camera sensors have an exposure time, during which photons incident on a pixel have a chance of exciting an electron in the pixel's semiconductor substrate into the conduction band. These 'photoelectrons' are stored in the pixel until the end of the exposure period, at which point the number of photoelectrons is measured and the pixels are cleared in preparation for the next exposure.

In a CCD, the pixels are formed of p-doped Metal-Oxide-Semiconductor (MOS) capacitors, which are biased with a positive voltage during camera exposure [3, p.115-121]. When a photon creates an electron-hole pair, they are separated, with the electron heading towards the biasing electrode and the hole into the bulk of the pixel. At the end of exposure, the electrons stored in each pixel are shifted down the camera sensor, with the bottom row shifted into a shift register. The electrons stored in each element of the shift register are amplified and digitised one element at a time, with the next row shifted in after digitisation of the last row has completed. A series of electron multiplying gain stages may be placed between the shift register and the output amplifier, a topology known as Electron Multiplying CCD (EMCCD), which can dramatically increase the sensitivity of the sensor at the expense of added noise due to the stochastic nature of the electron multiplication stages. The primary difficulty with modern CCD sensors is that the serial amplification and digitisation as well as the requirement to shift the electrons between pixels limits the maximum framerate of the cameras. Some sensors additionally suffer from bloom or 'smear' artefacts caused by photoelectrons being collected during the readout phase. To avoid this issue, manufacturers will sometimes mask off every other column of pixels to prevent light from reaching them to produce what is referred to as an interline-transfer CCD. At the start of the readout phase, the photoelectrons are immediately shifted into the masked columns, which are then read out as in a normal CCD sensor.

An alternative to the CCD architecture is the CMOS architecture. In the CMOS architecture, in addition to a capacitor that stores electrons, each pixel also contains an amplifier. Multiple analogue to digital converters can be utilised in order to perform digitisation in parallel and reduce readout times (typically a single A/D converter is used per row of the sensor). Unlike in CCD technology, in CMOS sensors, pixel readout and clearing are distinct operations, meaning a separate 'clear' signal must be sent to each pixel before the start of exposure. While CMOS sensors can be made significantly faster than CCD sensors, allowing for both higher frame rates and higher resolution sensors, it comes as the cost of increased noise. One of the primary sources of noise in CMOS cameras is from ineffective clearing of the pixels before exposure leaving a small amount of charge in the pixel before exposure starts. To solve this Correlated Double Sampling (CDS), whereby a measurement of the charge in the pixel at the start of exposure is taken and subtracted off the final value after exposure, can be employed. Some scientific cameras perform CDS automatically in the pixel itself, these are commonly referred to as Scientific CMOS (sCMOS) sensors. Another issue with CMOS technology is that the quantum efficiency (i.e., the chance that an incident photon is converted to a photoelectron) tends to be lower than that of CCD chips. This is due to the per-pixel circuitry reducing the pixel fill factor (i.e., the percentage of a pixel's area that is light sensitive). sCMOS cameras typically utilise either an array of micro-lenses to redirect light onto the photosensitive region of each pixel or back-thinning, where acid is used to thin the camera sensor in order to make it thin enough to be illuminated from what would usually be the back side.

Noise in camera sensors can be separated into three types: read noise, shot noise and fixed pattern noise. Read noise is primarily introduced by the electronics of the camera during readout and as a result is always present, doesn't vary with signal intensity and dominates in low light conditions. Fixed pattern noise is a result of pixel to pixel variations and as a result is almost non-existent in CCD technology. In CMOS sensors, however, it is difficult to accurately control the gain of each of the on-pixel amplifiers and these variations result in some pixels being more sensitive than others. When present, fixed pattern noise varies linearly with signal intensity and typically dominates at high light intensities. Shot noise isn't due to the camera sensor itself and is instead due to the quantal and stochastic nature of light. The magnitude of the shot noise varies with the square root of signal intensity and dominates the total noise in the transition region between the read noise and fixed pattern noise dominated regions. In addition to these noise sources, spontaneous creation of electron-hole pairs, often due to heat, contributes a slow but relatively constant accumulation of electrons within a pixel, even in complete darkness. This accumulation is termed dark current. In a well-designed and well-cooled sensor, however, neither CCD nor CMOS sensors suffer from significant dark current over the exposure times commonly used in microscopical applications. Dark current may become a problem if long exposures ($\gg 1\,\mathrm{s}$) are required, as is the case when studying low light systems such as some forms of bioluminescence. In these cases, EMCCD sensors cooled to $< -80\,^{\circ}\mathrm{C}$ are typically used to minimise dark current while maximising sensitivity.

While area sensors spread the output of the microscope across the sensor in an attempt to build a digital representation of the signal, when using point detectors, the entire output of the optical system is focused onto a single detector element which in turn produces an indication of the instantaneous intensity. Point detectors come in several varieties, Photomultiplier Tubes (PMTs), Avalanche PhotoDiodes (APDs), Silicon Photomultipliers (SiPMs) and hybrid PMTs to name a few. While each of the technologies vary slightly in parameters such as quantum efficiency and noise characteristics, they can generally all be used in one of two modes depending on the topology of the circuitry attached to the detector element. When used in linear mode, the detectors provide a continuos output current that is proportional to the photon flux incident on the detector. Typically, when used in this mode, the current is further amplified, sampled at a regular interval and digitised (or on some occasions viewed on an oscilloscope). The detectors can also be used in a single photon detecting mode (sometimes referred to as Geiger mode) where the output is a stream of pulses indicating the arrival of individual photons. When used in this mode, a Time-to-Digital Converter (TDC) is used to convert the stream of pulses into a stream of timestamps indicating the arrival time of individual photons. When used in single photon mode, two parameters are of particular importance: firstly, the dark count rate, which is roughly analogous to the dark current in a camera sensor, determines the noise floor of the detector; secondly, the dead time, which dictates the minimum detectable time between two photons that allows for the detection of both photons.

While the entire output of the microscope is usually focused onto a point detector, recently, arrays of point detectors have been utilised in an attempt to gather spatial information while still benefiting from the performance of point detectors (i.e., the ability to count and timestamp the arrival of individual photons). For example, the Zeiss Airyscan uses an array of Gallium Arsenide Phosphide (GaAsP) PMTs arranged in a hexagonal pattern in order to spatially sample the output of a Confocal Laser Scanning Microscopy (CLSM) (see Section 1.2.2) to improve resolution [14]. Additionally, attempts have been made to create camera sensors using single photon counting technology rather than CCD or CMOS architectures. While these have traditionally been relatively low resolution sensors with poor noise performance compared to their point detector counterparts, in 2020 Canon succeeded in developing a 1 megapixel single photon counting chip and in 2021 produced a 3.2 megapixel variant, with dark counts as low as 2 counts/s [15], [16].

1.2. Diffraction Limit and Super-Resolution

When increasing the magnification under which a sample is viewed, from for example 4× to 10×, finer structures within the sample become visible. This pattern will continue for a time, however, there will come a point when further increases to magnification do not reveal finer structures. If magnification is increased beyond this point, the image of the structures within the sample will become larger but not better defined, appearing significantly blurred. When imaging an infinitesimally small point source, the image would be spread out into a 'blob' known as the Point Spread Function (PSF). This effect, which places a fundamental physical limit on the resolution achievable with light microscopy, was first described by Abbe in 1873 [17]. As a brief but incomplete description of why this phenomenon occurs, consider the following situation: a thin sample, illuminated by a monochromatic plane wave of wavelength λ from the left, imaged on the right by a circular lens.

If the sample is assumed to extend infinitely in the xy-plane at $z = 0$ then following a similar argument to that of Goodman [18, p.55-61], first define the amplitude transmission function of the sample as:

$$t_A(x, y) = \frac{U_{0^+}(x, y; 0)}{U_{0^-}(x, y; 0)} \tag{1.1}$$

i.e., as the ratio between the field amplitude to the right of the sample, U_{0+}, to the field amplitude incident on the sample, U_{0-}. By the Fourier theory, t_A may be decomposed into an infinite number of complex exponentials, stated mathematically:

$$t_A(x, y) = \iint_{-\infty}^{\infty} A(f_X, f_Y) e^{j2\pi(xf_X + yf_Y)} df_X df_Y \tag{1.2}$$

where f_X and f_Y are spatial frequencies in the x and y dimensions respectively and $A(f_X, f_Y)$ is a complex valued function determining the amplitude and initial phase of the complex exponential associated with the spatial frequencies f_X and f_Y. Now consider the phaser form of the equation associated with a plane wave travelling with the direction determined by the vector $\vec{k}$, where $|\vec{k}| = \frac{2\pi}{\lambda}$:

$$\begin{aligned} U_z(x, y) &= e^{j\vec{k}\cdot\vec{r}} \\ &= e^{j(k_x x + k_y y + k_z z)} \\ &= e^{j\frac{2\pi}{\lambda}(\alpha x + \beta y + \gamma z)} \\ &= e^{j\frac{2\pi}{\lambda}(\alpha x + \beta y)} e^{j\frac{2\pi\gamma z}{\lambda}} \end{aligned} \tag{1.3}$$

Where α, β and γ are the direction cosines associated with the direction of the plane wave's propagation, governed by the equation $\alpha^2+\beta^2+\gamma^2=1$. At $z=0$ this reduces to

$$U_0(x,y)=e^{\frac{j2\pi}{\lambda}(\alpha x+\beta y)} \tag{1.4}$$

by making the substitutions $f_X=\frac{\alpha}{\lambda}$ and $f_Y=\frac{\beta}{\lambda}$ into Equation 1.2, it becomes

$$t_A(x,y)=\iint_{-\infty}^{\infty} A\left(\frac{\alpha}{\lambda},\frac{\beta}{\lambda}\right)e^{\frac{j2\pi}{\lambda}(\alpha x+\beta y)}d\frac{\alpha}{\lambda}d\frac{\beta}{\lambda} \tag{1.5}$$

The complex exponential inside the double integral is of the same form as equation 1.4, therefore, the field amplitude just to the right of the thin sample can be considered to be the superposition of an infinite number of plane waves with direction cosines $\alpha=\lambda f_X$ and $\beta=\lambda f_Y$ with $A\left(\frac{\alpha}{\lambda},\frac{\beta}{\lambda}\right)$ determining the various amplitudes and phases of the constituent plane waves. $A\left(\frac{\alpha}{\lambda},\frac{\beta}{\lambda}\right)$ is referred to as the angular spectrum of $t_A(x,y)$. As a sanity check that this result will hold not just for the plane $z=0$ but for all $z>0$, return to Equation 1.3 and observe that for a plane wave:

$$\begin{aligned}U_d(x,y)&=e^{j\frac{2\pi}{\lambda}(\alpha x+\beta y)}e^{j\frac{2\pi\gamma d}{\lambda}}\\&=U_0(x,y)e^{j\frac{2\pi\gamma d}{\lambda}}\end{aligned} \tag{1.6}$$

i.e., the field amplitude at the plane $z=d$ may be determined by simply multiplying the field amplitude at $z=0$ by $e^{\frac{j2\pi\gamma d}{\lambda}}$. This multiplicative factor may be expressed in terms of f_X and f_Y by observing:

$$\begin{aligned}\gamma&=\sqrt{1-\alpha^2-\beta^2}\\&=\sqrt{1-(\lambda f_X)^2-(\lambda f_Y)^2}\end{aligned} \tag{1.7}$$

The propagation by a distance, d, of a given angular spectrum through free space may therefore be represented by the transfer function:

$$H_d(f_X,f_Y)=e^{j\frac{2\pi d}{\lambda}\sqrt{1-(\lambda f_X)^2-(\lambda f_Y)^2}} \tag{1.8}$$

By making the approximations $\lambda f_X << 1$ and $\lambda f_Y << 1$, i.e., $\sqrt{1-(\lambda f_X)^2-(\lambda f_Y)^2}\approx 1-\frac{(\lambda f_X)^2}{2}-\frac{(\lambda f_Y)^2}{2}$ (which is equivalent to the paraxial approximation in geometrical optics) then it can be seen that:

$$
\begin{aligned}
H_d(f_X, f_Y) &\approx e^{j\frac{2\pi}{\lambda}d} e^{-j\frac{2\pi}{\lambda}\left(\frac{(\lambda f_x)^2}{2}+\frac{(\lambda f_Y)^2}{2}\right)} \\
&= e^{jkd} e^{-j\pi\lambda d(f_X^2+f_Y^2)}
\end{aligned}
\tag{1.9}
$$

which is equivalent to the transfer function associated with Fresnel diffraction [18, p.72].

In the proceeding analysis, it is assumed that $\alpha^2 + \beta^2 \leq 1$, however, since $\alpha = \lambda f_X$ and $\beta = \lambda f_Y$ and in principal the spacial frequencies contained in $t_A(x, y)$ may extend to infinity, this will not always be the case. In the case that $\alpha^2 + \beta^2 > 1$ then γ will be imaginary and $H_d(f_X, f_Y)$ (Equation 1.8) will reduce to a decaying exponential, or equivalently, the waves associated with spatial frequencies higher than $\frac{1}{\lambda}$ will be evanescent and their amplitudes will decay to close to 0 within a few wavelengths of the sample. This is the first fundamentally limiting factor in the resolution of fine structures within the sample, though it is often forgotten in the discussion of the diffraction limit.

The second, and generally more limiting, factor is the finite extent of the aperture of any physically realisable imaging system. Each spatial frequency component is equivalent to a plane wave travelling towards the objective lens of the optical system, with higher spatial frequencies travelling at a greater deviation from the optical axis. In the parlance of geometrical optics, each spatial frequency can therefore be considered as a parallel ray bundle travelling at an angle to the optical axis. Geometrical optics tells us that in any physically realisable optical system there will be a component that limits the maximum angle that can be accepted by the optical system, the aperture stop [1, p.183], and therefore the spatial frequencies associated with these rejected ray bundles will be filtered by the optical system. Any physically realisable optical system will act as a low pass filter, with the cut off frequency determined its NA.

This was the result first outlined by Abbe [17], who also noted that shorter wavelengths are diffracted at a shallower angle and therefore more readily accepted by an optical system. He postulated that a periodic structure could be resolved if the central and at least one of the 1st diffracted orders is transmitted by the optical system, resulting in the approximate formula for the limit of resolution as:

$$
d = \frac{\lambda}{2\mathrm{NA}_{\mathrm{objective}}}
\tag{1.10}
$$

Abbe also proposed a limit for resolution in the axial direction:

$$d = \frac{2\lambda}{\mathrm{NA}^2_{\mathrm{objective}}} \tag{1.11}$$

Abbe's analysis was later improved upon by Lord Rayleigh [19] who determined the PSF of an aberration free optical system with a circular aperture stop was an Airy function. He also refined the Abbe resolution limit by claiming that two point sources of light could just be resolved if they are placed a distance apart such that the maximum value of the Airy function produced by imaging one of the point sources falls exactly on the first minimum of the Airy function produced by imaging the other, stated mathematically:

$$d = \frac{0.61\lambda}{\mathrm{NA}_{\mathrm{objective}}} \tag{1.12}$$

The Rayleigh resolution criterion is particularly simple to utilise, however as noted above, only applies to the very specific case of diffraction by an optical system with no aberration and a circular aperture, a condition that isn't always met in practical optical systems [20]. An alternative definition of the limit of resolution was proposed by Sparrow in [21]. Sparrow noted that at the separation chosen as the resolution limit in the Rayleigh criterion, there will be still be a dip in intensity between the two PSFs. If the spacing between the sources is further reduced, the PSFs will be brought closer and the intensity dip will become shallower. At some point the PSFs will be sufficiently close that there is no dip between them and Sparrow chose corresponding distance between the sources as a resolution limit. Since this dip will be present for all well behaved PSFs, even those suffering from aberration, the Sparrow criterion can be employed as a measure of resolution in many more optical systems than the Rayleigh limit can. The Sparrow criterion, however, suffers from being significantly more complicated to apply than the Rayleigh criterion.

While a useful parameter in determining the performance of an optical system, considering resolution in isolation may be misleading and it is important to define exactly what is meant by the resolution limit. For example, the Abbe formulation implies that a diffraction grating consisting of sharp rectangular slits (or equivalently $t_A(x, y)$ is a square wave) would be "resolved" if the first diffractive order was transmitted by the optical system. From the Fourier perspective, however, it's clear that this would result in only the fundamental frequency components of the square wave contributing to image formation and the resulting image would be a sinusoid. The grating may be resolved in the sense that it is possible to differentiate it from a uniform sample, however, the resulting image doesn't accurately represent the structure of the object. Using resolution limits based on the minimum separation between two point sources has similar issues.

Apodization or techniques similar to those outlined in [22] can be used in order to improve the resolution as measured by the Sparrow criterion, however, may distort the images significantly.

Even though all definitions of resolution have problems, it's useful to be able to put a single number on the performance of a microscope when discussing super-resolution techniques, as long as care is taken to ensure the same definition is used for all comparisons. In this thesis, the preferred measure of resolution is the Full Width at Half Maximum (FWHM) of the central lobe of the image of a small bead. The size of bead is chosen such that the diameter is smaller than the resulting FWHM. Improvements in resolution are then described in terms of relative FWHMs measured with the various techniques. Care is taken to qualitatively compare the images produced by the different techniques in order to detect any distortions present in the image.

The discussion thus far has assumed perfectly coherent illumination of the sample with a monochromatic source. In reality this is impossible, however, narrow bandwidth sources with long coherence lengths (i.e., lasers) are readily available [1, p.324] allowing this condition to be met to essentially arbitrary degrees of accuracy. The opposite condition, that of fully incoherent illumination, presents a difficulty for the evaluation of the field amplitude. In the monochromatic coherent case, the field amplitude at any given position can be described in terms of a single phasor. In the coherent and/or polychromatic case, however, the phases of the waves superimposed at a particular location will be unrelated to one another and the resultant phase will therefore vary randomly over time [18, p.131], requiring statistical analysis to describe. Thankfully, the instantaneous field amplitude is unmeasurable within the confines of a typical microscopy experiment, with the measurable quantity instead being the square of the electric field amplitude, typically averaged over many periods of the constituent light waves, over which time the statistical effects will average out. In the case of incoherent illumination, the phase of $t_A(x, y)$ may be ignored and the irradiance may be calculated directly from the irradiance of the source wave and the modulus of $t_A(x, y)$ [23]. Fully incoherent illumination of a sample, while convenient for calculations, is also not physically realisable [23]. In reality the coherence of the illumination will fall somewhere between the two extremes of fully coherent and fully incoherent, a condition known as partial coherence. The mathematics of partial coherence was first investigated by van Cittert in 1934 [24] and later by Zernike in 1938 [25], both relying on statistical analysis for their formulations. In 1951, H. H. Hopkins refined the analyses of van Cittert and Zernike, producing a model of partial coherence that didn't rely on statistical methods [23] and later utilising it to produce a detailed description of image formation in optical systems [26].

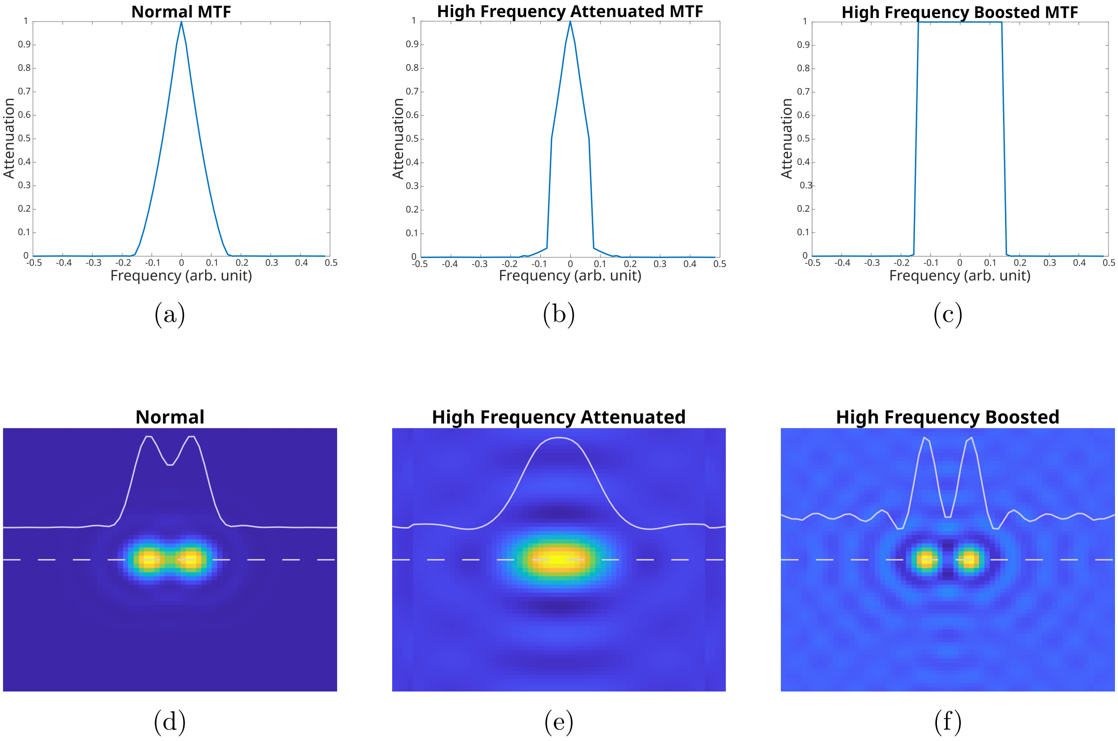
In [23], Hopkins describes how fully incoherent imaging of a sample isn't possible using transmitted light from an extended source, however, mentions that the conditions of incoherent illumination can (at least in theory) be met if the source is self luminous. In fluorescence microscopy, illumination is used to excite the fluorophores in the sample. After a short period of time the fluorophore relaxes to its ground state, emitting a photon. These emitted photons are used to form the image at the detector, with the illumination light filtered out. Since both excitation and emission of fluorophores are stochastic processes, even when illuminated by a coherent source, the process of fluorescence will act to disrupt the any coherence that may be present in the illumination light. When imaging fluorescence, the sample can be reasonably assumed to act as a self luminous object and can therefore be described using the simple mathematics of fully incoherent illumination. Since this thesis is primarily concerned with fluorescence microscopy, from now on imaging will be assumed to be incoherent unless otherwise stated.

1.2.1. Deconvolution

As outlined in the previous section, optical systems are unable to transmit all spatial frequencies associated with a sample due to their finite aperture. In general, an optical system will also act to attenuate certain frequencies more than others. Put more formally, if an optical system can be considered to be spatially invariant, then the output of the system can be determined by a convolution between the input distribution and the system's PSF. By the convolution theorem, this can equivalently be expressed as the system performing a multiplication between the angular spectrum of the input and the fourier transform of the PSF. The fourier transform of the PSF is referred to as the Modulation Transfer Function (MTF) and completely describes the effects of the system [1, p.579].

The diffraction limit can be explained as being due to the finite support of the MTF, however, the resolution of an optical system (as measured by the Rayleigh or Sparrow criteria) can be significantly influenced by how much particular frequencies are attenuated by the MTF of the system. Figure 1.2 illustrates this point, by simulating the imaging of two point sources using several different MTFs. The ability to resolve the point sources varies drastically despite all three of the MTFs being of equal support.

As its name suggests, deconvolution attempts to improve image quality by undoing the convolution that is performed optically by the imaging system. Since the convolution operation can be understood as a point-by-point multiplication between the angular spectrum of the input signal and the MTF of the optical system, naively one might assume that the input signal could be perfectly recovered by dividing the Fourier transform of the



(a) (b) (c)

(d) (e) (f)

Figure 1.2.: A comparison of the two point resolution associated with several MTFs. An Airy pattern was generated to serve as the 'normal' PSF. The MTF was calculated by Fourier transforming the PSF. (a) shows a plot through the centre of this MTF. The MTF was used to derive the two other MTFs. To create (b), the high frequencies of the original MTF (> 0.078 arb. unit) were reduced in amplitude by a factor of 10. (c) was created by dividing each of the Fourier components within the support of the original MTF by their absolute value in order to set their amplitudes to 1 while maintaining their phases. (d)-(f) show the result of using the PSFs corresponding to the three MTF to convolve an image of two closely spaced point sources. Profile plots through the centre of each image are imposed on top. The point sources were placed such that they would just be resolved by the Rayleigh criterion after convolution. In (e), the point sources are clearly no longer resolved by the Sparrow criterion (the Rayleigh criterion no longer being applicable). In (f), the sources are resolved better than in (d), however, at the expense of more 'ringing' artifacts. The three MTFs have identical support and therefore the three theoretical optical systems would accept the same spatial frequencies. This shows that changing the attenuation of the various spatial frequencies (as is done when performing deconvolution) has appreciable affects on the resolution as defined by the Sparrow/Rayleigh criteria, but not on the resolution as described by the Abbe criterion.

output signal by the system's MTF in order to 'undo' the convolution. This immediately runs into difficulties, however, due to the compact support of the MTF requiring division by zero; some frequencies are completely filtered out by the optical system and cannot be recovered in this manner. Even when the process is adjusted to only perform division in regions where the MTF is non-zero, practical difficulties remain. In all real imaging systems, there will be noise present on the output (e.g., camera read noise). This noise is typically white, i.e., contains power at all spatial frequencies. At frequencies where the optical system attenuates significantly, it is likely that the power contained in the noise is comparable or larger than the power contained in the signal. Amplification of these frequency components will have the result of boosting the noise and degrading the SNR of the image.

The problem of how to efficiently and accurately perform deconvolution in the presence of noise has been explored extensively in the literature [27], [28]. Many approaches have been suggested, including linear filters such as Wiener filters [29] and least squares [30], non-linear filters [31], statistical techniques such as maximum likelihood [32] and iterative techniques such as the Richardson-Lucy algorithm [33], [34]. An interesting subclass of deconvolution is blind deconvolution, where the algorithm attempts to estimate not just the input to the optical system but the PSF of the system itself [35]. Though the appeal of being able to perform deconvolution without explicitly measuring the PSF of the system is obvious, in practice performing such a task is non-trivial and most implementations suffer from degeneracy issues and large numbers of parameters [28].

1.2.2. Confocal

Widefield techniques achieve high signal-to-noise ratio by collecting as much of the emitted light as possible, however, this comes at the cost of poor axial resolution. Considering the case of fluorescence microscopy, the whole field, including parts out of the focal plane, is illuminated[1]. These out of focus regions of the sample will fluoresce and the emission will be indiscriminately collected by the objective, resulting in a final image containing both in focus and out of focus structures. The presence of the out of focus structures acts to blur and lower the contrast of the in focus structures. 3D deconvolution techniques, as outlined in 1.2.1, can be utilised to post process the data to improve the axial resolution [27], however, this requires z-stacks to be acquired which isn't always desirable due to lower imaging speed, larger data sizes, longer processing times and higher phototoxicity. Confocal techniques optically reject the light from out of focus planes by

[1] The exception to this is light-sheet microscopy, in which a thin plane of light passing approximately perpendicular to the objective illuminates the sample.

introducing a pinhole (or set of pinholes) to a sample conjugate plane [36].

Laser Scanning Confocal

The most basic form of confocal microscopy is that of Confocal Laser Scanning Microscopy (CLSM) [36]. A CLSM focuses a light source (typically a laser) onto a diffraction limited spot in the focal plane, causing fluorescence from solid cones above and below. Fluorescence from both these cones will once again be collected by the objective. Rather than focusing this light directly onto the detector, however, it is focused onto a small pinhole. The light passing through this pinhole is collected and focused onto a point detector. Light from out of focus planes will come into focus either slightly before or slightly after the pinhole. From a geometrical optics perspective, the light from these planes will therefore be spread over a disk at the pinhole plane, with the majority of the energy falling outside the pinhole's aperture. Typically, the diameter of the pinhole is usually set to 1 Airy Unit (AU) as this results in a good compromise between optical sectioning ability and signal-to-noise ratio. An Airy Unit is defined as the diameter of the circle described by the first zero of the PSF of an Airy-type microscope, i.e.

$$1\,\mathrm{AU} \equiv \frac{1.22\lambda_{em}M}{\mathrm{NA}_{objective}} \tag{1.13}$$

where M is the magnification of the optical system between the sample and the pinhole. CLSMs are often equipped with variable size pinholes to allow the user to optimise for different objectives. To construct a 2D image, the illumination volume must be scanned over the entire sample. This can be performed by either moving the sample or, more commonly, by employing optomechanical devices like mirror galvanometers to scan the laser over the sample.

While confocal provides a significant improvement in optical sectioning over widefield, CLSM suffers from low frame rates due to only collecting light from a single point in the sample at once. To collect a 1024 × 1024 pixel image with a $1\,\mu\mathrm{s}$ dwell time, imaging will take $\sim 1\,\mathrm{s}$. This presents difficulties for the imaging of dynamic processes or samples that move significantly over that timescale. Reducing the dwell time to increase imaging speed will also reduce the amount of light collected, which is already significantly less than equivalent exposure widefield systems due to the rejection of out of focus light. To counteract this effect, the illumination intensity must be increased, however, doing so runs the risk of introducing issues of phototoxicity, saturation and photobleaching [37]. Modern CLSM often employ resonant galvanometers, which continuously scan at a particular resonant frequency. This can improve the speed of imaging to video rates [38], however, this suffers from the same issues as low dwell time non-resonant scanning confocal. In the case an adjustable pinhole is present, the user can opt to open the pinhole in order to improve SNR by letting more out of focus light reach the detector. This will allow for faster imaging at the expense of poorer optical sectioning performance.

Though not usually considered as a technique that grants an increase in resolution, it was demonstrated in a series of papers by Sheppard from the 1970s and 1980s that confocal (and some variations on it) are at least theoretically capable of out performing the resolution limit of a conventional widefield microscope [22], [39]–[41].

Spinning Disk confocal Microscopy

As explained in the previous section, CLSMs struggle to perform high-speed imaging due to only collecting light from one point in the sample at any given time. The Spinning Disk Confocal Microscope (SDCM) increases confocal imaging speeds by collecting light from multiple points simultaneously [42], [43]. As its name suggests, a SDCM uses a disk with many pinholes, arranged in an Archimedes Spiral [44]. Unlike the CLSM, the pinholes are placed in both the excitation and emission pathways. Rather than the illumination light being focused directly onto the sample, the illumination is expanded to cover a section of the disk containing multiple pinholes. The light passing through the pinholes is relayed to the sample by the microscope optics, resulting in an image of the pinholes being formed at the focal plane of the objective. Emitted light is once again focused by the microscope optics onto the disk, where the pinholes reject out of focus light as in the CLSM case. SDCMs use area detectors rather than point detectors. By synchronising the rotation speed of the disk to the exposure time of the camera such that the array of illumination regions sweeps out the entire field an integer number of times, a confocal image is formed [36], [44].

While SDCMs allow for superior imaging speeds when compared to CLSMs, it does come with some drawbacks. Firstly, the illumination power is split between multiple illumination regions, a more powerful source is required to achieve the same illumination intensity at the sample. This issue is compounded by the fact that pinholes are placed in the excitation pathway, meaning a significant portion of the excitation light is prevented from reaching the sample at all. A novel solution to this problem was produced by the Yokogawa Electrical Corporation in 1992. The Yokogawa system utilises a pair of synchronised, matched spinning disks, the first containing microlenses instead of ordinary pinholes [45]. The microlenses focus laser light onto the pinholes of the second disk, which acts exactly as the disk in a traditional SDCM. The dichroic is placed between the two disks so that the microlenses are only present in the excitation path. The addition of microlenses increases the utilisation of the illumination light by over 10× [45]. It should be noted that the recent proliferation of low cost, high power laser diodes has enabled the production of several effective SDCM that don't rely on the use of microlenses.

A second issue with SDCM is that the pinholes are of a fixed size, reducing the opportunities for optimisation of the imaging protocols. As discussed previously, the optimal size for the pinholes is generally considered to be 1 AU, however, in a CLSM slightly larger may be used to increase signal and slightly smaller may be used to increase resolution. Even if the pinhole size could be changed, the effects are not as simple as they are in the CLSM case. Since the pinholes are used to produce the illumination pattern, increasing their size has the effect of letting more out of focus light through the pinhole but also of increasing the size of the illumination volume.

A related issue is that of inter-pinhole spacing. Since multiple regions of the sample are illuminated simultaneously, some out of focus light that would be rejected by the pinhole in a CLSM system may pass through adjacent pinholes in a SDCM, contributing a high background [46]. Increasing the spacing between pinholes will reduce this effect, however, this is not possible on most systems. The X-Light series of SDCMs from Crest Optics allow for changing the pinhole size and spacing by manually replacing the disks, however, this can't be performed during acquisition [47].

1.2.3. Structured Illumination Microscopy

The techniques discussed until now are not primarily deployed to increase the lateral resolution of an optical system, though both are capable of doing so. Structured Illumination Microscopy (SIM) is a blanket term for a group of techniques that seek to improve the lateral resolution of a system by utilising non-uniform illumination. In [48], Enderlein provides a simplified mathematical description of a structured illumination microscope that is applicable in the case of a thin, fluorescent sample placed in the focal plane of the microscope. An adaptation of which shall be used presently for discussions of structured illumination. Figure 1.3 shows the mathematical setup used in this analysis.

For the purposes of this discussion, the sample shall be placed in the $\xi\eta$-plane. The microscope shall transfer an image of the sample onto the xy-plane at which either a point or area detector shall be placed. The microscope will act with unity magnification for simplicity, though this need not be the case. In a fluorescence microscope, the signal of interest is the density of the fluorophores within the sample, which shall be denoted c. The system's illumination is provided via an optical system with PSF h_x and detection is performed via an optical system h_m (*NB*: h_x and h_m need not be the same. Even if the same optical system is used for excitation and emission, i.e., episcopic illumination, the presence of the Stokes shift will ensure they are at least slightly different). Fluorophores are assumed to respond linearly to excitation irradiance, i.e., if the excitation irradiance is given by $I_x(\xi, \eta)$ then the emission irradiance at the sample plane will be:

$$I_m(\xi, \eta) = I_x(\xi, \eta) c(\xi, \eta) \tag{1.14}$$

the exact method of generating the patterned illumination, I_x, is not specified, however, for simplicity of explanation, assume it is by incoherently imaging a patterned source through the illumination optics. In this case, the emission from the sample becomes:

$$I_m(\xi, \eta) = (P(\xi, \eta) * h_x(\xi, \eta)) c(\xi, \eta) \tag{1.15}$$

where P is the desired illumination pattern and $*$ denotes convolution. The fluorescent emission is imaged onto the detector by the microscope optics and as such the irradiance distribution at the detector is given by:

$$\begin{aligned} I_{det}(x, y) &= h_m(x, y) * I_m(x, y) \\ &= h_m(x, y) * [(P(x, y) * h_x(x, y)) c(x, y)] \end{aligned} \tag{1.16}$$

If P is chosen to correspond to even illumination, i.e., $P(x, y) = 1$, then the convolution within the brackets reduces to a constant term and the irradiance at the detector becomes

$$\begin{aligned} I_{det}(x, y) &= h_m(x, y) * c(x, y) \\ &= \iint_{-\infty}^{\infty} h_m(x - \xi, y - \eta) c(\xi, \eta) d\xi d\eta \end{aligned} \tag{1.17}$$

where a multiplicative constant has been omitted for simplicity. This is, as expected, the equation for the irradiance at the detector of a widefield fluorescence microscope, where a fluorophore emitting at position (ξ_0, η_0) would produce an image of the PSF of the system, h_m, shifted to the corresponding location in the xy-plane.

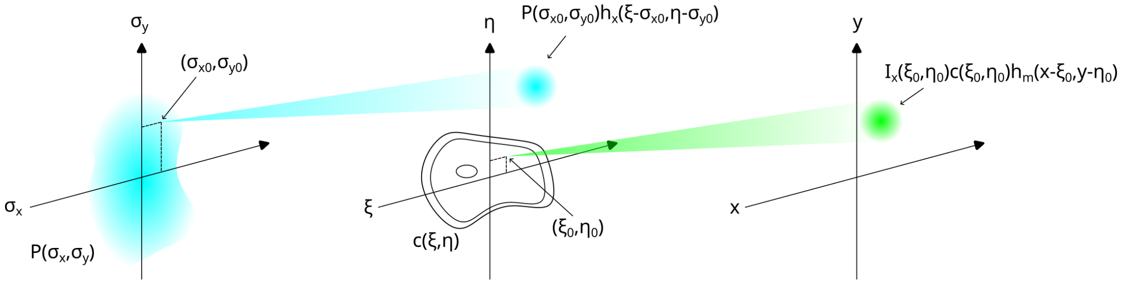

σ_y
$(\sigma_{x0},\sigma_{y0})$
σ_x
$P(\sigma_x,\sigma_y)$
η
$P(\sigma_{x0},\sigma_{y0})h_x(\xi-\sigma_{x0},\eta-\sigma_{y0})$
ξ
$c(\xi,\eta)$
(ξ_0,η_0)
y
$I_x(\xi_0,\eta_0)c(\xi_0,\eta_0)h_m(x-\xi_0,y-\eta_0)$
x

Figure 1.3.: A diagram of a hypothetical fluorescence microscope utilising structured illumination. The desired illumination pattern is formed in the source plane, (σ_x, σ_y), by an unspecified, incoherent, process. This pattern is imaged onto the sample plane, (ξ, η), by an optical system (not shown) with PSF h_x. An arbitrary point in the source, located at $(\sigma_{x0}, \sigma_{y0})$, is imaged onto the sample plane as an irradiance distribution given by the excitation PSF centred at $(\sigma_{x0}, \sigma_{y0})$ and scaled by the intensity of the source at that point in the source plane. The final excitation distribution at the sample is given by the sum of the contributions from all points in the source. A similar process occurs when the fluorophores emit and are imaged onto the detector plane, (x, y), by a second optical system (also not shown). Emission from fluorophores positioned at a point (ξ_0, η_0) is imaged onto a distribution in the detector plane given by the emission PSF, h_m, shifted to (ξ_0, η_0) and scaled by both the fluorophore density, c, and the excitation irradiance, I_x, at the point (ξ_0, η_0) in the sample plane. The final image at the detector plane is given by the sum of the contributions from each point in the sample plane. Both the excitation and emission optical systems are assumed to be unity magnification for simplicity.

Gustafsson SIM

SIM was first proposed by Gustafsson in 2000 [49], with the proposed variant operating similarly to an amplitude modulation transmission scheme and as such is best understood from the frequency domain. If I_x is chosen to be:

$$I_x(\xi, \eta) = 1 + \cos[2\pi f P_\theta(\xi, \eta) + \phi] \tag{1.18}$$

where f is the spatial frequency of the irradiance distribution in the $\xi\eta$-plane, $P_\theta = \xi \sin\theta + \eta \cos\theta$, θ is the angle of the sinusoidal pattern in the $\xi\eta$-plane and ϕ is the initial phase term. Inserting into Equation 1.14 gives:

$$I_m(\xi, \eta) = (1 + \cos[2\pi f P_\theta(\xi, \eta) + \phi]) * c(\xi, \eta) \tag{1.19}$$

Taking the Fourier transform of both sides gives

$$\begin{aligned} \tilde{I}_m(u, v) &= \left[\delta(u, v) + \frac{1}{2}\delta(u - u', v - v')e^{-j\phi} + \frac{1}{2}\delta(u + u', v + v')e^{j\phi}\right] \tilde{c}(u, v) \\ &= \tilde{c}(u, v) + \frac{1}{2}\tilde{c}(u - u', v - v')e^{-j\phi} + \frac{1}{2}\tilde{c}(u + u', v + v')e^{j\phi} \end{aligned} \tag{1.20}$$

where $u' = f \cos\theta$ and $v' = f \sin\theta$. The terms $\tilde{c}(u - u', v - v')$ and $\tilde{c}(u + u', v + v')$ represent frequency shifted versions of $\tilde{c}$. Shifting $\tilde{c}$ in the negative frequency direction will act to shift higher frequencies into the passband of the microscope, shifting in the positive direction will do similarly for higher negative frequencies. The resulting image will therefore consist of a mix of the low frequency terms and high frequency terms. By taking 3 images with differing phase terms, ϕ, a series of differential equations in $\tilde{c}$ can be formed, which when solved yield the values of $\tilde{c}$, $\tilde{c}(u - u', v - v')$ and $\tilde{c}(u + u', v + v')$ within the passband of the microscope. The two high frequency terms may then be shifted back to (u', v') and $(-u', -v')$ to produce a final image with increased resolution. As described here, it appears that the technique would be able to achieve arbitrarily high resolutions by selecting multiple values of f of increasingly high frequency in order to build up a map of the entire frequency domain. In practice this isn't possible as the pattern must illuminate the sample via the excitation optical system, which is diffraction limited. If f is chosen too large then the sinusoidal illumination won't reach the sample. The highest allowable value of f is determined by the MTF of the microscope, using such a value would result in a resolution enhancement of 2× the diffraction limit, in the sense that the support of the MTF would be extended by a factor of 2. This process will only improve the resolution along the θ direction, isotropic resolution enhancement repeating the process for multiple values of θ, with values of 0° and ±45° typically considered to give isotropic resolution enhancement. It is possible to use patterns consisting of the superposition of multiple sin waves with different θ values to generate isotropic resolution enhancement, however this requires more phases to collected (though still requiring fewer images than would otherwise be required) [50].

Image Scanning Microscopy

First described by Sheppard in 1988 [41] and implemented by Enderlein in 2010 [48], Image Scanning Microscopy (ISM) is a variant of SIM that uses confocal as the excitation pattern. It makes use of the fact that if a collimated laser fills the back aperture of an objective, the resulting illumination spot will contain all the frequencies transmittable by the microscope. Intuitively from the results of the previous section, if the illumination pattern contains frequency components up to f_{max}, then the resulting image will consist of a combination of multiple shifted versions of $\tilde{c}$ which, if they could be disentangled, would extend the support of the MTF by a factor of 2. The following discussion will describe the optical setup used by Enderlein in [48].

Consider a CLSM with the laser focused onto a position $(\xi_{scan}, \eta_{scan})$ in the sample. The excitation irradiance will be given by the excitation PSF, shifted to $(\xi_{scan}, \eta_{scan})$, therefore by Equation 1.15:

$$I_m(\xi, \eta) = h_x(\xi_{scan} - \xi, \eta_{scan} - \eta)c(\xi, \eta) \tag{1.21}$$

Note that the h_x has been flipped about the ξ and η axes to be consistent with the formulation by Enderlein [48], if h_x is radially symmetric then this has no effect. As before, an emitter at (ξ, η) will produce a PSF at the corresponding position on the sensor and therefore:

$$\begin{aligned} I_{det}(x, y; \xi_{scan}, \eta_{scan}) = \\ \iint_{-\infty}^{\infty} h_m(x - \xi, y - \eta)h_x(\xi_{scan} - \xi, \eta_{scan} - \eta)c(\xi, \eta)d\xi d\eta \end{aligned} \tag{1.22}$$

where the first convolution in Equation 1.16 has been written in its explicit integral form and I_{det} has been explicitly parametrised on the scan position. In the equation above, an off centre scan position will be correspondingly translated at the detector. In the Enderlein microscope, however, the emission is descanned by the galvanometers used for illumination, ensuring that the emission remains centred at the detector. This descanning is most easily expressed by introducing a shift term into the emission PSF, resulting in:

$$\begin{aligned}
&I_{det}(x, y; \xi_{scan}, \eta_{scan}) = \\
&\qquad \iint_{-\infty}^{\infty} h_m(\xi_{scan} - \xi + x, \eta_{scan} - \eta + y) h_x(\xi_{scan} - \xi, \eta_{scan} - \eta) c(\xi, \eta) d\xi d\eta
\end{aligned} \tag{1.23}$$

At this point it's interesting to consider the case of an ideal confocal microscope. As described by Sheppard [39], in such an ideal system, a point detector is placed at the origin in the detector plane and the intensity seen becomes the value of the image for that scan position. By Equation 1.23:

$$\begin{aligned}
I(\xi_{scan}, \eta_{scan}) &= I_{det}(0, 0; \xi_{scan}, \eta_{scan}) \\
&= \iint_{-\infty}^{\infty} h_m(\xi_{scan} - \xi, \eta_{scan} - \eta) h_x(\xi_{scan} - \xi, \eta_{scan} - \eta) c(\xi, \eta) d\xi d\eta \\
&= (h_m(\xi_{scan}, \eta_{scan}) h_x(\xi_{scan}, \eta_{scan})) * c(\xi_{scan}, \eta_{scan})
\end{aligned} \tag{1.24}$$

As remarked in [39], in this ideal case, the PSF of the system is seen to be the product of the excitation and emission PSFs, or equivalently in the frequency domain, the system's MTF is given by the convolution of the excitation and emission MTFs. The result of such a convolution would have support equal to the sum of the supports of the two MTFs, if the two MTFs are assumed to be approximately equal then the support of the MTF will be twice that of a traditional widefield microscope. An ideal confocal microscope therefore already achieves 2× the resolution of a conventional widefield microscope, as measured by the Abbe criterion. Such a system would require the pinhole to be infinitesimally small and as such would reject all light. In reality, the value of the resulting image at a given scan position must be the integral of I_{det} over some region of the detector plane. Sheppard considers this case in [39] and concludes that the larger the area integrated over, the lower the resolution achieved, with the limiting case being that of integration over the whole detector plane, in which case the system acts exactly as a widefield microscope when considering the lateral resolution. ISM may be considered as a way to recover the super-resolution functionality of the confocal system in the case of a pinhole of finite size.

In ISM, an image of I_{det} is captured for each scan position and off-axis detection is reassigned to position $\left(\xi_{scan} - \frac{x}{2}, \eta_{scan} - \frac{y}{2}\right)$ rather than contributing to the value at $(\xi_{scan}, \eta_{scan})$. The intuition for this is given in [41], the PSF associated with an off-axis point is given by the multiplication of the excitation PSF with a shifted version of the emission PSF. If the excitation and emission PSFs are identical and assumed to be approximately Gaussian, then the peak of this effective PSF will be at $\left(\frac{x}{2}, \frac{y}{2}\right)$ and assigning these photons to position $(\xi_{scan}, \eta_{scan})$ is therefore sub-optimal. Using Equation 1.24:

$$\begin{aligned}
I(\xi_{scan}, \eta_{scan}) &= \iint_{-\infty}^{\infty} I_{det}\left(x, y; \xi_{scan} - \frac{x}{2}, \eta_{scan} - \frac{y}{2}\right) dxdy \\
&= \iiiint_{-\infty}^{\infty} h_m\left(\xi_{scan} - \xi + \frac{x}{2}, \eta_{scan} - \eta + \frac{y}{2}\right) \times \\
&\qquad h_x\left(\xi_{scan} - \xi - \frac{x}{2}, \eta_{scan} - \eta - \frac{y}{2}\right) c(\xi, \eta) d\xi d\eta dxdy \\
&= \iint_{-\infty}^{\infty} c(\xi,\eta) \iint_{-\infty}^{\infty} h_m\left(\xi_{scan} - \xi + \frac{x}{2}, \eta_{scan} - \eta + \frac{y}{2}\right) \times \\
&\qquad h_x\left(\xi_{scan} - \xi - \frac{x}{2}, \eta_{scan} - \eta - \frac{y}{2}\right) dxdyd\xi d\eta \\
&= \iint_{\infty}^{\infty} c(\xi, \eta) h(\xi_{scan} - \xi, \eta_{scan} - \eta) d\xi d\eta \\
&= \qquad c(\xi_{scan}, \eta_{scan}) * h(\xi_{scan}, \eta_{scan})
\end{aligned} \tag{1.25}$$

where h is given by:

$$h(x', y') = \iint_{-\infty}^{\infty} h_m\left(x' + \frac{x}{2}, y' + \frac{y}{2}\right) h_x\left(x' - \frac{x}{2}, y' - \frac{y}{2}\right) dxdy \tag{1.26}$$

Using the substitutions $2v_x = x' + \frac{x}{2}$ and $2v_y = y' + \frac{y}{2}$, h becomes:

$$\begin{aligned}
h(x', y') &= \iint_{-\infty}^{\infty} h_m\left(2v_x, 2v_y\right) h_x\left(2x' - 2v_x, 2y' - 2v_y\right) dv_x dv_y \\
&= h_m\left(2x', 2y'\right) * h_x\left(2x', 2y'\right)
\end{aligned} \tag{1.27}$$

where a multiplicative constant has been dropped for simplicity. The PSF in a ISM microscope is therefore given by the convolution of the excitation and emission PSFs, sharpened by a factor of 2. The effects of this are most easily understood in the frequency domain where:

$$\tilde{h}(u, v) = \tilde{h}_m\left(\frac{u}{2}, \frac{v}{2}\right) \tilde{h}_x\left(\frac{u}{2}, \frac{v}{2}\right) \tag{1.28}$$

from which it can be seen that the support of the MTF has been extended by a factor of 2, assuming the two PSFs to be equal. It can also be seen that the MTF will roll off faster than the widefield equivalent MTF. If the PSFs are assumed to be Airy disks, the widefield MTF will roll off approximately linearly with frequency and the ISM PSF will therefore roll off quadratically. As shown in Figure 1.2, suppression of the high frequency components will reduce the resolution of the system as measured by the Sparrow criterion, making the apparent resolution enhancement less than the frequency domain explanation implies. Deconvolution is therefore used to restore the amplitudes of the higher frequency components [48], [51]. If the PSFs are assumed to be dissimilar then the support of the resulting MTF will be determined by the smaller of the supports of the excitation and emission MTFs.

Practically speaking, several implementation of ISM have been proposed. The Enderlein microscope [48] replaces the point detector in a traditional CLSM with a CCD camera. This results in an effective but slow implementation of ISM, the dwell time issues described in 1.2.2 are exacerbated by the slow readout times of CCD cameras. York *et al.* described a variant using multiple illumination regions generated by a Digital Micro-Mirror Device (DMD) (see Section 1.4) [52]. This speeds up ISM acquisitions in an analogous manner to how a SDCM improves upon CLSM imaging speeds. The system proposed in [52] has no physical pinholes, with ‘digital pinholing’ applied as part of the reconstruction algorithm to maintain the optical sectioning capabilities of confocal. A SDCM retrofitted with stroboscopic illumination is proposed in [53], which has the advantage of allowing conversion of any SDCM into a super-resolution system. Another paper by York *et al.* described an arrangement of microlenses that would allow pixel reassignment to be performed optically [51]. ISM has also been combined with various other techniques, for example [54] describes the incorporation of pixel reassignment into a confocal Raman Microscope in order to get sub-diffraction limited spectroscopic information.

Non-Linear SIM

As described above, the limit of resolution enhancement in SIM techniques is determined by the maximum spatial frequency that can be projected onto the sample. In principal, if a higher frequency could be projected onto the sample, then SIM techniques would be able to achieve a resolution enhancement greater than 2×. Non-linear SIM techniques attempt to exploit the non-linear response of fluorophores to changes in excitation irradiance under some conditions. For example, Saturated Pattern Excitation Microscopy (SPEM) and Saturated SIM (SSIM) [55], [56] both utilise fluorophore saturation to clip the peaks of the sin waves used in Gustafsson SIM. Mathematically, this can be understood as setting I_x in Equation 1.14 to be a clipped sin wave. A clipped sin wave is still periodic but will have an infinite number of frequency components, existing at multiples of the fundamental frequency, which is chosen to be transmissible by the illumination optics. This is possible only because the higher frequency terms are in effect 'injected' at the sample and therefore don't need to travel through an optical system. The utilisation of the non-linear process of stimulated emission on an ISM microscope has been reported [57], but not published in a peer reviewed journal.

1.2.4. Other Super-Resolution Techniques

The previous discussion has focused on SIM techniques because these will be of most practical importance to this thesis. There are, however, a plethora of other super-resolution techniques which are worth briefly mentioning.

Single Molecule Localisation Microscopy

In 2006, two closely related super-resolution techniques were developed, Photoactivation Localisation Microscopy (PALM) and Stochastic Optical Reconstruction Microscopy (STORM). In both techniques, the majority of the fluorophores in the sample are held in a state where they do not fluoresce [58]. Random fluorophores will switch into an emitting state for short periods of time before returning to a dark state. The average time a fluorophore remains in the dark state is controlled to be much longer than the average time a fluorophore emits for. This means that the probability of two PSFs generated by two fluorophores overlapping is low. The resulting PSFs can be analysed individually to determine the most likely location of the fluorophores that produced them to sub-pixel accuracy and typically to a resolution of $\sim$20 nm [59]. In terms of Equation 1.16, Single Molecule Localisation Microscopy (SMLM) techniques can be thought of as manipulating the c term rather than attempting to manipulate the effective PSF of the system. Many variations on the SMLM premise have been developed (e.g., dSTORM [60], DNA-PAINT [61], etc.), all utilising different physical properties to produce the blinking effect.

In 2017, Hell published a paper describing a method to further improve the resolution of SMLM, MINFLUX [62]. The technique works by utilising a toroidal excitation beam with a position controlled by Field Programmable Gate Array (FPGA) which acts to minimise the detected signal from a blinking fluorophore by keeping it at the centre of the toroid where the illumination irradiance is zero. This reduces the effects of limited photon budget that reduces the maximum achievable resolution in traditional SMLM techniques by minimising the number of photons required to accurately determine the position of the fluorophore.

Reversible Saturable Optical Fluorescence Transitions

In 1994, Hell and Wichmann proposed a method which utilised stimulated emission of fluorophores to temporarily disable selected emitters [63]. Stimulated Emission Depletion (STED) microscopy uses a scanning fluorescence laser beam, surrounded by a toroidal depletion beam. Due to the diffraction limit of the illumination optics, both the fluorescence and depletion beams will blur and overlap in the sample. The power of the depletion beam is set to the point that it will inhibit fluorescence at the detection wavelengths in the region of overlap of the excitation and depletion beams. In effect, this reduces the size of the excitation PSF in Equation 1.24, with the final size determined by the intensity of the depletion laser. Since the PSF of an ideal confocal system is given by the product of the excitation and emission PSFs, it's clear that a significantly contracted excitation PSF will result in dramatic resolution enhancement. A year later, Hell and Kroug published another technique, Ground State Depletion (GSD), that works on a similar principal [64]. GSD uses the depletion beam to force the fluorophores into a triplet state, rather than relying on stimulated emission, but otherwise the two techniques are the same. The drawback of both STED and GSD is that the high intensity depletion lasers have the tendency to cause phototoxicity, limiting their practical use in biological imaging. In 2005, Hell and Hofmann presented yet another technique, this time using photo-switchable proteins that could be enabled and disabled with application of appropriate wavelengths of light [65]. This technique requires significantly lower light intensities than STED and GSD. Together, these form a group of super-resolution techniques known as Reversible Saturable Optical Fluorescence Transitions (RESOLFT).

1.3. Spectroscopy

The techniques described thus far have been primarily concerned with the determining the spatial distribution of fluorophores within the sample. Spectroscopy is a complementary class of techniques in which properties of the emitted light (e.g., constituent wavelengths) are measured instead. What follows is a brief description of the spectroscopic techniques discussed in this thesis.

1.3.1. Förster Resonance Energy Transfer

The process of Förster Resonance Energy Transfer (FRET) was first observed in 1922 [66] and a full quantum mechanical description was developed by Förster in 1948 [67]. The basic principal is that when two suitably chosen fluorophores are brought sufficiently close together with one fluorophore, the donor, in an excited state, the excitation is transferred to the second fluorophore, the acceptor. Importantly, the transfer of excitation is not via emission from the donor followed by absorption by the acceptor, which in principal could happen over any distance. Instead it is caused by a resonance effect that is only present over short distances. The FRET efficiency is given by:

$$E = \frac{k_{ET}}{k_{ET} + k_f + k_i} \tag{1.29}$$

where k_{ET} is the rate of energy transfer, k_f is the rate of fluorescence of the donor molecule and k_i is the rate of non-fluorescent decay. From the quantum mechanical description it can additionally be seen that:

$$E = \frac{1}{1 + \left(\frac{r}{R_0}\right)^6} \tag{1.30}$$

where r is the distance between the two fluorescent molecules and R_0 is the Förster radius, defined as the distance between the two molecules which results in a 50% FRET efficiency. The definition of R_0 depends on several parameters of the two fluorescent molecules involved as well as a term, κ^2, that is related to the relative orientation of the fluorophore's dipoles. The strong dependence of R_0 on the relative orientation of the fluorophores causes problems for experiments seeking information about the distance between dyes, so typically experimental conditions are arranged such that the value of κ^2 can be treated as a known constant. Practically, this usually means ensuring that the fluorophores may rotate freely with respect to one another, in which case the time averaged value of $\kappa^2 = \frac{2}{3}$ may be used. Many studies have sought to to determine if and when this assumption is valid in practical settings [68]–[72].

Once the κ^2 problem is dealt with, measurements of changes to FRET efficiency may be used to infer changes in distance between the fluorophores. Considering again Equation 1.30, it can be seen that FRET efficiency varies sigmoidally with distance, with a linear region spanning from $r \approx 0.6R_0$ to $r \approx 1.4R_0$, typically providing sensitivity to differences in distance over the approximate range 1 nm-10 nm [73]–[76].

As an example of how this may be deployed, consider the work in [77]. It was known that a leaky ryanodine receptor could play a role in sudden cardiac death, though the process by which this happened was unknown. It was theorised that it was due to the inappropriate ‘unzipping’ of two domains within the receptor. To investigate this, a FRET pair, consisting of Yellow Fluorescent Protein (YFP) and Cyan Fluorescent Protein (CFP), was transfected into HEK293 immortalized human embryonic kidney cells such that one of the proteins would be expressed on one domain of interest and the other protein on the other. If unzipping was present, then a reduction in FRET efficiency would be measured with respect to the control. Such a change was indeed measured and the experimenters were even able to verify that the addition of drugs known to target leaky ryanodine receptors increased the FRET efficiencies once again.

In the experiment described above, many FRET pairs were imaged simultaneously. While this gives a measure of average movements, the experimenters would have been blind to changes in subpopulations; some FRET pairs could have moved together when the majority moved apart and the data would appear unchanged. A subclass of FRET techniques, known as Single Molecule FRET (smFRET) or sometimes more accurately Single Pair FRET (spFRET) [78], attempts to remove the ambiguity associated with ensemble measurements by measuring from single FRET pairs at a time. Two main methods for smFRET experiments are Total Internal Reflectance Fluorescence (TIRF) microscopy [79] and confocal [80].

In the TIRF variant, an evanescent field, which only penetrates 100 nm into the sample [81] at appreciable intensity, is used for excitation of the fluorophores. A standard widefield microscope can then be used to collect the emission from the entire field close to the cover slip. Since molecules (with attached FRET pairs) left in solution would spend insufficient time in the evanescent field to emit enough photons for reliable FRET efficiency determination, the molecules must be attached to the surface of the cover slip. Being attached to the surface of the cover slip may have effects on the dynamics of the molecule under investigation [82], however, techniques have been proposed to mitigate this [83].

On the other hand, Confocal smFRET utilises a confocal microscope with a fixed illumination volume. The molecules are allowed to diffuse randomly in solution and a sufficiently low concentration (of the order of tens of picomolar) is used to ensure only a single fret pair is present in the detection volume at once [84]. High-performance APDs are often used to allow single photon timestamping to a resolution of 10ns when paired with a counter based TDC (e.g., a NIDAQ) or of the order of picoseconds when using higher performance TDCs (e.g., PicoQuant’s HydraHarp). This allows analysis of faster subpopulation dynamics than is achievable with TIRF setups. The drawback is that each molecule is only within the confocal volume for a short time, of the order of milliseconds.

This means that long term dynamics of individual molecules can't be measured, shot noise is a limiting factor in the accuracy of analysis of individual molecules, and it typically takes hours to collect enough measurements for reliable analysis. Solutions have been proposed to this limitation, for example, a device is described in [85] that utilises electrodes to 'kick' charged molecules back to the centre of the confocal volume as they diffuse away.

smFRET measurements performed entirely with donor excitation suffer from the problem of not being able to tell the difference between a doubly labelled molecule where the fluorophores are far apart and a molecule where the acceptor labelling was unsuccessful. A solution to this is to periodically excite the acceptor directly in order to determine if the acceptor fluorophore is present, an illumination scheme referred to as Alternating Laser EXcitation (ALEX) [73]. When using ALEX on a confocal smFRET microscope, measurements of FRET efficiency, E, and stoichiometry, S, may be determined using two equations:

$$E = \frac{N_{DA}}{N_{DD} + N_{DA}} \tag{1.31}$$

$$S = \frac{N_{DD} + N_{DA}}{N_{DD} + N_{DA} + N_{AA}} \tag{1.32}$$

where N_{DD}, N_{DA}, and N_{AA} are the number of donor photons detected under donor excitation, acceptor photons detected under donor excitation, and acceptor photons detected under acceptor excitation respectively. This extra information can be used to correct inaccuracies in the data by ensuring that all donor only molecules appear with a stoichiometry of 1 and efficiency of 0, all acceptor only molecules appear with a stoichiometry of zero, and all doubly labelled molecules appear with a stoichiometry of 0.5. Once this correction has been performed, it can be used in the calculation of absolute distances [86].

1.3.2. Fluorescence Correlation Spectroscopy

Fluorescence Correlation Spectroscopy (FCS) was developed in the 1970s [87] to study the kinetics of chemical reactions in steady state, though has since been used to investigate many phenomena in the life sciences [88]. It relies on the fact that even in steady state, a system is not static, and will continuously fluctuate around the equilibrium. By calculating the autocorrelation of the fluorescence signal, several parameters of interest may be determined, including but not limited to diffusion coefficients and the number of molecules within an illumination volume. Several variants of the technique exist, for example Fluorescence Cross-Correlation Spectroscopy (FCCS) where two fluorescent molecules are used and the correlation between their emissions can be used to determine if their diffusions are correlated, anti-correlated or not correlated [89].

1.3.3. Fluorescence Lifetime

As mentioned in 1.1.4, fluorophores remain in an excited state for a short period of time before relaxing to the ground state and emitting a fluorophore. If a brief pulse of excitation light is used to excite fluorophores in a sample and the time between the excitation and the emission of fluorophores is measured to a suitable precision [90], then the average time a fluorophore spends in the excited state can be determined. The average time spent in the excited state is called the fluorescence lifetime [91]. The fluorescence lifetime is determined not just by the fluorophore but also by the local environment surrounding the fluorophore, allowing for accurate and localised measurement of several parameters, including but not limited to: viscosity [92], temperature [93], pH [94], and ion concentration [95]–[97]. Often, fluorescence lifetime measurements are implemented using CLSMs, utilising the scanning functionality of such a system, an image of spatial variations in fluorescence lifetimes may be built up, in which case the technique is referred to as Fluorescence Lifetime Imaging (FLIM). FLIM has also been implemented on TIRF systems [98] as well as being combined with techniques such as Optical Projection Tomography (OPT) to build 3D maps of fluorescence lifetime within a sample [99]. Recent developments in APD array technology are paving the way for cameras that are capable of directly performing FLIM, without the need for point scanning [15], [16].

1.4. Digital Micromirror Devices

The majority of the work presented in this thesis concerns a microscope with an integrated Digital Micro-Mirror Device (DMD) (see Figure 1.4). A DMD is a Microelectromechanical System (MEMS) consisting of many microscopic square mirrors that are each of the order of 10 µm in size [100]. The mirrors are arranged in a rectangular array and each is free to rock on a hinge mounted along one of its diagonals. Two electrodes are placed underneath the corners of the DMD that are free to move. The DMD is held at a constant bias voltage with the same voltage applied to one of the electrodes, the complementary voltage being applied to the other. The electrostatic forces associated with this configuration will push one corner up and pull the other down, holding the mirror in one of two positions determined by the voltages on each electrode. Nominally, the two mirror positions will be ±12° from the horizontal, however, this will vary slightly due to manufacturing tolerances.

Typically, DMDs are illuminated from a 24° angle to the horizontal (the Green-Red plane in Figure 1.4). From a geometrical optics perspective, that ensures that the illumination light striking a pixel oriented towards the illumination source is steered perpendicular to the DMD, whereas light striking the DMD on a mirror in the opposite orientation will be deflected elsewhere. If an optical system is placed such that it accepts light only light travelling perpendicular to the DMD, the DMD can be used to control the input distribution to the optical system. If the optical system forms an image of its input distribution, then the DMD can be used to project patterns onto that location. In addition to the binary scheme described here, DMDs can often pulse-width modulate the mirror positions in order to reduce the average intensity to particular locations in the projected image, allowing for almost continuously varying illumination intensities within the image, when considering sufficiently long exposure times.

While the operation described above seems simple, unfortunately it doesn't take into account diffractive effects. The DMD acts as a double-ruled, blazed, diffraction grating [101], producing a complex 2D diffraction pattern when illuminated. Significant energy will be present in the higher orders of the diffraction pattern due to the blazed grating effect [1, p.499]. Optical designs must take care to capture enough of these diffracted orders while also making sure that any non-captured orders are blocked appropriately. Additionally, when attempting to use a DMD in the reverse direction, it will introduce a 24° 'tilt' into the image which distorts the PSF and must be corrected out [102].

1.4.1. Uses in Microscopy

Despite the optical difficulties presented by DMDs, they have been successfully employed in several microscopes to provide patterned illumination. As mentioned in Section 1.2.3, a DMD was used to provide the illumination pattern in the ISM implementation presented by York *et al.* [52]. DMDs have also been used to generate the sinusoidal patterns used in Gustafsson SIM implementations [103], [104].

In this thesis, the DMD is used as part of a microscope addon called the CairnFocal (see Section 2.1.1). This addon converts a widefield microscope into what has been previously described in the literature as a Programmable Array Microscope (PAM). Initially designed by the Jovin group at the Max Plank Institute for Biophysical Chemistry in the 1990s, PAMs use a Spatial Light Modulator (SLM) to project arbitrary illumination patterns onto the sample [105]–[108], similar to the other uses of DMDs described. Unlike the other techniques, however, the SLM is often placed in the emission path as well as

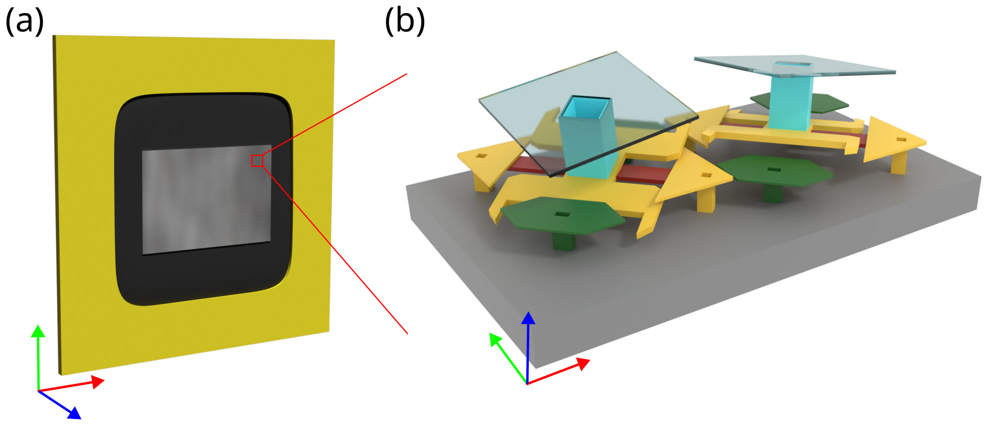


Figure 1.4.: Rendering of a DMD chip, (a), and two individual micro-mirrors, (b). The relative orientation of (a) and (b) is indicated by the three coloured arrows to the left of each of the images. The micro-mirrors are free to rotate on a hinge (shown in red) by 12° in either direction. The electrodes that control the position of the mirrors are shown in green.

the excitation path, allowing filtering of the signals collected by the microscope. Either a DMD or a Liquid Crystal Display (LCD) can be used as the SLM [109]. By varying the pattern displayed on the SLM, various properties of the imaging can be controlled, e.g., the imaging modality. When using a DMD as a detector, the pattern displayed can be used to control both the illumination pattern and to direct light between one of two detectors. In [106], the theory behind the use of a DMD PAM for performing confocal was described. The paper outlined two methods, one for emulating a traditional SDCM and one for using pseudorandom patterns with post processing to extract the confocal image, however, difficulties with the correction of the 24° tilt and the DMDs diffractive effects reduced the effectiveness of any implementations. An improved optical design, utilising a Schiefspiegler mirror system was later proposed, patented by Cairn Research [102] and incorporated into the CairnFocal. A similar system was used in [98] as a way to improve the speed of FLIM. On one side of the DMD a camera was positioned in order to continuously image the sample in widefield. The DMD was used to selectively direct Regions of Interest (ROIs) to a high speed point detector to perform lifetime measurements. This system differed significantly from the one discussed in this thesis by only using the DMD in the emission pathway.

1.5. Software

As microscopy has evolved, the techniques employed have become more complex and so too have the demands placed on software for both microscope control and data analysis. In a microscopy system designed primarily to perform widefield, where the focus is set by an operator looking down the oculars by manually turning a focus knob on the microscope frame, data acquisition consists of a snapping singular images or taking a time lapse, and data analysis consists of tweaking the brightness and contrast, it is possible for this to be handled entirely from a single piece of proprietary software, likely produced by the manufacturer of the system's camera. Modern microscopy techniques typically rely on control of a large number of different automated components, may produce large amounts of high dimensional data [99], [110], [111] and/or rely on significant amounts of complex post processing [49], [51], [99], [112].

While some companies, such as Zeiss and Leica, still produce turn-key systems with a single piece of software for control and data analysis, the development of such systems tends to be slow, taking years or even decades for systems to make it into the hands of biologists. For the rapid adoption of new microscopy techniques in the life sciences, as well as detailed descriptions of the assembly of the novel microscopes, robust and freely available software must be available to support them. The projects described in this thesis involved a large amount of software development. What follows is a brief review of the advantages and disadvantages of various programming languages used in the projects described in this thesis and other software tools of particular interest to microscopy.

1.5.1. Programming Language Comparison

While most modern programming languages are 'Turing Complete' and therefore capable of performing any computation given sufficient time and resources, they exhibit striking variability in the ease at which those computations can be expressed. Furthermore, the existence of various language adjacent resources, such as software development tools (e.g., Integrated Development Environments (IDEs), package managers, debuggers, etc.) and third party libraries has a major impact on what is realistically achievable with a given language. The consequence of this is that a given programming language is typically only really appropriate when used for tasks falling into a particular niche.

C/C++

C is a systems programming language designed at Bell Labs in the 1970s [113]. It was initially designed to allow the UNIX operating system to be compiled for multiple CPU architectures and as such is geared towards simplicity and speed. Being designed for this purpose, C is still used in all modern operating systems and is similarly popular in many situations where low level communication with external devices is required. Memory management in C must be performed manually, this gives the programmer explicit control of memory allocations and deallocations but introduces the risk of memory leaks and memory access errors. Unlike most other languages, C has a simplistic and specified Application Binary Interface (ABI), leading to regular use in interfaces allowing communication between programming languages. Since most other programming languages come with some method to interact with C code, it is also regularly used as the Software Development Kit (SDK) language for device manufacturers, allowing them to maintain a single library that can be accessed from most user programs.

C++ was initially developed in the 1980s as an extension to C, adding language features to support Object-Oriented Programming (OOP), compile time polymorphism and enhanced type safety [114], [115]. While not a strict superset of C, C++ maintains excellent backwards compatibility and interoperability with C, leading to widespread adoption in both systems and applications programming. C++ also introduced the concept of Resource Acquisition is Initialisation (RAII), whereby resource allocations (e.g., memory) are wrapped in objects whose lifetime can be tracked by the compiler and automatically deallocated, removing the need for the error prone manual memory management of C, while still allowing fine grained memory management when needed. The age of C and C++ mean there are third party libraries available for a large number of common tasks.

Rust

Developed in the 2010s at Mozilla, Rust is a systems and applications programming language that aims to remove the memory and thread safety bugs common to C and C++ programs, while still maintaining the performance associated with those languages [116]. Additionally, Rust includes integrated tooling for automated testing and a package manager, Cargo. Rust utilises RAII and additional compiler checks to strictly enforce memory safety, preventing programs from compiling if potential bugs exist. Unfortunately, these checks are often overzealous, with requirements impacting performance and reuse of code. To counteract this, Rust allows developers to disable the compiler checks by marking code as unsafe. As noted in [117], the interaction between ‘safe’ and ‘unsafe’ code often behaves in unexpected ways, resulting in the sorts of bugs Rust claims to remove. Since the interaction with C code is considered unsafe by the Rust compiler and it is unclear if the majority of Rust’s safety features would provide any benefits over C/C++ for the use in microscope control without device manufacturers providing explicit support in their device SDKs.

One feature of Rust that could provide benefits is its compile time verification of multi-threading logic, which prevents common multi-threading bugs such as data races. A data race occurs when two or more threads attempt to access the same object or region in memory simultaneously and at least one of the threads is writing to the shared memory. Since the order of operations between threads is not guaranteed, the reads and writes of the various threads may occur in any order and the behaviour of the program is determined by parameters out of control of the programmer. Additionally, since the operating system will schedule the execution of threads differently depending on the state of other programs currently running on the computer, multi-threading logic mistakes may not always manifest as incorrect behaviour, meaning they are often not discovered until long after the bug is introduced into the code. In the worst case scenario, changes in optimisation and scheduling when debugging the program may result in different behaviour, a situation known as a 'Heisenbug', where attempting to observe the bug results in its effects changing or disappearing entirely. Rust's ability to shift detection of these sorts of bugs from runtime to compile time has the potential to significantly ease the writing of multi-threaded code.

Java and the JVM

Java is an object-oriented applications programming language developed in the 1990s by Sun Microsystems [118]. The language was designed to support a 'write once run everywhere' approach and to that end, Java code is not compiled to machine code (as in the case of the previously discussed languages). Instead it is compiled to an intermediate 'bytecode' that is translated into machine code at runtime by the Java Virtual Machine (JVM). While the Java Development Kit (JDK) doesn't provide a package manager, several third party options exit to provide this functionality (e.g., Maven [119], Gradle [120], Ant [121], etc.). In older versions of the JDK, interaction with C libraries was done through the Java Native Interface (JNI) libraries, which required the programmer to manually write implementations of Java classes in C that interacted with the C library and the JVM, which unsurprisingly is time consuming and error prone. Newer versions of the JVM provide support for automated C interoperability [122], however, due to large language and licensing changes between JDK 8 and 9, many projects are yet to update and are unable to make use of these features.

Since Java compiles to JVM bytecode rather than machine code, a plethora of languages have been developed that also target the JVM and strive for interoperability with Java code. One such language is Kotlin, developed by JetBrains and released in

2011 [123]. Kotlin was, in effect, a redesign of the Java language from the ground up, avoiding many of the mistakes in Java's design that became apparent in the years after its release. Kotlin maintains 100% interoperability with Java and is able to target JVM versions as old as version 6, making it a good choice to allow a modern programming experience from within projects utilising older JVM versions.

The JVM utilises a high performance Just-in-Time (JIT) compiler to perform compilation from bytecode to machine code at runtime, which is capable of performing optimisation at runtime, utilising statistical information about which code paths are actually taken at runtime that aren't available to languages like C/C++ [124]. In practice JVM languages are still typically outperformed by C/C++ [125], which may cause issues for high-speed microscope control applications. JVM languages are garbage collected and provide no methods for programmers to control when memory is deallocated. The garbage collector must interrupt the execution of the Java program while it cleans up memory that is no longer needed and since the JVM may choose to execute the garbage collector at any time, this can lead to unpredictable performance some scenarios. This is a particular issue when large amounts of memory is allocated as the garbage collector will run more frequently and for longer. It is often worth offloading such tasks to a language like C, C++ or Rust with deterministic memory management.

MATLAB

MATLAB is a proprietary language, developed by MathWorks in the 1970s, primarily for numerical computation and data analysis [126]. It provides a wide array of inbuilt functionality applicable to many domains of mathematics, engineering and sciences. MATLAB's primary use is for data analysis, for which many academics prefer the use of free alternatives such as Python. In the author's view, the MATLAB and Python should only be used for rapidly prototyping analysis algorithms. Once the algorithm is settled on, a robust implementation should be provided in user friendly package, implemented in a language tailored towards application development, preferably as part of an open source analysis package such as ImageJ. Short scripts may be provided as 'executable pseudocode' as part of publications, in which case a MATLAB script should be sufficiently short that it will run in the open source implementation, Octave [127], and the choice between MATLAB and Python becomes moot.

1.5.2. ImageJ and µManager

Developed in the early 2000s, ImageJ is an open source image analysis suite, written in Java [128]. ImageJ has since become a *de facto* standard in the biological image analysis community. It provides a Graphical User Interface (GUI) driven experience that allows users to apply complex image processing and analysis methodologies without any programming experience. It also provides a comprehensive macro system that allows users to record a series of operations and reapply them later, enabling high throughput batch processing with relatively little human interaction. It also provides a plugin framework, allowing programmers to implement algorithms in any JVM language to expose new functionality to end users. A large number of plugins have been created by the community, extending ImageJ's functionality significantly [129]–[134]

ImageJ provides a large amount of functionality geared towards image analysis, but no functionality for image capture. µManager provides image capture functionality to ImageJ [135] aiming to provide a unified interface for control of a wide variety of microscope components from different manufacturers. Support for devices is provided by the implementation of 'device adapters', Dynamic Link Libraries (DLLs) that are loaded at runtime based on the components present in the system. Device adapters are loaded and communicated with via the 'MMCore' library, which provides synchronisation between the various components in the system. Though MMCore can be used as a standalone library, the µManager project implemented a Java based GUI as an ImageJ plugin, that interacts with MMCore through an automatically generated JNI wrapper. This GUI provides an interface for end users to specify the components in their microscope, interact with cameras in both single shot ('snap') and video modes, move xy-stages, change positions of filter-wheels and objective turrets and perform 'multi-dimensional acquisitions', i.e., automatically controlling the channel, z-stage position and xy stage position, as well as allowing video acquisitions. On top of this built in functionality, µManager provides a plugin interface, similar to that of ImageJ, that allows programmers to write Java programs that are capable of controlling any devices connected to the µManager instance. Though these GUI elements and plugins are written in Java, device control is still via the wrapped MMCore library which is written in C++. This allows the Java code to instruct devices to perform actions while ensuring that the execution of those instructions is performed outside the JVM, avoiding the potential performance issues described in section 1.5.1.

In the work presented in this thesis, plugins for both ImageJ and µManager were developed. In both cases, the plugins were developed in Kotlin, rather than Java. This is primarily due to ImageJ (and by extension µManager) using JDK 8. Kotlin provides a significantly more features than Java 8 and this streamlined the development process significantly. The 100% Java interoperability guaranteed by Kotlin means there are no disadvantages to mixing Kotlin in a Java project in this manner.

1.5.3. Other Programming Languages of Note

Python

Initially released in 1991, Python [136] has gained popularity in the academic community in recent years. Python uses a simplistic syntax designed to be easier to read than languages like C or Java. Its dynamic type system means that programmers don't need to specify the type of a variable before it can be used, and the type can be reassigned during program acquisition. Additionally, Python is an interpreted language meaning no separate compilation step is required to run programs. These features have made Python popular with educators and new programmers alike and resulted in numerous 3rd party libraries to perform a wide range of different tasks. Of particular note are the numpy [137] and scipy [138] libraries which provide optimised implementations of n-dimensional array manipulation algorithms and common statistical/data analysis algorithms respectively.

While these features make Python quick to learn and prototype algorithms in, they also pose difficulties for maintaining larger software projects. Since Python code isn't compiled, any end user of a Python program must have a compatible Python interpreter and copies of all required libraries installed. Package managers such as Pip and Anaconda have attempted to make it easier to install the appropriate versions of the interpreter and libraries, however, this is still a complex and involved process for end users that may not know anything about Python. From the programmer's perspective, the dynamic type system may provide flexibility in some limited cases, it comes at the cost of reducing the amount of information available to the computer when the program is being written. This reduces the ability for development tools to provide features such as code completion, code suggestions and on-the-fly error detection. Additionally, it introduces numerous classes of type errors into programs that would be caught at compile time in statically typed, compiled programming languages. Ensuring that these errors are not present requires large automated testing suites that take significant time to develop.

LabVIEW

LabVIEW is a graphical programming environment developed by National Instruments that has been popular in the microscopy community. It provides a drag-and-drop interface for controlling supported devices and producing basic GUIs. While a simple option for small one off projects, LabVIEW quickly becomes unmanageable as code complexity increases; the use of proprietary binary code files prevents the use of modern source control with LabVIEW projects; the restrictive, paid for licenses required to develop LabVIEW code inhibits collaboration; and the relatively small number of devices available with a LabVIEW Application Programming Interface (API) (in comparison to those with a C/C++ API) means it is likely that a system with several devices will not be controllable entirely using LabVIEW without writing LabVIEW to C interface code. These issues make LabVIEW a poor choice for most modern research projects.

1.6. Project Aims

The aim of the project discussed in Chapters 2-4 was to develop a DMD based confocal microscopy platform, the CairnFocal, to allow it to perform multi-modality acquisitions. Chapter 2 discusses the hardware and software developments that form the basis of the future chapters. The implementation of several microscopy techniques on the CairnFocal platform is described in Chapter 3. Finally, Chapter 4 details the application of these techniques to biological systems. Chapter 5 concerns a secondary project to produce software to support the smfBox, confocal smFRET microscope.

CHAPTER

TWO

DEVELOPMENT OF THE CAIRNFOCAL PLATFORM

2.1. Introduction

This chapter outlines the software and hardware platforms that underpin the data collection described in subsequent chapters. The development work focused on four main areas: synchronisation hardware, µManager device adapters, a µManager plugin for data acquisition, and a control program for uploading patterns to the DMD.

2.1.1. Optical Design of the CairnFocal

The CairnFocal is a DMD based microscope attachment that provides the illumination and detection optics necessary to convert an epi-fluorescent widefield microscope into a confocal system [102]. While DMDs are somewhat common components in optical microscopy, they are typically only used to generate patterned illumination [52], [104]. The CairnFocal differs in this respect by using the DMD in both the illumination and detection paths of the microscope, as shown in Figure 2.1. In the illumination path, light striking an '*on*' pixel is directed towards the sample, whereas light striking an '*off*' pixel is prevented from reaching the sample. The DMD is placed in a conjugate image plane of the microscope, such that a demagnified image of the *on* pixels appears in the sample. Light collected by the objective is similarly focused onto the DMD and is directed to either the *on* or *off* side cameras depending on whether it hits a pixel in the *on* or *off* state. The left and right light paths are symmetrical and which of them corresponds to the *on* or *off* path is determined by which is used for illumination. Aside from diffractive effects, the primary difficulty with using the DMD in this configuration is the angle introduced into the image of the sample as it passes through the DMD. To counteract this effect, two curved mirrors, one concave and one convex, are arranged in a Schiefspiegler Relay to correct this angle.

Originally designed for use in mirror telescopes, the Schiefspiegler Relay [139] consists of two curved mirrors, with the secondary mirror placed off the optical axis such that it does not block the incoming light. When used in a telescope, the tilt of the two curved mirrors introduces significant astigmatism and coma aberration that must be carefully corrected. In the CairnFocal, these aberrations are deliberately introduced in order to cancel out the equivalent aberrations introduced by the DMD.

FPGA Synchronisation

In most imaging modalities achievable with the CairnFocal platform, synchronisation between the DMD and the other electronic components in the system is of paramount importance; in some cases, two different modalities are achievable with the same patterns and are differentiated only by the manner in which the system is synchronised. A discussion of the specific synchronisation patterns utilised in this project is given in Section 2.3.1.

The solution commonly employed to synchronise elements of a complex microscopy system is to use a data acquisition card (e.g., the NIDAQ range from National Instruments) [140], [141]. Such data acquisition cards typically provide several digital outputs that can produce arbitrary digital waveforms suitable for synchronising common pieces of lab equipment. Unfortunately, they come with several drawbacks. First, they are often only available with other features (e.g., digital inputs, analogue I/O, high speed counters, etc.). These extra features often go unused but increase the cost of the device and complexity of the control API. Secondly, the control flow is usually unidirectional; digital outputs can be used to instruct devices to perform tasks, but the data acquisition card cannot respond in hardware to signals from devices (e.g., trigger ready signals or expose out signals from cameras). It is often possible to have the device react to signals by reading an input in software and respond by changing the waveform applied to the outputs, however, this suffers from large overheads associated with transmission between the data acquisition card and computer, making it unsuitable for situations where a fast response is required (<1 ms). One final problem with the National Instruments data acquisition boards in particular, is that the preferred interface method is via LabVIEW. The difficulties that LabVIEW introduces are discussed in Section 1.5.3. While the NIDAQ boards do come with a C/C++ API, it is complex and poorly documented.

A low-cost solution commonly employed in place of a data acquisition card is an Arduino with custom firmware [135], [142], [143]. This solves many of the issues described

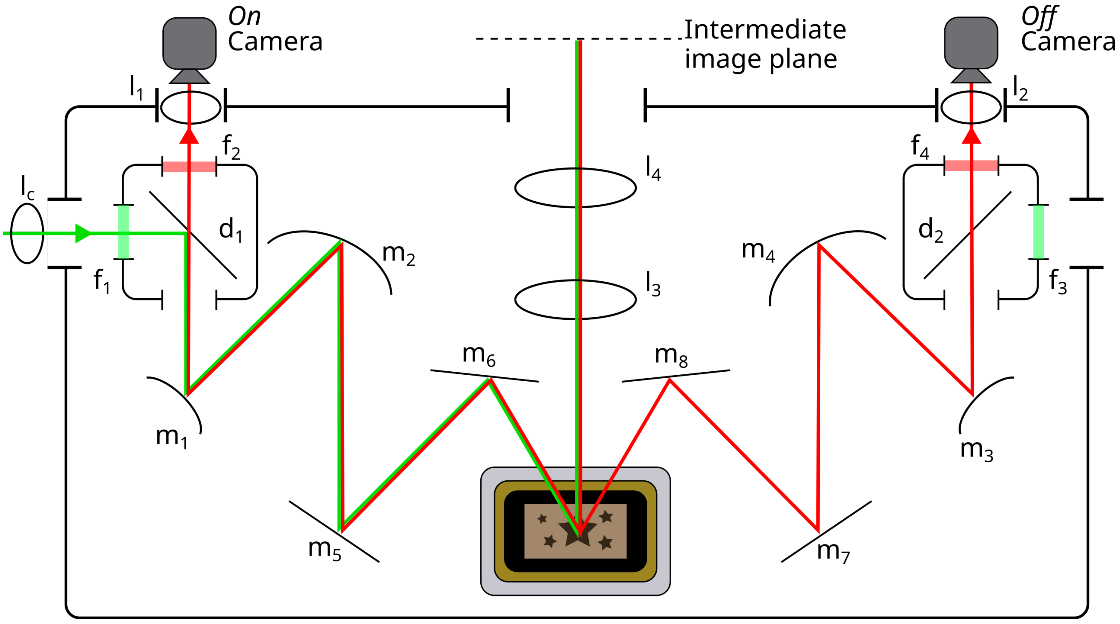


Figure 2.1.: Layout of the CairnFocal. The illumination path is shown in green, the detection path in red. Laser illumination is filtered by excitation filters f_1/f_3 and reflected by dichroics d_1/d_2 through mirrors $m_{1,2,5,6}/m_{3,4,7,8}$ onto the DMD. Lenses l_3 and l_4 relay a 1:1 image of the DMD *on* pixels onto the intermediate image plane of the microscope, where it is demagnified through the tube and objective lenses onto the sample. The light striking the off pixels will miss l_3 and therefore not make it to the sample.
In the reverse direction, l_4 and l_3 will relay the intermediate image onto the DMD. Regions of the image falling on *on* pixels will be directed to along the *on* light path (left in diagram) and parts falling on *off* pixels will be directed along the path opposite (right in diagram). In the *on* direction, mirrors m_1 and m_2 form a Schiefspiegler Relay that corrects out the angle introduced to the image as it passes through the DMD, at the same time acting as the first 'lens' in the 1:1 relay onto the camera, for which the second lens is l_1. Components m_3, m_4 and l_2 provide the equivalent function on the *off* side. Filters f_2 and f_4 are the emission filters required to detect fluorescence. l_c is the lens used to collimate the light from the illumination source before it enters the curved mirror system.

above; price is low and complexity can be added after the fact via widely available 'shield' boards, bidirectional communication can be achieved by writing appropriate firmware, and firmware is written in C++ with a comprehensive and well documented Board Support Package (BSP), while Arduino-computer communication is performed via a serial (over USB) interface that is natively supported in most modern programming languages.

Arduinos do come with one major drawback, however: controlling many devices at once can degrade timing accuracy. Generally speaking, synchronisation firmware written for Arduino must set pins high or low individually. Changing the state of a pin requires at least one processor instruction, which in turn requires at least one clock cycle. If many pins must have their states changed at once, there can be a significant time delay between the first and last pins changing state. This can be worked around by eschewing the use of the high-level Arduino BSP and interfacing with CPU registers directly in order to switch banks of pins simultaneously, however, this negates one of the advantages of using an Arduino: the simplicity of writing firmware. Furthermore, if fast response to signals is required, interrupts will be required to ensure an immediate response. Interrupt handling routines add significant complexity to firmware code and pause the Arduino timing functionality while they run; if interrupts occur frequently or the Interrupt Service Routine (ISR) takes a significant time to run then the timing of outputs may be impacted.

Data acquisition boards typically achieve tight timing constraints by employing FPGAs [144]. When programmed, a microcontroller (e.g., the ATmega328P used by the Arduino Uno) executes one command of the program at a time. If multiple tasks need to be performed they must be interleaved in code or execution must be multiplexed via the use of interrupts. Time-critical operations (e.g., communications interfaces like I^{2}C or SPI) are typically implemented in hardware components called peripherals, allowing user code to be run in parallel and without interfering. In contrast, an FPGA is programmed with a hardware description, which causes the device to reconfigure its internal circuitry such that it performs an equivalent set of logical operations to those contained in the hardware description. Since the ‘program’ on an FPGA is in effect made up of several physical circuits, independent modules within that program will inherently run in parallel, similar to the hardware peripherals on a microcontroller. This makes them ideal for performing many tasks in parallel, but in a synchronous manner. Historically, FPGAs have been difficult to utilise outside of professional electronics engineering companies due to their high cost and stringent electronic requirements, however, in recent years, cost effective hobbyist FPGA development boards have become available (e.g., the Au and Cu boards from Alchitry and the Arty and CMOD series’ from Digilent), which have made FPGA technology more accessible, similar to what has been achieved by Arduino in the microcontroller space [145]. Even though several vendors now provide hobbyist development boards for FPGA development, programming the boards is usually significantly harder than programming an Arduino due to the complexity of the development software. Alchitry have attempted to alleviate this issue by developing their own hardware description language, Lucid, and corresponding IDE.

The microscope system used in this project requires synchronisation between the DMD, up to 7 lasers and 3 cameras. Additionally, the ability to account for rolling shutter artifacts by gating the lasers using the ‘expose all’ signal from the camera is desirable. The large number of inputs and outputs would have made implementation on an Arduino difficult and the laser gating requirement prevents the use of a NIDAQ board. It was therefore decided that an FPGA would be utilised for microscope synchronisation.

2.1.2. CairnFocal Control Software

Operation of the CairnFocal requires uploading images to the DMD that are appropriate for the desired technique. Previous work undertaken in the Cadby Lab used custom MATLAB scripts to interface with the DMD SDK. While effective for testing, the scripts provided only a basic user interface, required users to supply their own patterns in MATLAB’s matrix file format and MATLAB’s licensing prevented Cairn Research, the industrial sponsors of this project, from distributing this code to other customers. As a result, a new program was developed to control the DMD. This program did not use MATLAB and included a GUI designed to make performing common microscopy modalities simpler.

2.1.3. µManager Device Adapters and Plugins

As the system is composed of devices from several different manufacturers, integration into a single software package would take a significant amount of time and effort. However, with the exception of the laser bank and the DMD, all the devices were already supported in µManager. A device adapter was written for the laser bank and the DMD was controlled via the software mentioned in section 2.1.2 and described in detail in Section 2.3.2. One drawback of using µManager is that its Multi-Dimensional Acquisition engine contains restrictions to the order in which the dimensions are traversed. For example, when performing a z-stack, the z dimension must be traversed before the time dimension, preventing multiple images from being taken at each z position before moving onto the next. This interferes with the acquisition of some modalities and so a plugin to control acquisition was also developed. This plugin will be discussed in future chapters. Throughout the majority of this project, the DMD was controlled primarily through the CairnFocal Control software, independent of µManager; an SLM device adapter and accompanying plugin were developed primarily for use with other DMD illumination systems developed by Cairn Research.

2.2. Materials and Methods

2.2.1. Optical Components

The CairnFocal used in this project utilised a 13.7 µm pixel V-7001 XGA Super-Speed V-Module DMD (ViALUX GmbH, Chemnitz, Germany), control of which is via the ALP 4.3 API (a C library made available by ViALUX on Windows versions 7-11, Microsoft Corporation, Washington, USA). The DMD was housed in the CairnFocal unit (Cairn Research Ltd., Faversham, UK), which also contained the correction mirrors and DMD to microscope relay optics. Illumination was provided by a Laser Diode Illuminator (LDI) 7 (89 North, Vermont, USA) through the left-hand illumination port on the CairnFocal. The dichroic mirror was a zt-402-468-555-638rpc excitation clean up filtering was performed with a zet-402-468-555-638x and emission filtering with a zet-402-468-555-638m (Chroma Technology Corporation, Vermont, USA). The CairnFocal unit was mounted on the left side port of a Nikon Ti microscope frame (Nikon UK, Surrey, UK). Imaging was performed with either a 1.49 NA CFI Apochromat TIRF 100XC Oil or a 1.49 NA CFI Apochromat TIRF 60XC Oil objective (Nikon UK, Surrey, UK). Detection was performed with two 11 µm pixel, 95% QE, back-side illuminated 95B cameras (Teledyne Photometrics, Arizona, USA) mounted on the camera ports of the CairnFocal unit. A white light LED ring illuminator was suspended above the objective in order to provide transverse illumination as well as to allow phase contrast imaging when coupled with an appropriate phase objective lens. The LED illuminator was driven by a variable bench-top power supply, allowing the brightness to be controlled by varying the output voltage. When control of the LED illuminator via the FPGA was required, a VN2106N3-G N-channel MOSFET (Microchip Technology, Arizona, USA) was inserted between the negative terminal of the LED illuminator and the negative terminal of the supply, such that application of +3.3 V between the gate of the MOSFET and the negative supply rail would enable the LED illuminator, without requiring a large current to be supplied by the FPGA trigger unit. Positioning of the sample in z was performed using a Nano-Z100 stage (Mad City Labs, Wisconsin, USA).

2.2.2. FPGA Development

The development board selected for the trigger interface development was the CMOD A7-35T (Digilent, Washington, USA) which uses an Artix-7 XC7A35T FPGA chip (Xilinx, California, USA). FPGA programming was done using Xilinx Vivado 2019.1.3 and Xilinx SDK on Ubuntu 18.04 for hardware description and C MicroBlaze firmware respectively. Initial electronics prototyping was performed on stripboard, using TinyCAD and VeeCAD for layout. The Printed Circuit Board (PCB) was designed in the EasyEDA design package. PCB manufacturing and Surface Mount (SMT) soldering was performed by JLCPCB; Through-Hole (THT) components were soldered by hand. Complimentary computer software libraries were written in the Rust programming language using the CLion IDE (JetBrains, Prague, Czechia).

2.2.3. CairnFocal Control Development

The CairnFocal control program was developed in C++ 20 using the Microsoft Visual Studio 2019 IDE on Windows 10. The GUI was written using the GTKmm library, using the Glade user interface designer for GUI layout. For integration with the FPGA trigger interface libraries, a C wrapper was developed in JetBrains CLion to allow the Rust code to be called from C++.

2.2.4. µManager Development

µManager device adapters were written in C++, using Visual Studio 2019. Initially, the Windows 7.1 SDK compilers were used for compatibility with the upstream µManager project; once the µManager project updated to the Visual C++ 2019 compilers, the device adapters were updated to use the same, in order to maintain compatibility. Plugins were developed in JetBrains IntelliJ using the most up to date version of the Kotlin programming language available at the time.

2.3. Results

2.3.1. FPGA Synchronisation

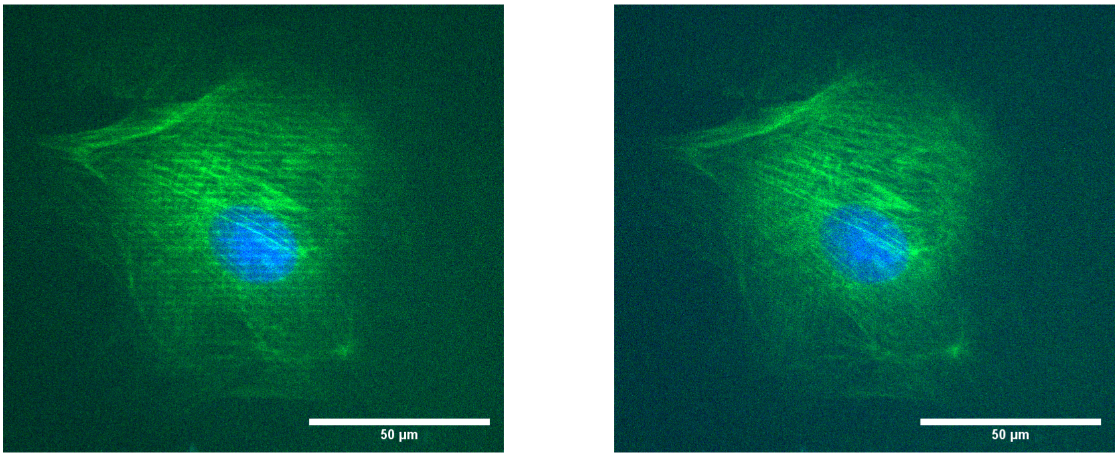
As mentioned in Section 2.1.1, careful control of the synchronisation between the DMD, cameras and lasers are required for correct operation of the CairnFocal, with different modalities requiring different synchronisation between the elements of the system. The synchronisation required for the modalities presented in this work fall roughly into three categories: ‘widefield-type’, ‘confocal-type’, and ‘ISM-type’.

Widefield

In widefield-type modalities, a single image is uploaded to the DMD and it is displayed continuously throughout acquisition. In this mode, no synchronisation between the camera and the DMD is strictly necessary, as long as the image is displayed on the DMD before the camera starts acquisition. It is desirable to prevent illumination from reaching the sample when the camera isn't taking data, in order to reduce photo-bleaching and phototoxic effects. However, this is better achieved by disabling the light source due to the comparatively low contrast ratio (i.e., the ratio between the transmitted light intensity when the DMD is in the all on position to the light intensity when the DMD is off) of the DMD. Disabling the light source to prevent damage is not overly time critical and as such can be done in software, for example by using µManager's auto-shutter feature. When using rolling-shutter sCMOS cameras to image fast moving samples, it is often desirable to gate the light source such that the sample is only illuminated during time periods where all the rows of the camera are exposing simultaneously, in order to prevent rolling-shutter artifacts. Unlike disabling the laser to prevent sample damage, the light control required for effective global shutter operation *is* time critical and must be performed in hardware, usually by connecting the 'all rows exposing' signal to the enable input on the light source. For cameras with only a single expose out signal, this can cause difficulties when attempting to perform multi-colour imaging as an expose out signal is required per illumination colour. These multiple signals can't simply be produced by connecting the one expose out signal to all the laser enable inputs, as this would cause all the light sources to be illuminating simultaneously; the expose out lines must be cycled, such that only a single light source is enabled during the exposure of any given image. The 95B provides a solution to this problem for up to four colour imaging, by providing four separate expose out signals, which it can be instructed to multiplex between by setting the appropriate setting within µManager. In reality, no special hardware synchronisation, beyond that which is already available as a result of the hardware already in the system, is required to perform widefield-type acquisitions. The ability for custom synchronisation hardware to produce the signals required for widefield-type acquisitions is still required, however, as otherwise switching between widefield-type modalities and modalities requiring more complex synchronisation would require physically rewiring connections, so as to cut the synchronisation hardware out of the loop when performing widefield-type modalities.

Confocal

Confocal-type modalities are those in which multiple images are uploaded to the DMD all of which are displayed during a single camera exposure. The synchronisation between the camera exposure and the lasers is identical to that when performing widefield-type acquisitions but the synchronisation between the DMD and camera is more complex. In a standard confocal pattern, the DMD images consist of blocks of *on* pixels separated in all directions by *off* pixels; between consecutive DMD images, the blocks are translated along one dimension, when this would result in a repeated image, the blocks are also translated in the perpendicular direction, performing a raster scan of the field. This sequence of DMD images ensures that the whole field is swept out by the blocks of *on* pixels. While the exact number of times the pattern is displayed during an exposure is unimportant, each image in the pattern must be displayed for exactly same amount of time to ensure even illumination of the sample. Figure 2.2 demonstrates the difference between confocal images[1] with and without synchronisation between the DMD and the camera. If the light source is gated to perform effective global shutter, the exposure time used to determine the DMD update frequency should be the 'expose all rows' duration; the total exposure time of a single row should be used otherwise.



(a)

(b)

Figure 2.2.: (a) Confocal image where no synchronisation between the DMD and the camera was enforced. The DMD was left to free-run, changing images at its maximum rate (44 µs per image). The camera's exposure time was set to 10 ms. The confocal pattern contained 100 images and as such, was swept out 2.3 times in a single camera exposure, resulting in the visible striping. (b) Confocal image with synchronisation between the DMD and camera enforced. The same confocal pattern and exposure time settings were used but an external trigger signal was applied to the DMD such that the 100 images were all displayed once and for an equal time during exposure

[1] To avoid ambiguity, this thesis will refer to images uploaded to and displayed on the DMD as 'DMD images', a collection of DMD images as 'patterns', raw camera data as 'camera frames' and final processed images as 'images'

ISM

Similarly to confocal-type patterns, ISM-type patterns consist of multiple images. Unlike confocal-type operation, however, ISM-type operation collects one camera frame per DMD image. Generally, effective global shutter operation is a requirement in this mode of operation, to prevent 'ghosting' in parts of the image as the DMD switches image. In the case of single colour operation, it's possible to achieve the necessary synchronisation between the DMD and the camera by connecting the expose out signal of the camera directly to the trigger change input of the DMD, however, this doesn't allow for multi-colour imaging. Even if the multiple expose out signals available on the 95B were connected together via a digital OR gate to correctly update the DMD, the enabled laser colour would change once per camera frame, rather than once the pattern had completed; e.g., if the ISM-type pattern contained 100 DMD images, the laser colour should only change after 100 camera frames have been acquired.

FPGA System Overview

In order to achieve the synchronisation requirements described above, the design needed to support enough outputs to drive the DMD, the 7 lasers on the LDI, and one output for each of the cameras in the system in order to support a mode of operation where the FPGA controls the cameras or where one camera's expose out signal triggers the other cameras to expose. It was decided that the device would support up to 4 cameras, taking the total number of outputs to 12. Additionally, to allow the FPGA to determine which of the light sources should be on at any given time and to allow for effective global shutter operation, the device requires one input per camera.

To allow the device to be reconfigured to perform any of the above modalities, a connection to the acquisition computer is required. The CMOD-A7 optionally exposes a serial-over-USB interface that can be used for this purpose. Typically, FPGAs are coded in a Hardware Description Language (HDL) (e.g., Verilog), which describes the circuit required to perform the desired calculations. While good for describing basic digital logic or tasks that can be described in terms of a collection of small state machines, it can be difficult to use HDLs to describe sequential tasks that are typically well suited to implementation in procedural languages, e.g., C/C++, executed on microcontrollers; handling serial communication is one of these tasks. Typically, implementing a working serial API requires the device awaiting a new message from the host computer, parsing that message to determine the desired operation, performing the desired operation, and responding with a message detailing the result of executing the operation. While certainly not impossible to describe in an HDL, it is a task much better suited to procedural languages and microcontrollers. As the FPGA is capable of being configured to perform the job of arbitrarily complex circuits, it is possible to implement a circuit that performs the job of a microcontroller which can then be programmed in C/C++. Xilinx provides an implementation of a Reduced Instruction Set Computer (RISC) microprocessor architecture called MicroBlaze, which can be instantiated via a drag-and-drop interface within Vivado. The MicroBlaze core provides the hardware required for basic program execution (i.e., what is required for a working fetch-execute cycle). The core can be extended via the use of appropriate Advanced eXtensible Interface (AXI) peripherals to perform custom hardware operations, e.g., communication protocols, interaction with other devices such as ROM and RAM chips, and the custom microscope synchronisation circuits developed during this project. AXI peripherals expose a series of registers which can be read from and/or written to control the behaviour of the peripheral. The registers are mapped into the MicroBlaze's address space such that communication with the peripherals from C/C++ code is performed by accessing the appropriate memory locations. Figure 2.3 shows a block diagram detailing the interconnections between the MicroBlaze and the peripherals in the trigger system.

FPGA Trigger Peripheral Design

The trigger peripheral designed in this project is designed around several independent, 32-bit counter circuits. Each counter circuit can be configured to either count the CMOD A7's internal 100 MHz clock or one of the 4 exposure inputs. The counter's 'period' input determines the value at which the counter value wraps back around to 0 and the 'on' and 'off' parameters determine the counter values for which the output signal should go high. The output signal can be configured to be gated by one of the exposure input signals, to allow for effective global shutter operation. When counting the internal clock, an optional clock division setting allows for reducing timing resolution in exchange for allowing longer clock periods. Finally, there is an option to allow the counter to run in 'continuous enable' mode in which the 100 MHz timebase is the counter source but the counter only increments when the selected exposure signal is high. Figure 2.4 shows a block diagram of the counter module, 12 of which are combined in parallel in to produce the final FPGA trigger peripheral. The AXI interface consists of 49 32-bit registers (4 per counter and one for global settings). The first 3 registers associated with each camera control the period, on and off settings respectively. The final register associated with each counter is the configuration register, which contains the exposure selection, gate selection, clock division, and 'continuous enable' settings. The global configuration

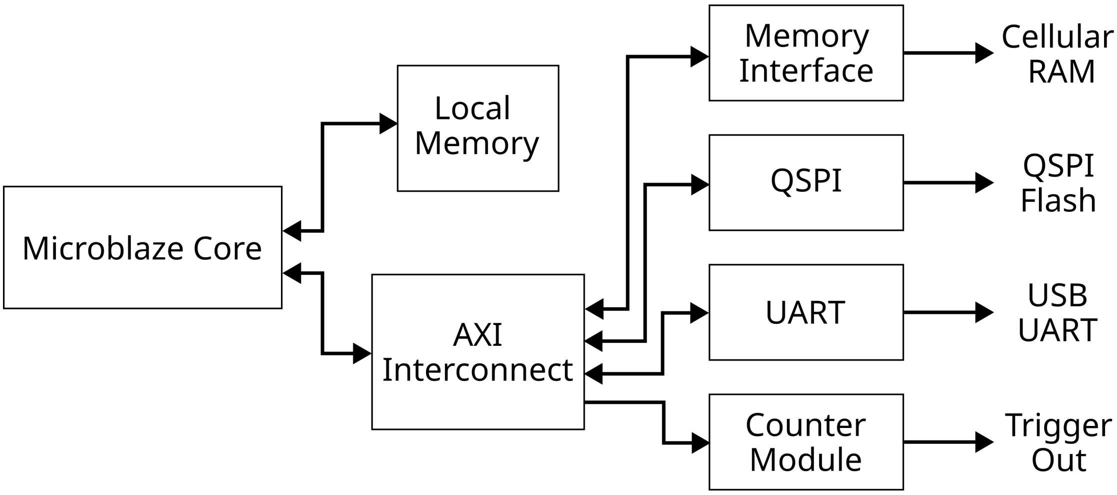


Figure 2.3.: Block diagram of the FPGA trigger system. The MicroBlaze module implements a basic single core Central Processing Unit (CPU) running on a 100 MHz clock. The 100 MHz clock is derived from the external 8 MHz quartz crystal oscillator using Phase-Locked Loop (PLL) circuitry available on the FPGA. The MicroBlaze core is directly connected to two modules, the Local Memory module, which provides a small amount of quick to access Random Access Memory (RAM) on the FPGA itself, and the AXI interconnect, which provides a memory-mapped interface to each of the 4 CPU peripherals. The Memory Interface generates the signals required to communicate with the off-FPGA RAM chip, which provides a larger but slower to access pool of RAM than the on-FPGA RAM. The QSPI module provides an interface to the Quad Serial Peripheral Interface (SPI) Read Only Memory (ROM) chip, which is used to configure the FPGA on startup. The UART block implements the control signals for the Serial-over-USB chip used to communicate with the host computer. Finally, the Counter Modules block implements main the counter peripheral as described in the FPGA Trigger Peripheral Design Section

register contains a global reset signal, which is used to reset all counters simultaneously; a pause counters signal, which prevents all counter values from changing until it goes low; and an enable outputs signal, which disables all outputs while it is low.

As an example use case, consider an imaging protocol where it is necessary to take 3 widefield images using the LDIs 640 nm laser followed by a 2 using the 470 nm laser followed by 1 using the 405 nm line. Images need to be recorded on two cameras simultaneously and effective global shutter based on the first camera's exposure needs to be employed. This protocol requires the use of one of the trigger interface's inputs and 6 of its counters/outputs (one per camera, one per laser and one for the DMD). As the

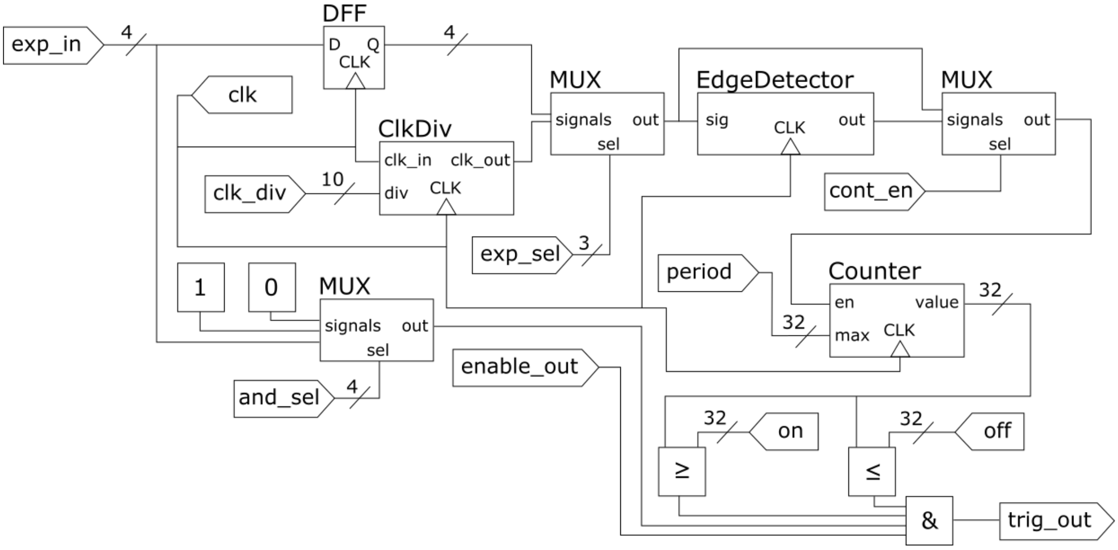


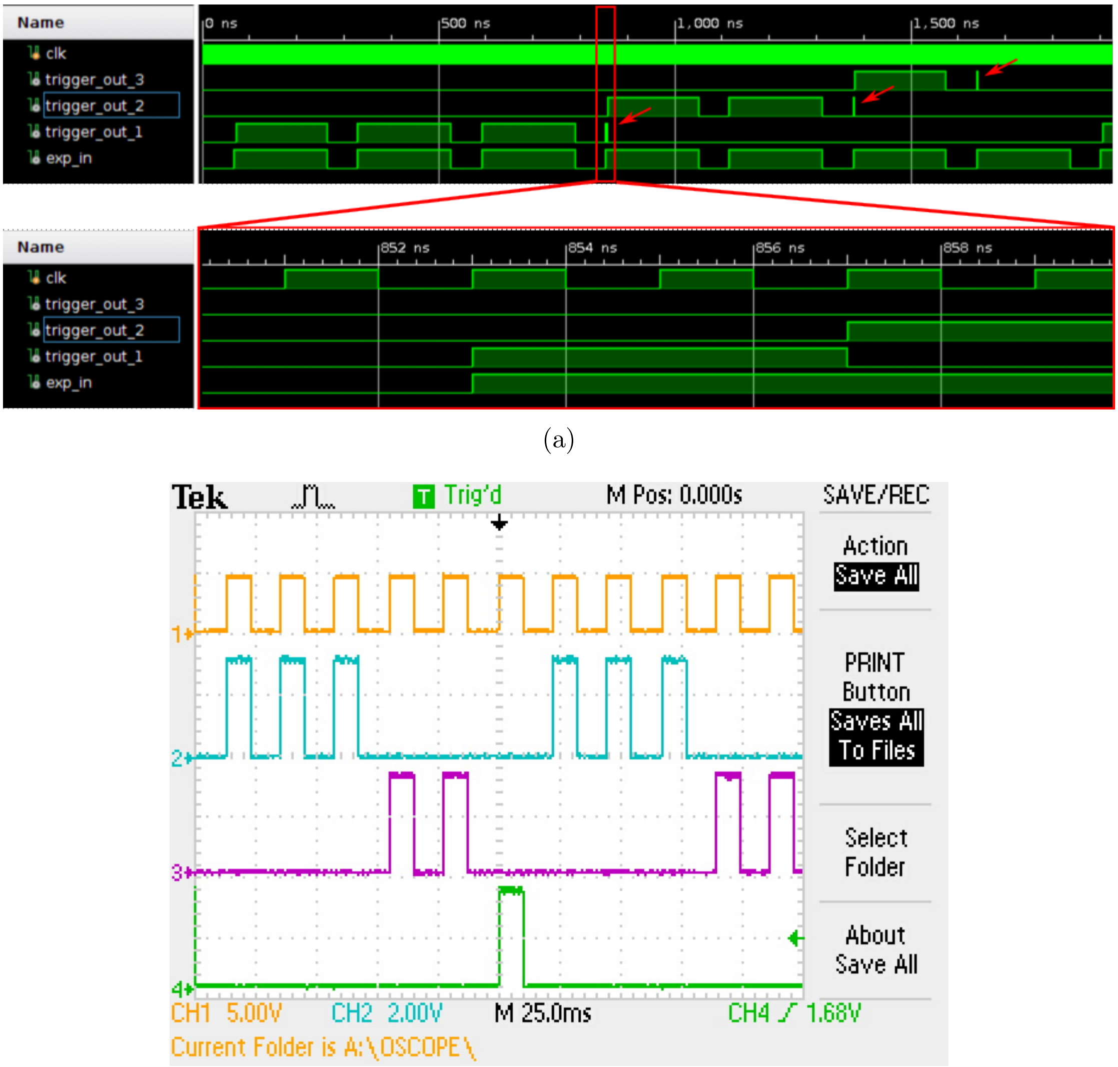
Figure 2.4.: Block diagram of the FPGA counter module. Inputs are shown as elongated pentagons with wires connecting to their pointed ends and outputs as pentagons with wires connected to their flat ends. The module uses 5 submodules, each shown with their name on the top left, inputs on the right and bottom and outputs on the right. Reset signals have been omitted for clarity. Clock inputs are indicated with a triangle below the name. The DFF module acts as a single cell buffer, transferring the value on its input to its output on the rising clock edge. The MUX modules select one of their input signals based on their select input and mirror it on their output. The ClkDiv module acts as a clock divider, taking an input clock signal and producing a slower output clock, where the frequency of the output clock is given by the frequency of the input clock divided by the div input. The EdgeDetector module produces a 1 tick long output pulse whenever the input signal transitions from a low to a high state. The Counter module is a 32 bit counter with configurable maximum value, which increases its value on a rising clock edge if its enable signal is high

imaging modality required is widefield, the setup for the DMD output is simple, setting the period to 0, the on time to 0 and the off time to 1 ensures that the output will go high once the acquisition is started, triggering the DMD to display its image immediately. Since 6 images are required in total, the period for all the laser counters should be set to 7, allowing the counters to count from 0 to 6 before resetting. The on time for the 640 nm laser should be set to 0 and the off time to 2. The on and off times for the 470 nm laser should be 2 and 3 respectively and both should be set to 6 for the 405 nm laser. The count sources and 'and source' for both laser channels should be set to exposure 1 (which is assumed to be connected to the expose all signal of camera 1) such that the counter values increment once per camera frame and the lasers are only enabled during the global shutter period of camera 1. Since synchronised acquisition on multiple cameras is needed, it makes sense to place both cameras in external trigger mode and to generate the trigger signals using the internal clock of the trigger interface. Assuming edge trigger operation of the cameras, the count source can be set to the internal clock, the period set to match the desired interval between frames, the on time set to 0 and the off time set to some number less than the period but long enough that the camera will detect the trigger pulse (half the period is usually a suitable choice). Figure 2.5 shows a simulation of the output of the counters associated with the lasers when set up in this configuration and corresponding measurements of the final physical trigger interface.

FPGA Computer Interface

As mentioned previously, communication between the FPGA and the host computer is over a serial interface. The interface runs at 9800 baud with 8 data bits, 1 stop bit and no parity bits via the serial-over-USB protocol. Table A.1 describes the valid serial commands. Commands are parsed by C firmware running on the MicroBlaze soft processor. The firmware continuously reads from the serial port one byte at a time, placing each onto the end of a 32 byte buffer and echoing the character back to the computer (echoing the received bytes allows a rudimentary Teletype (TTY) style interface if the user is typing commands directly over the serial port via a program such as PuTTY). If the end of the command buffer is reached, the firmware starts overwriting from the beginning of the buffer in order to prevent a crash. The firmware recognises two special characters, firstly, a backspace character (ASCII 0x8 or `'\b'`) which discards the previous character from the command buffer and sends the 3 character sequence `'\b \b'` (backspace, space, backspace) to the computer in order to remove the previous character from the terminal and reset the cursor appropriately. Secondly, the `';'` character, which delimits the end of a command. Upon receiving a semicolon, the firmware inserts a null character at the end of the command buffer to delimit the end of the contained string, sends a new line character to the computer, parses the contents of the command buffer and executes the contained command. The firmware responds with either the information requested or an error message if parsing fails; in both cases $\sim$ and a newline character are used to delimit the end of the response.

The firmware was designed to support a TTY style interface to allow a user to directly communicate with the device without needing specific software for control of the trigger interface. This is mainly of use when debugging and it is usually preferable to have software that provides an abstraction over the reading and writing of the configuration



(a)

(b)

Figure 2.5.: (a) Simulation of the FPGA trigger interface. The simulation replicates the output of the trigger interface using the setup described in Listing C.1, whereby 3 outputs are used in global shutter mode. The first (bottom) output is instructed to turn on for 3 images, the next for 2 and the final (top) for 1 image. All outputs are placed into effective global shutter mode. The simulation was run with a clock period of 2 ns so the simulated timings don't directly correspond to the equivalent real world setup. The bottom trace shows the simulated trigger-in signal, which is high for 100 ns with a 30 ns gap between simulated exposures. The outputs show small glitches when switching from enabled to disabled due to the outputs being gated using the un-clocked expose in signal. The signal counted by the counters has to pass through metastability removal and edge detection circuitry so is delayed by a couple of clock cycles, as shown in the zoomed traces. (b) Oscilloscope measurements of the true FPGA outputs with the same setup. The orange trace is the expose all signal from the camera, the other three traces are the outputs from the FPGA

registers. To that end, a small SDK was developed to make programmatic communication with the trigger interface simpler. The SDK was written in Rust using the 'serialport' crate for cross platform serial communication. Wrapper libraries were written for C and Kotlin to allow the library to be consumed in C/C++ and JVM projects respectively.

The initial prototyping of the trigger interface was performed using LEDs to indicate when outputs went high and an Arduino to simulate the expose out signals of a camera. Once the design of the FPGA code was finalised, a prototype circuit was built on stripboard to allow interfacing with the actual components in the system. The circuit was designed to overcome two primary issues: firstly, the expose out signals produced by the 95B are 5 V Transistor to Transistor Logic (TTL) signals and the FPGA chip is not 5 V tolerant; secondly, to provide a mechanism to increase the maximum current output of the trigger interface and to protect the FPGA chip from short circuits on its output pins. The first problem is solved by reducing the 5 V TTL signals to 3.3 V TTL signals. In the prototyping stage, this was done using a Metal-Oxide-Semiconductor Field-Effect-Transistor (MOSFET) configured to act as a bidirectional logic level shifter, however, this had the disadvantage of its inputs effectively being connected to pull-up resistors, causing the inputs to go high when not connected. In the final version, this was performed using TXB0101 logic level shifters (Texas Instruments, Texas, USA). On the output side, SN74LVC244AN line driver chips (Texas Instruments, Texas, USA) were used to buffer the outputs to allow greater maximum current output and protect the FPGA from short circuits and static on the outputs. The outputs from these chips were connected to the outputs of the device but also used to switch 'debug' LEDs, indicating when each output is high or low. The inputs were also connected through spare pins on the line drivers to allow debug LEDs to be attached without loading the inputs. The external chips were powered via a 3.3 V regulator (TC1264 3.3VAB, Microchip Technology, Arizona, USA) that was in turn powered by a 5 V power supply.

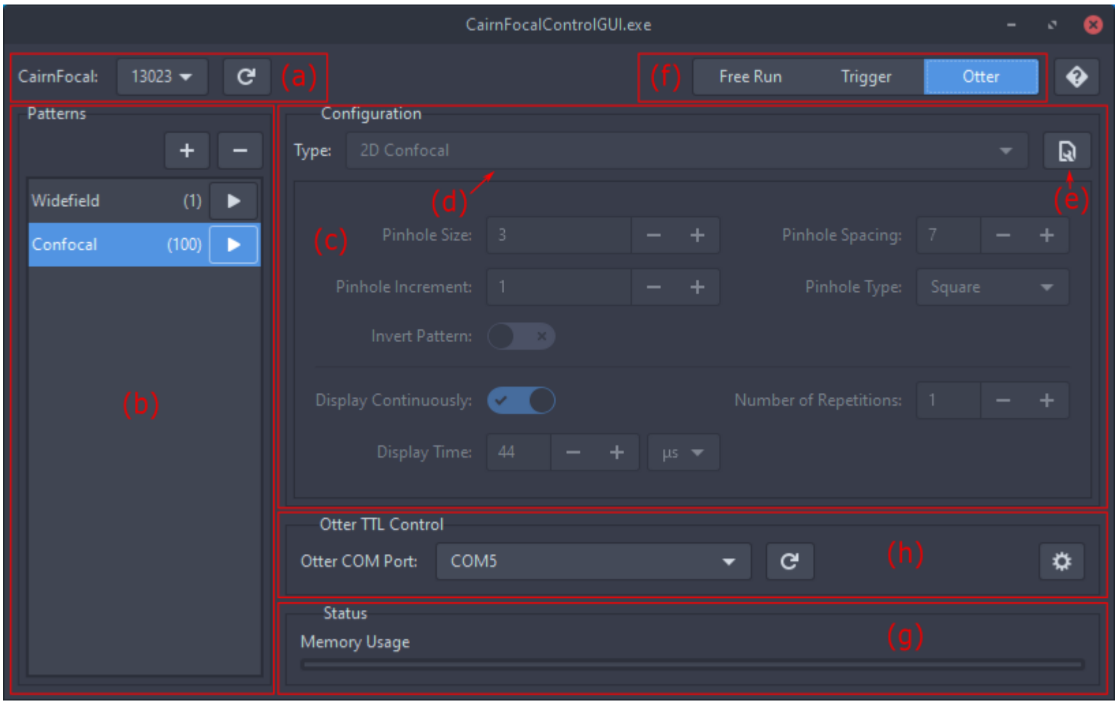
2.3.2. Software (CairnFocal Control)

As an industrially sponsored PhD, one of the primary aims of the project was to improve the usability of the CairnFocal system in order to allow it to be more easily sold to biologists rather than groups developing microscopes. To that end, a software package was produced in order to simplify interaction with the DMD. The software was designed to provide an intuitive GUI for configuring the system to perform common microscopy techniques. Figure 2.6 shows the main window of the GUI. The software revolves around the concept of 'patterns', sequences of one or more related images and associated timing information, which can be used to perform a function (e.g., imaging, setup, calibration, etc.). Once a pattern has been created and uploaded to the DMD, the user can display that pattern at the click of a button. Choosing to display another pattern automatically stops the previously running pattern. As all patterns are pre-loaded to the DMD on creation, swapping between patterns is fast, taking only the time for the command to be sent over the USB interface.

The user can also control the synchronisation method for the DMD, by switching between 'Free Running' and 'Triggered' modes. In Free Running mode, DMD image updates are not synchronised with any other devices, instead updating periodically according to the timing settings provided in the pattern description. In Triggered mode the DMD will ignore the provided timing settings and update the displayed image in response to the rising edge of an external trigger pulse. The DMD supports pulse width modulation of individual mirrors to allow finer control over illumination intensity at each position, however, in order to simplify timing and synchronisation, the software restricts all patterns to being binary (i.e., no pulse width modulation) and places the DMD in 'binary uninterrupted' mode. In the default mode of operation, the DMD is not able to update the image without first blanking the display during the configuration of the next image (the dark time), and each image is displayed for a set exposure time (the illumination time). When placed in triggered mode, the illumination time still applies and once completed the DMD will blank while awaiting the next trigger pulse. Trigger pulses that arrive during the illumination time are ignored. In binary uninterrupted mode, however, the next image can be loaded onto the DMD during the display of the current image. As a result, no dark time is required and the image will be displayed continuously until the next trigger pulse arrives. This significantly simplifies setup, particularly in triggered mode. The final display mode also places the DMD into triggered mode, however, also interfaces with the FPGA trigger interface in order to automatically reconfigure it in response to pattern changes (see the next section).

FPGA Trigger Interface Integration

The CairnFocal control software provides integration with the FPGA trigger interface to allow it to automatically reconfigure the trigger interface's settings in response to a change in pattern. Once the trigger interface has been connected, pressing the settings button next to the COM port selector (see Figure 2.6) opens the trigger interface configuration menu. The first page allows the user to label each of the inputs and outputs as well as mark which are physically connected. The second tab allows the user to configure the desired settings for each of the patterns. The exact settings vary slightly depending on which of the three categories described in Section 2.3.1 the pattern falls into, how-



CairnFocal: 13023

Widefield (1)

Confocal (100)

Type: 2D Confocal

Pinhole Size: 3

Pinhole Spacing: 7

Pinhole Increment: 1

Pinhole Type: Square

Invert Pattern: ○

Display Continuously: ✓

Number of Repetitions: 1

Display Time: 44 µs

Otter COM Port: COM5

Memory Usage

Figure 2.6.: The CairnFocal Control GUI. The dropdown box in (a) is used to select the serial number of the DMD to allow for the possibility of systems with more than one DMD installed. When a serial number is selected, the software automatically attempts connection. If the DMD is off or not connected when the software is started, it will not appear in the list. The refresh button rescans the available DMDs. Once the DMD is connected, the controls in the pattern list, (b), become available. Pressing the '+' button prompts the user to enter a name for the new pattern. Once a name has been entered, a new pattern is created and the pattern configuration section, (c), is enabled. Once configured, patterns can be activated by pressing the play button next to each name, which then changes to a stop button to allow ending of pattern display. Pressing the '-' button deletes the selected pattern. During pattern configuration, the dropdown menu (d) is used to select the type of pattern. The other controls in the pane are automatically reconfigured to provide settings to that pattern type. Once the desired settings have been selected, the upload button, (e), can be pressed to upload the pattern to the DMD; this button then becomes the edit button which can be used to reconfigure the pattern. The controls in (f) are used to select the trigger mode. (g) displays the amount of DMD memory used by the currently uploaded patterns. (h) allows the setup of the FPGA trigger interface.

ever, they all provide options that allow the user to specify which device should control the timing of the acquisition, whether effective global shutter is desired and the order that different light sources should be triggered in. Some settings are dependant on the exposure time of the camera, which may vary regularly as the user optimises the imaging settings. To prevent the user from needing to transfer this information manually, the FPGA trigger interface integration can receive this information via the network interface described in the next section. Once the configurations have been set and the appropriate trigger mode is selected, the software automatically reconfigures the trigger interface when the user selects a new pattern to display. The software caches the values of the configuration registers and only requests a value change when the new value differs from the value currently in the cache in order to reduce the time it takes to switch between patterns with similar trigger settings. The 'BGN' and 'END' commands (see Table A.1) are also automatically sent on pattern start and stop to enable and disable the outputs appropriately.

Network Interface

While a stand-alone program to control the DMD was necessary for commercial reasons, communication between the software controlling the DMD and the software controlling other devices in the system is often desirable, particularly for performing 'smart microscopy' protocols, which was the aim of Chapter 4. As will be described in Section 2.3.3, the control of all the other devices in the system is performed via µManager. At first glance it would make sense, to provide a device adapter and plugin that would allow µManager to control the DMD directly, however, this would have resulted in a large amount of duplicated effort to provide feature parity between the stand-alone control program and the µManager equivalent. An alternative approach taken in this project, was to allow for communication between the stand-alone program and µManager. To enable this inter-process communication, the CairnFocal Control starts a Transmission Control Protocol/Internet Protocol (TCP/IP) server on which it can receive commands from other programs. The server is implemented using the ZeroMQ networking library, which provides a common abstraction over several inter-thread, inter-process, and networking protocols (including TCP/IP) and provides several general purpose sockets that provide implementations of common messaging patterns that can be combined to produce complex messaging systems.

The ZeroMQ library introduces several abstractions, most importantly protocol agnostic 'messages' and 'sockets' [146]. In ZeroMQ, messages represent the data sent between sockets and are in effect represented by a linked list of 'frames'. Each frame consists of a length (in bytes) followed by an array of bytes containing the frame's data. In general, a ZeroMQ message will consist of zero or more address frames followed by a zero length frame and finally one or more frames containing the message data. Sockets represent endpoints capable of sending and receiving data. Each of the different socket types perform a different operation when sending or receiving a message. The sockets used in the CairnFocal Control network interface are, 'ROUTER', 'DEALER', 'PUB' and 'SUB'. ROUTER sockets allow asynchronous sending and receiving of messages from multiple sources simultaneously. When a message is received from a client socket, the ROUTER prepends a unique address to the address frames of the message before passing it onto application code. When the application sends a reply, it pops the first address frame off the message and uses it to determine the physical address (e.g., a TCP/IP address) of the client to forward it on to. The DEALER socket is used as the counterpart to the ROUTER socket, allowing the client program to asynchronously send messages to and receive messages from the server running in the stand-alone program; the dealer doesn't touch the address frames, simply forwarding them onto the client program or to the ROUTER socket. The PUB and SUB sockets are used to implement multi-cast techniques where one message is sent to multiple endpoints. The PUB socket manages connections from multiple SUB sockets and ensures each message from the application is sent to all connected SUB sockets; there is no data flow in the opposite direction.

The network architecture used is outlined in Figure 2.7. When a client initiates a connection to the CairnFocal Control program, it sends the message 'CONNECT' to the CairnFocal Control program's ROUTER socket from its DEALER socket. Upon receiving a connect message, the CairnFocal Control program responds directly to the client with the IP address and port number of the PUB socket. When the client receives this response it immediately connects its SUB socket to the PUB socket at the received IP address before requesting an update via its DEALER socket. Whenever the CairnFocal Control program receives and update request or performs an action that changes the state of the DMD (e.g., uploading a new pattern, running a pattern, etc.), it sends an appropriate update to all connected clients, allowing all clients to be kept abreast of changes caused by other clients or user interaction with the CairnFocal Control program. Additionally, if a message has not been sent over the PUB connection within the last second, the CairnFocal Control program will automatically send a 'heartbeat' message. If no messages (heartbeat or update) have been received by a client within several seconds, it can infer that it has lost connection to the CairnFocal Control software and either attempt reconnection or abort. Table A.2 gives a description of the commands that can be sent by clients, all of which are single frame messages (ignoring the address frames). Table A.3 describes the updates sent by the CairnFocal Control program. With the exception of the Heartbeat message, all the update messages consist of two frames; the first frame indicating the type of update, and the second containing the update information. As the update information can be quite complex, it is encoded in Java Script Object Notation (JSON). The JSON schema for the various forms of update information are given in Appendix B.

2.3.3. µManager

While the CairnFocal control program provides the ability to interact with the DMD and the FPGA trigger interface, it doesn't provide the ability to interact with any other components in the system. With the exception of the DMD, the most important electronic components in the system are the cameras. Photometrics use µManager as their preferred imaging software, which also supported almost all of the other components, i.e., the microscope frame's objective turret, auto-focus system and the z-stage. The only component that wasn't supported was the LDI, so a µManager device adapter was written to allow it to be controlled from within µManager.

The LDI uses a serial interface, the commands for which were provided by 89-North. The device adapter provides properties that allow the user to change the laser powers, laser shutter states, the functional mode of the system (i.e., if the laser bank is enabled or in idle mode), enable and configure external control of laser powers or shutters and to clear any fault states detected by the LDI. It also implements the µManager CShutterDevice interface, allowing automatic integration with µManager's auto shutter features. The user can select up to 4 separate laser lines to be automatically enabled and disabled in response to µManager's auto shutter commands. The device adapter was subsequently integrated into µManager's repositories and it is now shipped with all new µManager installations.

Section 2.3.2 describes the design and implementation of a mechanism to allow µManager to communicate with the CairnFocal Control program to instruct it to change the currently displayed pattern. This is sufficient to support the work in future chapters, however, there are some protocols (e.g., Fluorescence Recovery After Photobleaching (FRAP), targeted optogenetic stimulation, etc.) that benefit from the ability for the user to select regions to illuminate by drawing on images received by the camera. This is not possible using the network interface described above, though it could be extended to make it possible. These sorts of protocols were the primary use case that µManager's SLM device type was designed to solve and since these protocols are unlikely to be used at the same time as the protocols made available by the CairnFocal Control program, a device adapter was written to allow µManager to communicate directly with the DMD via the SLM API. The device adapter allows µManager to upload a single pattern consisting

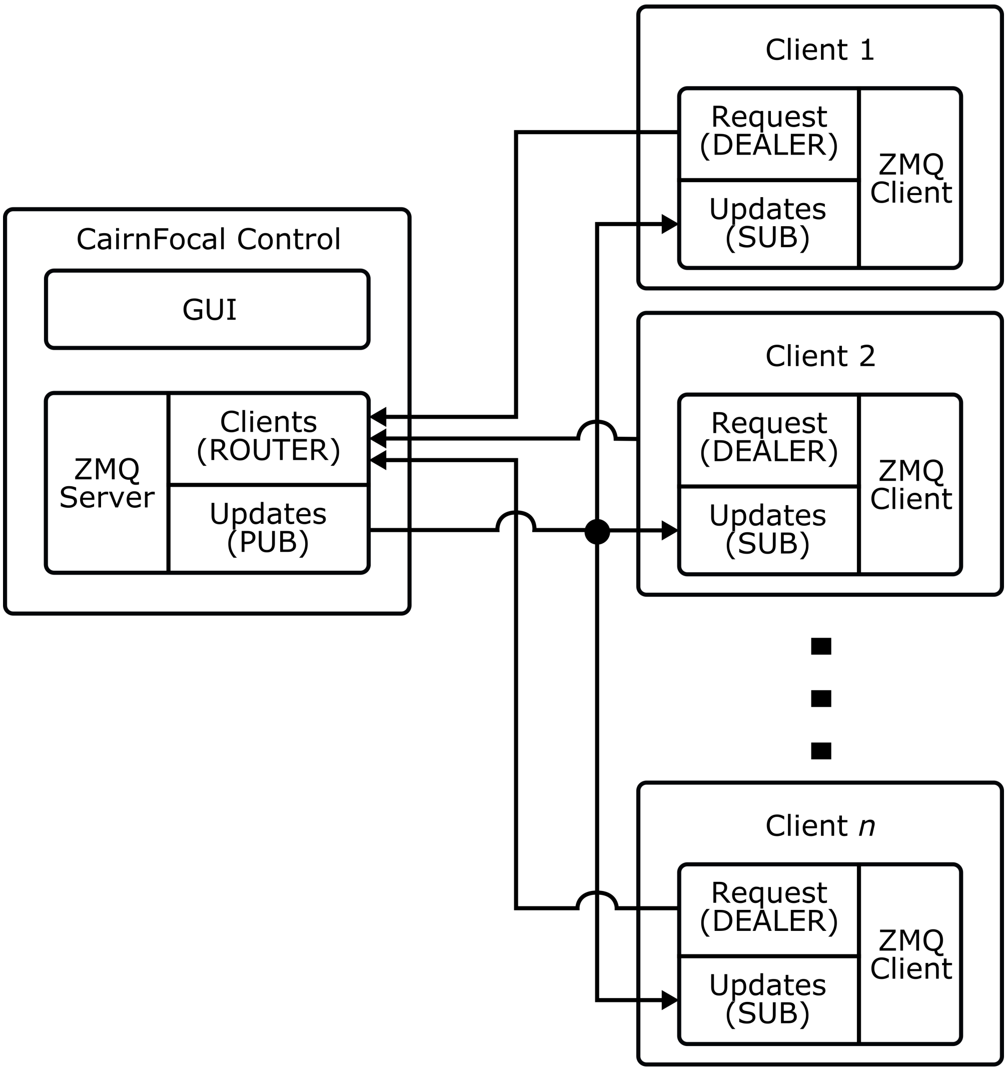


Figure 2.7.: The network architecture of the CairnFocal control Network Interface. The server hosted in the stand-alone CairnFocal control program consists of two sockets. The ROUTER, hosted on port 59014, is used to handle connection and command requests from clients. The PUB socket, hosted on port 59451, is used to send updates about the current state of the DMD to all connected clients. Similarly, each client has two sockets. The DEALER socket is used to send command requests as well as to request the PUB socket's port number on first connection. The SUB socket is used to subscribe to updates from the CairnFocal control program.

of multiple frames. Since µManager's SLM API expects to upload 8-bit images and has abstractions to deal with finite illumination times, the DMD is not placed in binary uninterrupted mode and as such 256 illumination intensities are available for each pixel.

The device adapter also has properties to allow the user to place the DMD into triggered mode.

The Projector plugin that ships with µManager provides a calibration routine that produces a bidirectional mapping between DMD pixel coordinates and camera coordinates [135]. First, an image is taken of the DMD with all pixels in the off position, which is used as a background image in further analysis. Next, individual images are displayed on the DMD, each containing a circle of radius 10 pixels. The background image is subtracted from the image of the circle and the result is heavily blurred using the ImageJ `GaussianBlur.blurGaussian` method, with sigmaX = sigmaY = 10 pixels and accuracy = 0.01. The position of the maximal pixel value in the blurred image is taken to correspond to the centre of the displayed spot on the DMD. Initially, 5 spots are displayed close to the centre of the DMD; their detected positions are used to calculate an affine transform, via linear least squares regression, to map points between DMD and camera space. This transform assumes that the mapping between DMD and camera space is linear, which is not true in many optical systems where aberrations (e.g., field curvature or astigmatism) may distort the image non-linearly with distance from the optical axis. To account for this, the second stage of the calibration first approximately determines the maximum extent of the DMD that is visible from the camera using the linear mapping calculated in the previous step. Next, it determines the camera space positions of a 7×7 grid of circles (rectilinear in DMD space) that spans the visible extent of the DMD. Each 4×4 group of circles defines the corners of a patch in the image; the corner positions are used to determine the affine transform over the corresponding patch. Once the affine transforms for all the patches are calculated, an improved estimate of the DMD pixel coordinate for a given camera coordinate can be achieved by determining the patch in which the camera coordinate falls and using the affine transform corresponding to that patch. The plugin then allows the user to draw ROIs on camera images using ImageJ's built-in tools. It can then automatically map the corners of the ROI to DMD coordinates and upload an image to the DMD with the corresponding ROI filled.

While the µManager projector plugin's calibration routine works well, the rest of the plugin is somewhat lacking for many protocols. In particular, any protocol where the user wishes to take a series of images each with a single ROI illuminated is not supported; this is because µManager's plugin uploads a single image to the DMD where all the provided ROIs are illuminated at once. Additionally the µManager projector plugin is required to support all projector devices which, due to the inevitable variation between DMDs manufactured by different companies, results in unintuitive behaviours when using the other features of the plugin on certain DMDs. In order to improve the user experience, a new projector plugin was written, which was optimised to work with the ViALUX DMDs specifically. The plugin utilises the same calibration routine (albeit reimplemented in Kotlin) but allows a sequence of images containing individual ROIs to be uploaded to the DMD. Once calibrated, the DMD can be placed in external trigger mode so that, with appropriate connections between the expose out signal of the camera and the DMD, the displayed image will be updated whenever the camera takes a picture. The calibration can be saved to a JSON file and reloaded at a later date, to prevent the requirement to recalibrate every time the plugin is initialized. Figure 2.8 shows the newly created projector plugin's interface and example images produced using it to sequentially illuminate several ROIs.

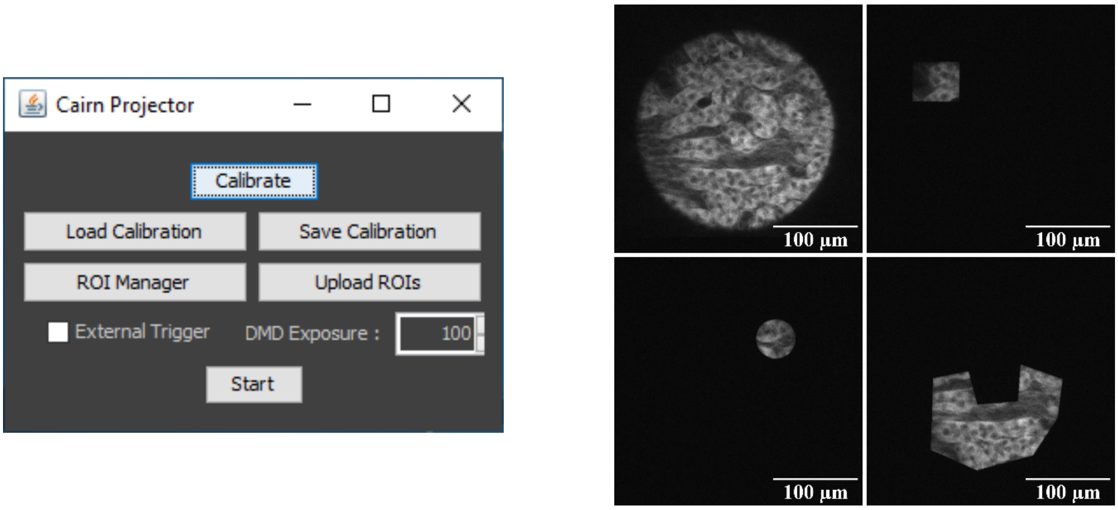
In addition to the work described here, a µManager plugin was developed to improve the usability of the CairnFocal via the stand-alone CairnFocal Control program from within µManager; the development of this plugin is detailed in Chapters 3 and 4.

2.4. Discussion

2.4.1. FPGA Synchronisation

The results described in Section 2.3.1 demonstrate the ability of the FPGA trigger interface to produce the signals required to synchronise the various components of the microscope when performing modalities falling into one of the widefield-like, confocal-like or ISM-like categories. Communication with the trigger interface has been integrated with the CairnFocal control program to allow it to automatically configure the trigger interface to control acquisition.

The main alternative to FPGA trigger interface is the NIDAQ series from National Instruments. NIDAQ cards typically have a large number of digital outputs available, arranged into several banks that National Instruments refer to as 'ports', with each port supporting slightly different features. Ports supporting 'buffered output' modes are capable of producing hardware timed signals. They achieve this by having a RAM buffer on the device that stores a list of desired outputs. A clock signal is selected that determines when the the output should switch to the value contained in the next element of the buffer. This method allows for more complex signal patterns than the counter based approach used in the FPGA trigger interface. The FPGA trigger interface's output shape is configured by selecting a period, start time, and end time, meaning that they can only output periodic waveforms with one 'pulse' per period. While the outputs of the NIDAQ board are also periodic (with the period being determined by the number of samples uploaded to the buffer), the waveform can consist of multiple 'pulses' of arbitrary width



(a) (b)

Figure 2.8.: (a) The CairnFocal Projector Plugin GUI. Initially all the controls other than the Calibrate and Load Calibration buttons are disabled. Pressing the Calibrate button starts the calibration routine. Images are taken using µManager's SnapLiveManager so the user can view the calibration data as it comes in. Additionally, a red cross is drawn at each detected spot location. The Save Calibration button allows the user to save the calibration results to a JSON file, which can then be loaded via the Load Calibration button in subsequent runs of the program. The ROI Manager button opens the ImageJ ROI Manager window, allowing the user to select multiple ROIs. Once the user has selected all the desired ROIs, clicking the Upload ROIs button produces the DMD images corresponding to the ROIs and uploads them to the DMD as a single image sequence. Since the DMD is not used in it's binary uninterrupted mode, the exposure must be specified, which is done by entering a value in the DMD Exposure spinbox. The External Trigger checkbox simply sets the corresponding property in the device adapter, enabling or disabling the external trigger mode of the DMD appropriately. The Start button instructs the DMD to start running through the images in the image sequence. (b) Example images taken using the plugin. A sample of fluorescently labelled mouse kidney was imaged using a 470 nm laser using a 20x air objective. 4 ROIs were uploaded to the DMD and the external trigger mode was used to ensure a single image of each region was acquired. The top left image corresponds to the ROI filling the entire camera field. The remaining images correspond to rectangular, circular and polygon ROIs respectively.

per period. This means that a protocol consisting of, for example, two images using the 470 nm laser, followed by one image with 640 nm illumination and one final image with

470 nm light, wouldn't be possible with the FPGA interface but would be trivial with the NIDAQ. While this may seem like a major restriction, it is rare that asymmetric sequences like this are actually required and the FPGA more than makes up for this with its flexibility in trigger sources.

It is possible to select an external clock source with a NIDAQ board, however, this feature was primarily designed to allow synchronisation between multiple NIDAQ devices by using one of the NIDAQ's clocks to drive the output of the others. The expected frequency of the external clock signal must be provided to the NIDAQs during configuration for accurate timing of their outputs, as a result they are not tolerant to clock signals with frequencies that are unknown or vary significantly during acquisition. As the FPGA trigger interface's counters are just counting rising edges, it does not require the counting signal to be of a constant frequency. Another advantage of the FPGA trigger interface is that each of the counter modules are independent of one another, whereas the NIDAQ uses a single RAM buffer per output port, resulting in all outputs for a port updating simultaneously in response to a clock edge. This allows the FPGA trigger interface to effectively drive multiple independent subsystems of a microscope asynchronously with no more complexity than driving a single synchronous system, whereas the NIDAQ would require generating complex patterns to drive all parts of the system synchronously. Additionally, the FPGA trigger interface allows the outputs to be gated using any of the 4 inputs, allowing for operation of rolling shutter cameras in an effective global shutter mode, which is not possible with a NIDAQ without supplying external circuitry. The NIDAQ boards provide many functions that the FPGA trigger interface does not, e.g., digital inputs, analogue inputs and outputs, etc., all of which can be synchronised with respect to one another. However, these additional features come at the cost of a significantly more complex programming interface that is unintuitive, poorly documented and relies on a proprietary communications protocol/drivers that are only available on Windows. In contrast, FPGA interface communicates using a serial interface over a USB connection, drivers for which are available on all major operating systems. The SDK was designed to be as simple as possible and, being written in Rust, is available on Windows, MacOS and Linux. An example program for configuring it to perform a simple protocol from the Kotlin programming language is shown in Listing C.1.

While the FPGA trigger interface has been sufficient for the work discussed in future chapters, working with the interface over a prolonged period of time and discussions with Cairn Research have revealed ways in which it would benefit from further development. The first is the ability to support analogue outputs for the control of, for example, laser intensity. The CMOD A7 doesn't have built in Digital to Analogue Converters

(DACs), so this would require external chips to be added to the PCB. Using a RAM buffer, similar to that used by the NIDAQ boards, would be the most flexible approach to providing analogue outputs; however, a large number of protocols would be satisfied by simply allowing on-demand software control of the analogue outputs, which would be significantly easier to implement. If the RAM buffer approach was taken, then it may be desirable to use an external RAM chip to save FPGA resources.

A second area of improvement is in the design of the counter modules. Currently, each counter can count the rising edges of one of the 4 inputs or the internal clock. One trivial extension would be to allow the counters to be configured to count falling edges or both rising and falling edges. More interesting would be the ability to count an arbitrary combination of the 4 inputs. Similarly, the gating functionality could be extended to allow gating on arbitrary combinations of inputs as opposed to just one. These features could be easily implemented by mimicking the way that FPGAs represent the majority of their boolean logic, i.e., by using the 4 inputs as an address into a lookup table, which is filled with 0's and 1's corresponding to the outputs in the boolean logic table representing the desired logical operation. Taking this idea one step further, the outputs of the FPGA trigger interface could be decoupled from the outputs of the counter modules. By defining a 16 input lookup table for each output, each output could be made to correspond to arbitrary combinations of all inputs and counter outputs. Interestingly, this would allow the example requiring a NIDAQ (discussed earlier) to be supported with the FPGA trigger interface by gating the 470 nm laser output with the exposure signal and the inverse of the 640 nm output. In principle, it would also be possible to feed back the current output of the counters their inputs using a similar technique, though this would require careful considerations of the FPGA's setup and hold times and it is unclear if this would provide any real world benefit. Each of these additions would require a corresponding change to the SDK and control software, care would need to be taken to propagate these changes in such a way that the SDK and software does not become overly complex.

2.4.2. CairnFocal Control

The CairnFocal Control program provides a user-friendly interface that allows users to upload patterns to the DMD to perform common microscopy techniques. Multiple patterns may be uploaded to DMD memory at once and the user may switch the currently running pattern by clicking the play button next to its name; switching is fast due to the patterns already existing in the DMD's memory. The software supports running the DMD in free-running mode, where the DMD updates according to the pattern's timing parameters, or triggered mode, where the DMD only updates in response to an external trigger. The software also supports integration with the FPGA Trigger Interface, to allow automatically reconfiguring the FPGA in response to a change in pattern. Finally, the software provides a Network Interface that allows for other processes to programmatically control the operation of the DMD.

Previous work in the Cadby Lab used custom MATLAB scripts for communicating with the DMD. The ViALUX SDK doesn't provide code to allow directly accessing the DMD from MATLAB, instead it provides only a C library, i.e., a DLL and header file. MATLAB provides an API for interfacing with C libraries, however, the resulting MATLAB code can be difficult to follow due to the convoluted syntax for calling the library's functions. While the scripts did produce GUIs, they were very simplistic and exposed only a very limited subset of the DMD's functionality. Additionally, each script was designed for a particular experiment and produced its own GUI. In one respect this was a good thing, each GUI only contained controls required for performing a particular experiment, however, it also resulted in a lot of duplicated code between scripts and a disjointed user experience if a user required performing multiple experiment types. Additionally, most of the experiments required the user to supply their own patterns in MATLAB's MAT format, requiring end users to have a basic understanding of MATLAB in order to use the software. Additionally, an installation of MATLAB was required to run the scripts, which caused issues with distribution for Cairn Research. The CairnFocal Control software solved these issues. The code was written in C++, which is capable of natively interfacing with C libraries so code remained idiomatic and understandable to anyone with a basic knowledge of C++. The CairnFocal Control program has a single GUI that exposes all the functionality required to perform any supported experiment and provides a set of configurable patterns that allow the user to perform all supported microscopy techniques without needing any programming skills. Finally, it only relies on free and open source libraries, so there are no difficulties in distribution.

Since ViALUX don't supply DMD drivers for platforms other than Windows, cross-platform compatibility is not a concern and one of the many Microsoft GUI frameworks would have been suitable. Unfortunately, the Universal Windows Platform (UWP) versions available at the start of the project (2018) had some severe limitations, the main one being that UWP apps had a hard dependency on the version of the operating system that they were compiled for and would not run on newer or older versions of Windows. Since Microsoft released multiple versions of Windows 10 that were difficult to tell apart, this would likely have presented a major support issue for Cairn Research. These issues have since been fixed and if the project were started anew now, C++ UWP would likely be the GUI library chosen. Windows Presentation Foundation (WPF) was another potentially good choice, although required writing a .NET wrapper for the DMD library. The first attempt at writing the CairnFocal Control program was written in C# with WPF, however, eventually this path proved to be unmanageable. Unlike C++ DLLs, C# DLLs may contain code compiled for both 32-bit and 64-bit architectures (so called Multi Platform applications). Unfortunately, ViALUX only provide a 64-bit version of their DMD library so attempting to compile a Multi Platform project will fail due to missing the 32-bit dependencies. Therefore, all C# code needed to be compiled in 64-bit exclusively. Unfortunately, at the time, the most up to date version of Visual Studio (Visual Studio 2017) was only available as a 32-bit executable and while it allowed compiling in 64-bit mode, the GUI builder would crash if this was attempted, making GUI development in WPF unworkable. This has since been fixed in Visual Studio 2022, which is now distributed as a 64-bit executable. Since the other GUI frameworks available from Microsoft were too old, focus shifted towards cross-platform alternatives. JVM options were dismissed, as their use would have necessitated writing another wrapper and they did not provide significantly better development experiences than the C++ alternatives to justify the extra time writing the wrapper. In hindsight, writing the program in a JVM language would have allowed code sharing between a stand-alone version of the program and a µManager integrated version, removing the need for the network interface. Out of the two C++ options considered, GIMP toolkit (GTK) was chosen due to worries that the Qt's more restrictive licensing might cause problems for distribution in future.

One major decision taken during the development of the CairnFocal Control program was whether to duplicate effort introducing DMD control into µManager or to find some form of interprocess communication to allow µManager to control the existing stand alone program. Since the majority of the programming time on the project went into development of suitable GUIs, it made sense to try and reduce the number of GUIs required. It was decided that a web server should be used to provide communication between the CairnFocal Control program and plugins written in µManager and ZeroMQ was chosen due to its relatively simple interface when compared to alternatives such as Boost's ASIO library, availability in multiple programming languages and the fact it had been used extensively in the PycroManager bridge, which allows control of µManager from the Python programming language [147]. An alternative approach would have been to use an interprocess communication library, such as Boost's Interprocess library. This would likely have been a simpler approach as interprocess communication like this is less likely to fail than network communication; the web server approach requires a more complex design so that it is robust to messages going missing. However, implementing a web server has two major benefits, firstly, it is easier to attach multiple client programs to a single instance of the CairnFocal Control program. Secondly, in principle, connection from clients running on separate computers should also be supported without any code changes, allowing control of the DMD from computers running operating systems other than Windows. NVidia produces a range of single board computers designed for AI and machine learning, the Jetsons [148]. They run Linux, so direct control of the CairnFocal's DMD is not possible, however, the CairnFocal Control program's integrated web server opens up the possibility of a Jetson using a trained neural network to analyse images on the fly in order to decide which modality the CairnFocal should be running at any given time, updating the running pattern over the DMD.

One of the drawbacks of the CairnFocal Control program when compared to the MATLAB scripts used previously is that it doesn't currently support the user uploading patterns of their own design. The program was designed to make adding new pattern types in future relatively simple, so this should not pose a significant barrier to future development. Providing means for new patterns to be created and configured over the network interface, both by sending commands to configure the standard patterns and by sending complete DMD images over the network, would allow for a wide range of new experimental protocols to be carried out using the CairnFocal Control program (e.g, FRAP, targetted optogenetic stimulation, etc.). This would even allow protocols where, for example, the illumination region was updated in response to changes in the sample. This would be an interesting direction for further exploration, however, it is not immediately clear how it would integrate with the FPGA trigger interface or if the overheads associated with transferring images over the network and then to the DMD would be prohibitively slow for use in live sample imaging.

2.4.3. µManager

Other than the DMD, communication with all the devices in the system is performed though µManager. This allows basic software synchronisation between various components in the system (e.g., the cameras and the light sources via µManager's auto shutter feature) and advanced acquisition modes (e.g., z-stacks and multi-channel imaging) via µManager's Multi-Dimensional acquisition. Since Photometrics use µManager as their primary camera software, there wasn't an alternative choice short of writing custom software to control every device in the system; using µManager means much wider device support out of the box than custom software would. The LDI device adapter integrated the LDI into this ecosystem and is used by several research groups internationally.

The ViALUX DMD device adapter allowed the CairnFocal to be controlled entirely from µManager. This allows either the µManager projector plugin or the new projector plugin developed during this project to use the CairnFocal to perform techniques such as FRAP or targetted optogenetic stimulation. The standard projector plugin supports any SLM or laser scanning device with a µManager device adapter and has a larger feature set than the projector plugin developed during this project. As it is required to support a wide range of physical devices, µManager's projector plugin often performs suboptimally when using a given device and has several issues when interacting with the CairnFocal's DMD, e.g., attempting to change the DMD's illumination time to less than its minimum results in an error; since the µManager projector plugin attempts to update the DMD illumination time whenever a character in the associated textbox is changed, many errors are generated if the user attempts to change from, for example, 100 ms to 200 ms by deleting the 100 and retyping 200. Since the new projector plugin only works with the CairnFocal's DMD, it is able to work around these idiosyncrasies and provide a better user experience. Additionally, it provides the ability to upload patterns to the DMD where each frame contains a single illuminated ROI, whereas the µManager projector plugin only allows all ROIs to be illuminated simultaneously, allowing experimental protocols that would otherwise not be possible. The newly developed plugin uses a reimplementation of the same calibration routine from the original plugin and as such suffers from the same restrictions, namely that the camera must be aligned so that it can see the centre of the DMD and that the DMD and camera spaces are rotated approximately an integer multiple of 90° from one another. Future work on this plugin will likely attempt to address these restrictions as well as provide support for more devices.

This Chapter described two mutually exclusive methods for controlling the DMD, one via the CairnFocal Control program and one entirely via µManager. The two methods are complimentary, the former providing a simple method by which complex patterns can be created and displayed irrespective of the state of the rest of the system and the latter providing a method by which the user can select regions in the image to illuminate. The two methods allow for following distinct protocols, however, there are some protocols which would require a combination of both approaches, e.g., performing confocal within a selected region and widefield outwith. Since both methods require acquiring exclusive control of the DMD, performing such techniques would require integrating the features of one into the other. As discussed previously, integrating the features of the CairnFocal Control software directly into the µManager would require duplicating a large amount of work. Conversely, integrating the features of the µManager approach into the CairnFocal Control software would only require extending the network interface to allow specification of new patterns and a change to the projector plugin to use these features. It is also worth noting the slight differences between how the two methods interact with the DMD. The µManager approach attempts to conform as closely as possible tho the specification laid out by the SLM device adapter API and as such must be able to display 8-bit images, though this feature is never used in the µManager projector plugin or this projects plugin. The requirement to be able to display 8-bit images means that when controlled through µManager, the DMD is not placed in binary uninterrupted mode, limiting the maximum switching speed and complicating the timing setup. Additionally, the µManager device adapter currently only supports uploading a single sequence of images to the DMD at once, making switching displayed images relatively slow. By contrast, the CairnFocal Control software preloads multiple sequences of images, making switching pattern quick, and places the DMD in binary uninterrupted mode.

2.5. Conclusion

This chapter has outlined the software and hardware development undertaken to prepare the CairnFocal system to act as a platform for performing several common microscopy techniques. Work consisted of development of an FPGA based trigger interface for synchronisation of the component elements of the system, C++ software for interaction with the DMD and extensions to the µManager platform to support previously unsupported components. Future chapters will build on this work in order to describe the use of this system for performing multiple different imaging techniques (Chapter 3) and for multi-modality imaging of live specimens (Chapter 4).

CHAPTER

THREE

IMPLEMENTATION OF COMMON MICROSCOPY TECHNIQUES ON THE CAIRNFOCAL PLATFORM

3.1. Introduction

Building on the work from Chapter 2, this chapter will describe the implementation of several common microscopy techniques using the CairnFocal platform and analyse the system's performance. Though it will primarily focus on widefield, confocal and ISM, it will briefly touch upon others. Unlike the other two techniques, ISM requires significant post processing and optionally requires the ability to predict the camera position that corresponds to a particular position on the DMD. This chapter will also detail the development of ISM reconstruction code and a method for calibrating the DMD to the camera that retains its accuracy even in the presence of highly non-linear distortions.

3.2. Materials and Methods

3.2.1. Samples

The work described in this chapter was primarily carried out on standard commercially available samples. For qualitative performance assessments on thin samples, Bovine Pulmonary Artery Endothelial (BPAE) Cells triply labelled with MitoTracker Red CMXRos, Alexa Fluor 488 and DAPI were used (FluoCells - Prepared Slide #1, ThermoFisher Scientific, Massachusetts, USA). For thicker samples, Mouse Kidney sections labelled with Alexa Fluor 488, Alexa Fluor 568 and DAPI were used instead (FluoCells - Prepared Slide #3, ThermoFisher Scientific, Massachusetts, USA). For PSF and quantitative resolution measurements, 200 nm and 100 nm TetraSpeck beads were imaged (Fluorescent

Microspheres Size Kit, ThermoFisher Scientific, Massachusetts, USA).

3.2.2. Optics

The optical path remained largely unchanged from that described in Chapter 2. For improved efficiency when imaging TetraSpeck beads, the multi-band pass filter set was replaced with a single-band Cy5 filter set consisting of an ET700/75m emission filter and T660lxpr bandpass filter (Chroma Technology Corporation, Vermont, USA). An HQ630/60m (Chroma Technology Corporation, Vermont, USA) was also used for laser clean-up. To ensure the system's Photometrics 95B was capable of surpassing the Nyquist sampling limit when using the 100× lens, a 1.5× C-Mount Fixed Focal Lens Extender (Edmund Optics, York, UK) was placed between the camera and the C-Mount port on the CairnFocal.

3.2.3. Data Acquisition

Data acquisition was primarily performed in µManager, with DMD images being uploaded via the CairnFocal Control program. For widefield, confocal and xy ISM imaging, µManager's Multi-Dimensional Acquisition (MDA) was used. For multi-dimensional ISM a custom acquisition plugin was written in Kotlin, using IntelliJ IDEA (JetBrains, Prague, Czechia). Hardware synchronisation was provided using the FPGA trigger interface described in the previous Chapter when necessary. A PM100USB power meter with S120C detector (Thorlabs, New Jersey, USA) was used for measuring illumination intensities at various points in the system. For measuring illumination distribution at the sample, a 10×, 0.3 NA objective (Nikon UK, Surrey, UK) was used to form an image of the DMD onto the sensor of a 2.4 µm pixel CellCam Centro 200MR (CairnResearch, Kent, United Kingdom).

3.2.4. Data Analysis

Image analysis was performed in ImageJ, using a combination of in-built functions, freely available plugins, custom written plugins and macros. Pinhole locations for use in ISM reconstruction were localised using the ThunderSTORM ImageJ plugin [131]. Custom plugins for reconstruction of ISM data and PSF analysis were written in Kotlin, using IntelliJ IDEA (JetBrains, Prague, Czechia). Deconvolution was performed using the DeconvolutionLab2 ImageJ plugin [28], using a simulated PSF generated by the PSF Generator plugin [130], both made available by the Biomedical Imaging Group at the

École Polytechnique Fédérale de Lausanne (EPFL). All other data analysis was performed using custom MATLAB scripts.

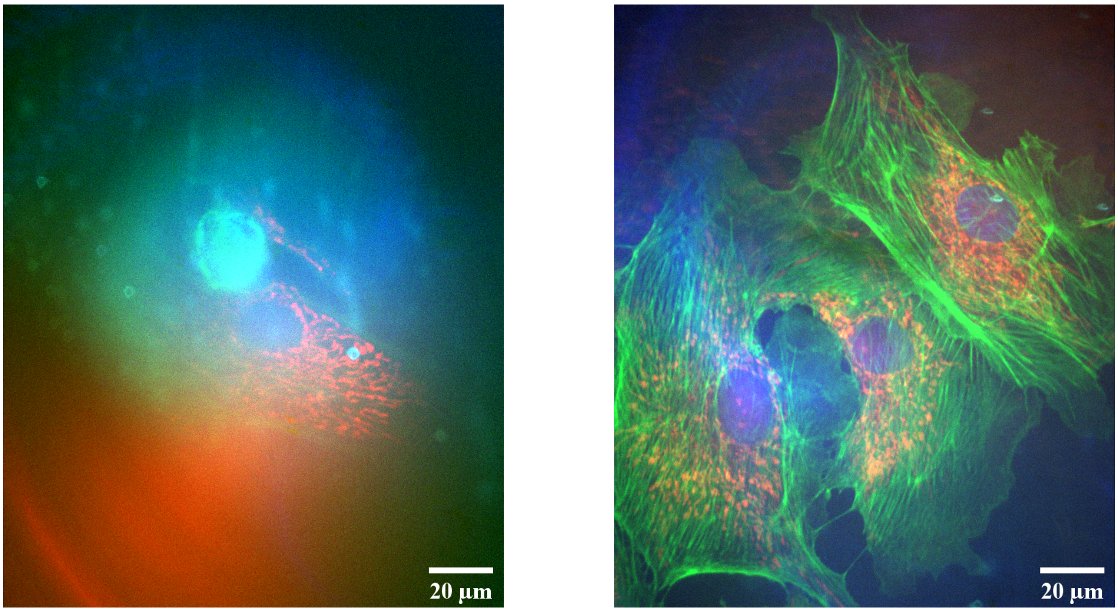
3.3. Results

3.3.1. Widefield

Performing widefield on the CairnFocal requires placing all the micro-mirrors in the *on* position. In this configuration, the DMD transmits all the illumination light striking it to the sample and diverts all returning light to the *on* camera; conceptually the DMD can be thought of as having no effect and the system behaves as if the DMD is not present and the camera is mounted on the side port of the microscope body (ignoring than distortions and other aberrations introduced by the DMD and any misalignment of the correction optics, which will be discussed further in Section 3.4.1).

At the start of the project, the lab's CairnFocal was a version 1 prototype. Work by previous students had been incorporated into Cairn's design of version 2, which was offered as a free upgrade. Once the upgrade was completed, without further modification, the CairnFocal was capable of generating epi-fluorescent widefield images similar to those shown in Figure 3.1a. Baffling was introduced around the filter cubes and lenses in order to prevent scattered light from reaching the cameras and introducing background, excitation filters were introduced to reduce direct bleed-through of excitation light to the camera, all optics were cleaned, and the illumination optics external to the CairnFocal were changed to ensure that excitation light was collimated when entering the correction optics. Specifically, the new illumination optics consisted of a 35 mm achromatic lens mounted in an adjustable z mount. The LDI's light guide was mounted in an x-y adjustable mount such that the position of the end of the light guide was adjustable relative to the collimating lens in all 3 dimensions. In this setup the combination of the collimating lens and the CairnFocal correction optics form an image of the end of the light guide on the DMD, magnified by approximately 10×. The z mount was adjusted to ensure the sharpest image of the end of the light guide appeared on the DMD (as viewed by the on-side camera) and the position of the image was translated to the centre of the DMD using the x-y mount. Once these optimisations had been made, the CairnFocal was capable of producing images similar to those shown in Figure 3.1b.

In order to reduce phototoxic effects, it is often desirable to avoid epi-fluorescent imaging, opting to use transverse illumination and label-free imaging techniques instead. To allow these techniques to be performed, the system was equipped with a ring of white light LEDs, mounted above the objective. With no further adjustments, this



(a)

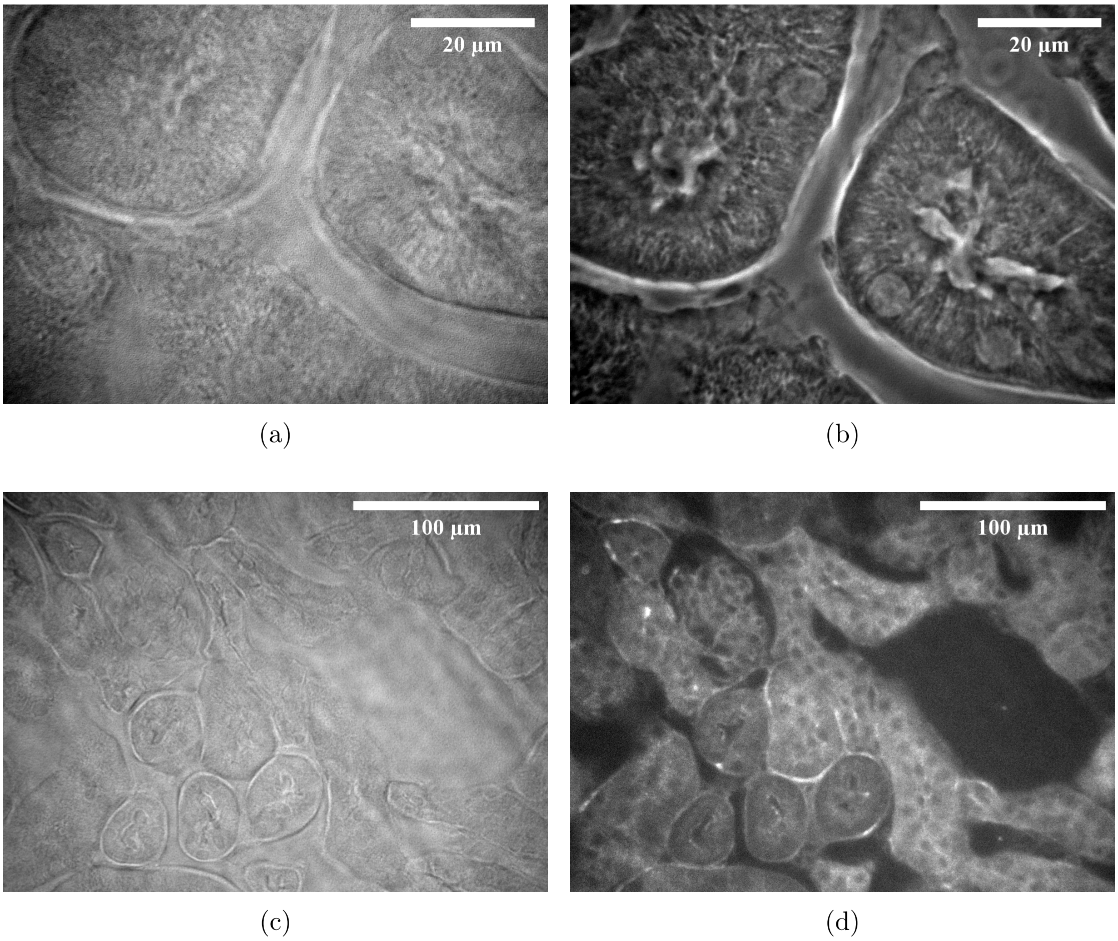
(b)

Figure 3.1.: (a) A representative 3-color image of a BPAE cell produced by the CairnFocal V2 without optimisation (60x, widefield). The image suffers from low contrast as well as several structures due to dust on various optical components. The small circular objects are primarily due to dust particles on the DMD window. The larger 'swirl' patterns are primarily due to dust on the filters. The low contrast is caused by excitation light scattered off the DMD reflecting behind the emission filter and reaching the camera. (b) A representative 3-color image of BPAE cells after optimisation of the CairnFocal by including baffling and an excitation filter as well as cleaning the system's optical components. There are still some dust particles and swirling visible which was removed with further cleaning.

allows brightfield imaging as shown in Figure 3.2c and 3.2a. Additionally, as described in [5], by using a lens with an integrated phase plate and positioning the illuminator appropriately, phase contrast can be performed (Figure 3.2b) and by using a low NA (< 1) lens and dropping the LED ring down so that it sits outside the cone of acceptance of the lens, darkfield imaging may be performed (Figure 3.2d).

3.3.2. Confocal

Widefield imaging allows for the use of low intensity illumination by collecting as much emitted light as possible. Alternatively, label free techniques can be used to further reduce phototoxic effects. The drawback of these techniques is that resolution is limited, both



(a)

(b)

(c)

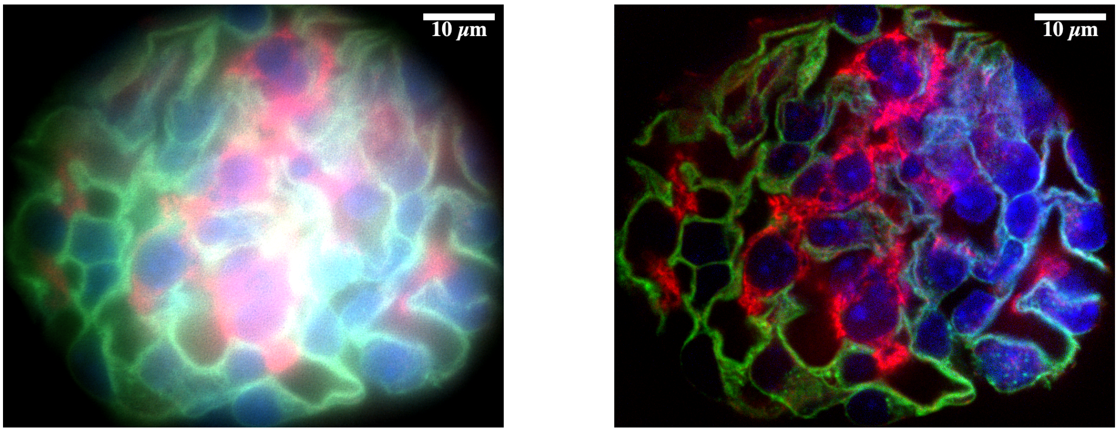
(d)

Figure 3.2.: A comparison of various label free imaging techniques performed on the CairnFocal. (a) and (c) show brightfield images of a mouse kidney section at 100× and 30× magnification respectively. Contrast is determined by absorption of light as it is transmitted through the sample and as such tends to be lower than achievable with other techniques. The sample appears darker on a light background. Using the system's ring illuminator and an appropriate objective lens, phase contrast can be performed, as shown in (b). Brightness in the phase contrast image is determined by relative phases of light that has passed through different parts of the sample. The phase contrast image is of higher contrast and more detailed structures are visible. By using a lens with a numerical aperture less than 1, the ring illuminator can be dropped such that it is placed outside the acceptance angle of the objective lens. As a result, direct illumination light cannot reach the detector and an image consisting of only the light scattered by the sample is formed. This results in a light-on-dark image which is of higher contrast than the equivalent brightfield image, as shown in (d). While an improvement over brightfield, darkfield requires the use of a low numerical aperture objective lens, reducing resolution, and generally requires much higher illumination intensities due to the scattered light signal being of significantly lower intensity than the transmitted light signal which is rejected.

laterally and axially. The lack of z resolution associated with widefield techniques causes light from out of focus planes to be detected alongside the light from the focal plane. While this isn't a major problem for thin samples, it causes images of planes deep inside thick samples to appear blurred and low contrast. Confocal provides an increased axial resolution by utilising non-uniform illumination and a pinhole placed in an image plane to reject out of focus light. It also provides a marginal increase in lateral resolution [41], however, this effect is small in comparison to the effects on axial resolution so this is often overlooked. The cost of this increased resolution is a requirement for higher illumination intensities and/or longer illumination times to get a SNR comparable to that of Widefield.

On the CairnFocal, confocal operation is achieved by placing small groups of pixels in the *on* state, surrounded by pixels in the *off* state [102]. When illumination light strikes a DMD pixel in an *on* state the light is directed to the sample, when it hits an *off* pixel it is not; the microscope optics focuses an image of the pattern displayed on the DMD onto the sample. Typically, the numbers of *on* pixels in a group is chosen to be small, so that when demagnified onto the sample, the image consists of several small (often diffraction limited) illumination spots, separated by regions of no illumination. Light emitted from the sample is collected by the microscope optics and similarly focused onto the DMD. Light striking an *on* pixel is directed through one branch of correction optics onto one of the cameras and light striking an *off* pixel is directed through the opposite branch and onto the opposite camera. The pattern is then translated across the DMD within a single camera exposure to build up an image of the entire field.

In this configuration, when considering the image formed on the *on* camera, the DMD acts like the pinhole disk used in a traditional SDCM. The CairnFocal has several differences when compared to a traditional SDCM, most notably that the emitted light that would be rejected by the pinhole disk on a SDCM is instead directed to the *off* camera on the CairnFocal. Additionally, in a SDCM, the size of the pinholes and spacing between the pinholes is fixed, on the CairnFocal these parameters are determined by the images displayed on the DMD and can be trivially changed during acquisition in order to optimise for the particular sample being imaged. The number of DMD images per confocal image is determined by the pinhole size and spacing. An individual DMD image can be displayed multiple times during the camera exposure as long as all the DMD images in the pattern are displayed the same number of times and as long as the on time of an individual DMD image is greater than 44 µs. Figure 3.3 shows a comparison between the images produced using widefield and confocal on a thick sample of mouse kidney. In the confocal image, the pinhole size was chosen to be 4 pixels × 4 pixels and the pinhole spacing 15 pixels (corresponding to 548 nm and 2.055 µm respectively when demagnified onto the sample).



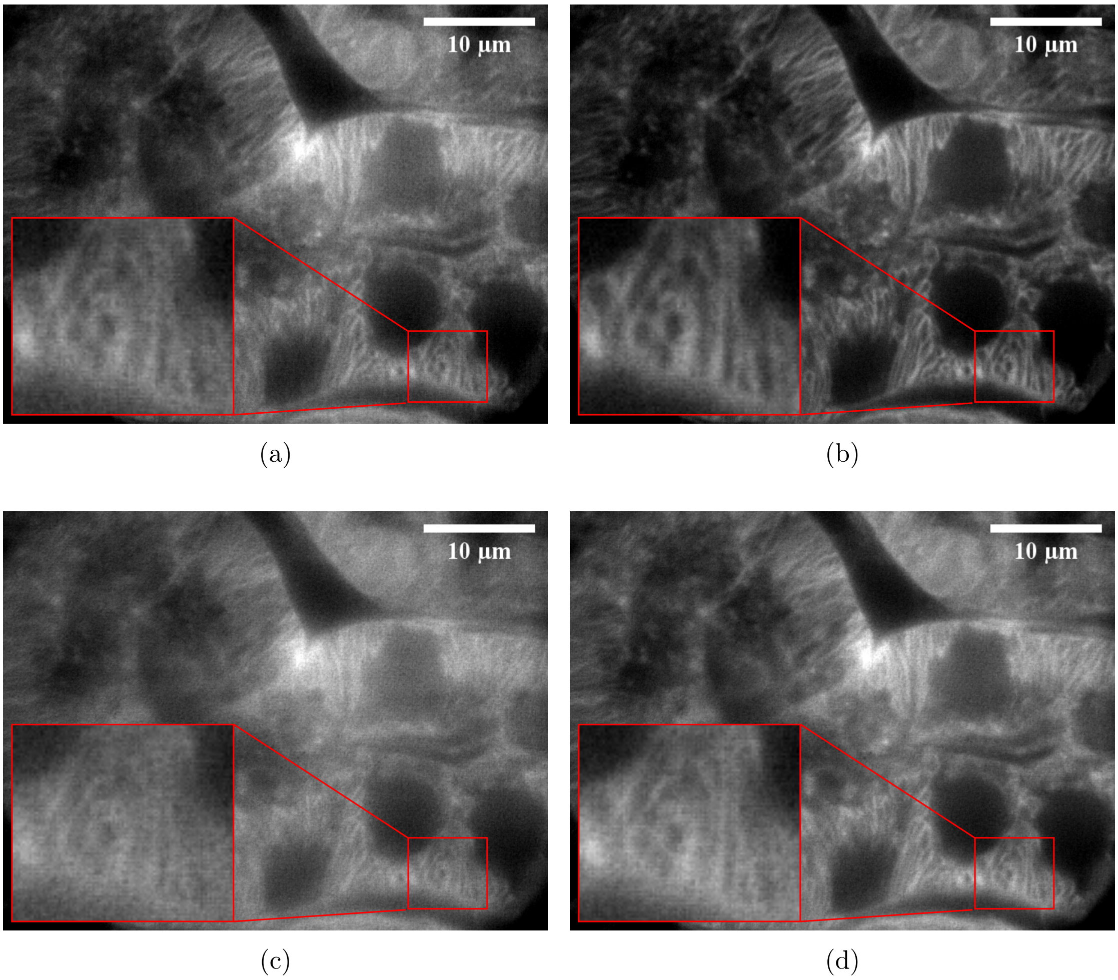
(a)

(b)

Figure 3.3.: (a) 3 colour widefield image of a mouse kidney section than on the CairnFocal. Due to the thickness of the sample, light emanating from out of focus planes mixes with the signal from the in focus plane, resulting in low contrast and obscuring fine detail. (b) The same region of the sample taken in SDCM emulation mode. In this mode, most of the light from out of focus planes is rejected by the pinholes, improving optical sectioning and allowing detail to be seen even deep inside thick samples. Laser powers and exposure times were changed between (a) and (b) in order to provide the best quality images for both modalities. The circular appearance of the field is due to under-filling the DMD, as is discussed in detail in Section 3.4.1.

Intuitively, changing the pinhole size and spacing of the confocal patterns allows a trade off between SNR and confocal sectioning, however, it is not immediately clear how much of an effect changing each of the parameters makes. Imagining a two-spot confocal pattern, it's clear that reducing the spacing between the pinholes will reduce the effectiveness of the confocal as emitted out of focus or scattered light from one confocal volume may make it through the other pinhole; reducing the space between pinholes increases crosstalk between pinholes and reduces the effectiveness of the confocal. An ideal confocal system would use an infinitely small pinhole, however, in practice this would reject all light coming from the sample so a compromise between signal levels and sectioning ability is made. Typically the pinhole size is set at approximately 1 airy unit, however, some CLSM systems employ a variable pinhole to allow a user to adjust the pinhole size. Changing the pinhole size on a SDCM (and by extension the CairnFocal) has a secondary effect, however. Since the pinholes are also used to form the confocal illumination volumes, increasing the size of the pinhole also increases the size of the illumination volume. On the CairnFocal, with a 100x lens, setting the pinhole size to 4 pixels or more results in a confocal illumination volume that is no longer laterally diffraction limited, a condition which is often assumed in analysis of confocal systems. Figure 3.4 provides a qualitative comparison between various pinhole sizes and spaces.

To investigate the quantitative effect of varying the pinhole size and spacing, z-stacks of 200nm beads were collected in confocal with pinhole sizes ranging from 3 to 10 pixels and spacings ranging from 5 to 15 pixels. An ImageJ plugin for analysis of the resulting PSFs was written. The plugin utilised the same analysis methods used by the publicly available MetroloJ plugin [129], but extended them to support multiple PSFs within one image. The PSF Analysis plugin allows the user to select several regions of the image, each containing an individual PSF, by drawing several ROIs around the beads within the image. Each ROI is subsequently considered independently. A maximum value projection through the z-planes in the ROI is calculated and the coordinate of the highest intensity pixel is used to determine the centre of the PSF in the xy dimension. The region is then orthographically re-sliced (i.e., the y and z dimensions are swapped to produce a stack of images with axes x and z), a z-projection is taken and the maximum intensity pixel is used to localise the centre of the PSF in z. A Gaussian profile is fitted to the row of pixels passing through the centre point in all three x, y, and z directions. The standard deviations of these three fits are then used to calculate the FWHM of the PSF in each of the three dimensions. The FWHMs for each of the PSF selected by the user are then exported as a CSV file where they can be further processed as needed. Each of the confocal images showed a slight improvement in lateral resolution over the widefield, with the largest increase produced by size 3×3 pinholes with 12 spacing (296 nm ± 38 nm vs 314 nm ± 8 nm for the widefield). Figure 3.5 shows how the average FWHM in the axial dimension changed as the pinhole size and spacing was varied. A flat plane was fitted through the data using the method of Least Squares Regression. The fit indicates that increasing the size of the pinhole by 1 pixel in diameter, increased the FWHM of the axial PSF by 65 nm, whereas moving the pinholes 1 pixel further apart only reduced the FWHM by 25 nm. The minimum FWHM of the axial PSF recorded was 509 nm ± 62 nm with 3×3 pixel pinholes with a spacing of 9 pixels a $\sim 1.48\times$ improvement in optical sectioning vs the widefield's 752 nm ± 28 nm, an improvement roughly inline with the theoretical improvement suggested by [129]. Figure 3.6 shows a comparison between the PSFs of the system in widefield and two confocal modalities with different pinhole sizes and spacings.



(a)

(b)

(c)

(d)

Figure 3.4.: Comparison of selected pinhole size and spacing settings when performing SDCM emulation on the CairnFocal. All images are of a mouse kidney section, using 470 nm illumination and a 100× objective. The settings were as follows: (a) 3 pixel × 3 pixel pinholes, 7 pixel spacing; (b) 3 pixel × 3 pixel pinholes, 15 pixel spacing; (c) 8 pixel × 8 pixel pinholes, 7 pixel spacing; (d) 8 pixel × 8 pixel pinholes, 15 pixel spacing; Going from (a) to (b) resulted in a improvement in the level of detail within the image, presumably due to improving the optical sectioning by preventing crosstalk between pinholes. This comes at the expense of signal levels, requiring the laser powers to be increased by $\sim$ 5× to get comparable signal levels. Increasing the size of the pinholes (moving from (a) to (c)) increases signal levels (the exposure was dropped by $\sim$ 4× to prevent the camera saturating) at the expense of optical sectioning; the larger pinholes are worse at rejecting out of focus light. Moving the pinholes further apart ((c) to (d)) regains some optical sectioning performance, though once again at the expense of lower signal levels.

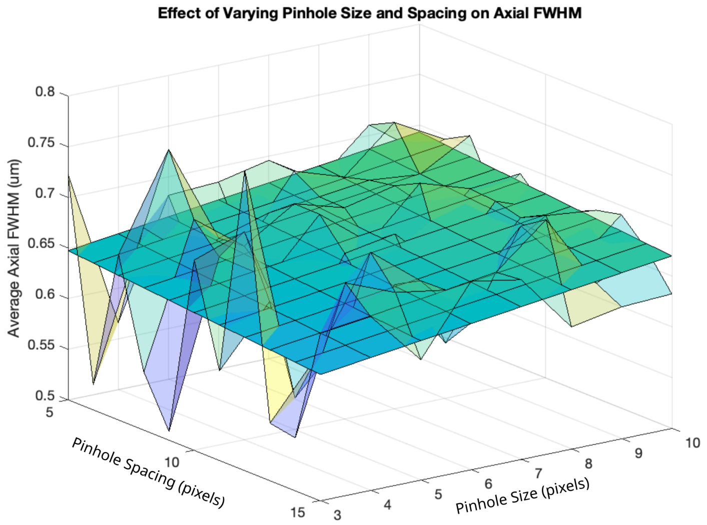


Figure 3.5.: Effect of varying pinhole size and spacing on the confocal sectioning ability of the CairnFocal. The faded surface shows the measured axial FWHM of the PSFs for a given pinhole size and spacing. The solid flat plane is a plane of best fit calculated using linear regression. The best fit plane slopes down to the left and front of the plot, indicating that optical sectioning improves when reducing pinhole size and increasing spacing. This relationship is unlikely to be linear over a large range of pinhole sizes and spacings and it's unclear if the noise on the FWHM measurement that's particularly noticeable with 3×3 pinholes is obscuring a deviation from linearity over the range of values measured (the increase in noise is due to difficulties defining the FWHM due to the high background present when using small pinhole sizes and large spacings). Assuming the plane is representative of the underlying processes, changing the pinhole size has $\sim 2.8\times$ the effect of changing the pinhole spacing. An interactive version of this plot may be generated using the MATLAB/Octave script in Appendix D.

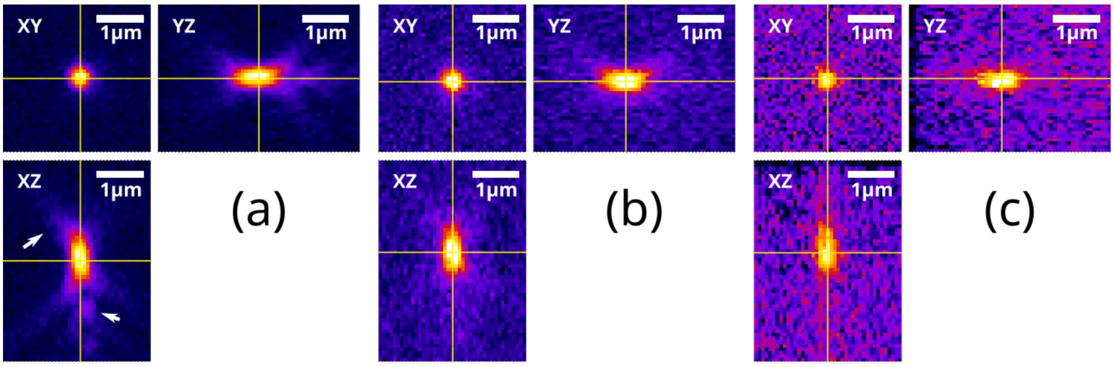


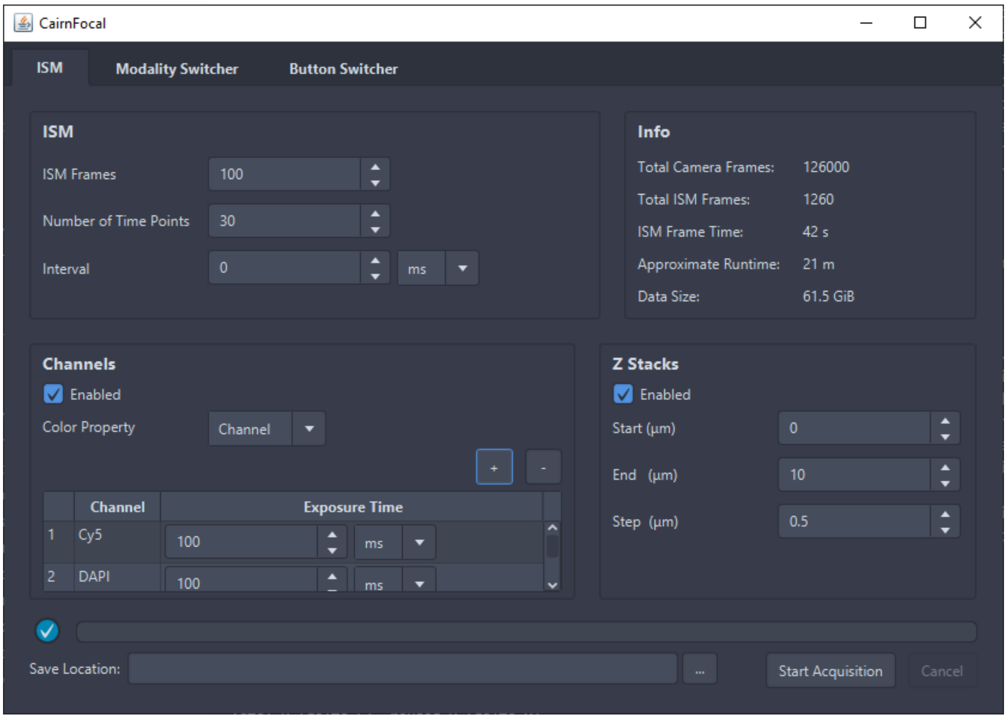
Figure 3.6.: Orthographic views of the system PSF measured using 200 nm beads and 640 nm illumination. The modalities used were (a) widefield; (b) Confocal, 8×8 pixel pinholes with 7 pixel spacing; (c) Confocal, 3×3 pixel pinholes with 15 pixel spacing. The out of focus airy rings visible in the widefield PSF are markedly reduced in the large pinhole confocal and aren't visible in small pinhole PSF, indicating both improve optical sectioning. The widefield PSF appears roughly symmetrical in the XY and YZ images, however, shows a clear asymmetry in the XZ plane, most noticeable in the regions marked with white arrows. Since modelling of the CairnFocal in Zemax has shown that it should be capable of achieving diffraction limited performance, this is a good indicator of misalignment in the system. Since they are the most difficult to align, this is most likely due to the mirrors in the correction optics being misaligned.

3.3.3. Image Scanning Microscopy

The proceeding section demonstrates that the system can perform confocal similarly to a traditional SDCM, in order to improve axial resolution. The system is also capable of performing a variant of ISM, similar to the Multi-focal Structured Illumination Microscopy (MSIM) technique outlined in [52], the key difference being that the "pinholing" is performed optically on the CairnFocal. The DMD patterns used when performing ISM are identical to those utilised in performing confocal, however, the synchronisation between the camera and the DMD is different. Every DMD image is displayed within a single camera exposure when performing confocal; ISM requires processing of the emission from each confocal illumination spot individually and as such one camera frame must be acquired per DMD image. Summing the images acquired during an ISM acquisition gives an approximation of the equivalent confocal image. If the image of each pinhole is shrunk by a factor of two around its centre before this summing (a process known as pixel reassignment [149]), then a lateral resolution enhancement of $\sim \sqrt{2}$ over that achievable with widefield is expected [48]. Appropriate deconvolution can then be applied to further increase the resolution enhancement to $\sim 2\times$ [52].

In principle, the system is capable of performing multidimensional ISM (multi-colour and z-stacks), however, µManager's MDA mode places restrictions on the order in which these dimensions can be traversed. While the colour and z dimensions can be exchanged (i.e., taking an entire z-stack in one colour before moving onto the next vs taking 3 different colour images before moving onto the next z-position), the time axis cannot. This means that there is no way to instruct µManager's MDA to, for example, take multiple camera frames before moving onto the next z-position. As ISM requires multiple camera images per processed ISM frame, µManager's MDA prevents multi-dimensional ISM imaging. To circumvent this restriction, a µManager plugin was developed to allow multi-dimensional ISM imaging. The plugin provides a GUI (Figure 3.7), similar to that provided by µManager's MDA but with the ability to specify the number of camera frames per ISM image. Unlike, MDA the user is required to specify the output file path. Each raw ISM frame is automatically saved to disk and removed from memory to allow for long term acquisitions where the volume of data would not fit in the computer's memory. Each ISM frame is saved to a separate TIFF file to make subsequent data processing easier.

In ISM, the image of each pinhole is scaled down by a factor of two around its centre before being summed to produce a single image with improved lateral resolution. The first step in processing, is finding the locations of the pinholes within the raw ISM images. Since the images of the pinholes resemble the images of individual fluorophores collected during SMLM, a simple and convenient method is to use SMLM reconstruction software to localise the pinholes (this work used the ThunderSTORM ImageJ plugin [131]). An ImageJ plugin was developed to perform the ISM reconstruction. The ISM reconstruction plugin first creates a new blank image of dimension $2MW \times 2MH$ where M is a user supplied magnification factor (typically $8\times$) and W and H are the width and height of the input images respectively. Each individual input image is scaled up by the magnification factor (i.e., to dimension $MW \times MH$) using bilinear interpolation. A square region of side length MR (where R is a user supplied value, used to determine the size of a pinhole in the camera image) surrounding each pinhole is added, element-wise, to an identically sized region at position $(2Mx_i, 2My_i)$ in the output image (where x_i and y_i are the x and y coordinates of the ith pinhole, as determined by the SMLM software). Since the output image has effectively been scaled by $2M$ and the input images only scaled by M, this is equivalent to reducing the size of each pinhole by a factor of two before summing. After summing the contributions from all pinholes, the output image is scaled down by a factor of M in order to produce a final image of dimension $2W \times 2H$. The plugin also



ISM

ISM Frames: 100

Number of Time Points: 30

Interval: 0 ms

Info

Total Camera Frames: 126000

Total ISM Frames: 1260

ISM Frame Time: 42 s

Approximate Runtime: 21 m

Data Size: 61.5 GiB

Channels

☑ Enabled

Color Property: Channel

	Channel	Exposure Time
1	Cy5	100 ms
2	DAPI	100 ms

Z Stacks

☑ Enabled

Start (µm): 0

End (µm): 10

Step (µm): 0.5

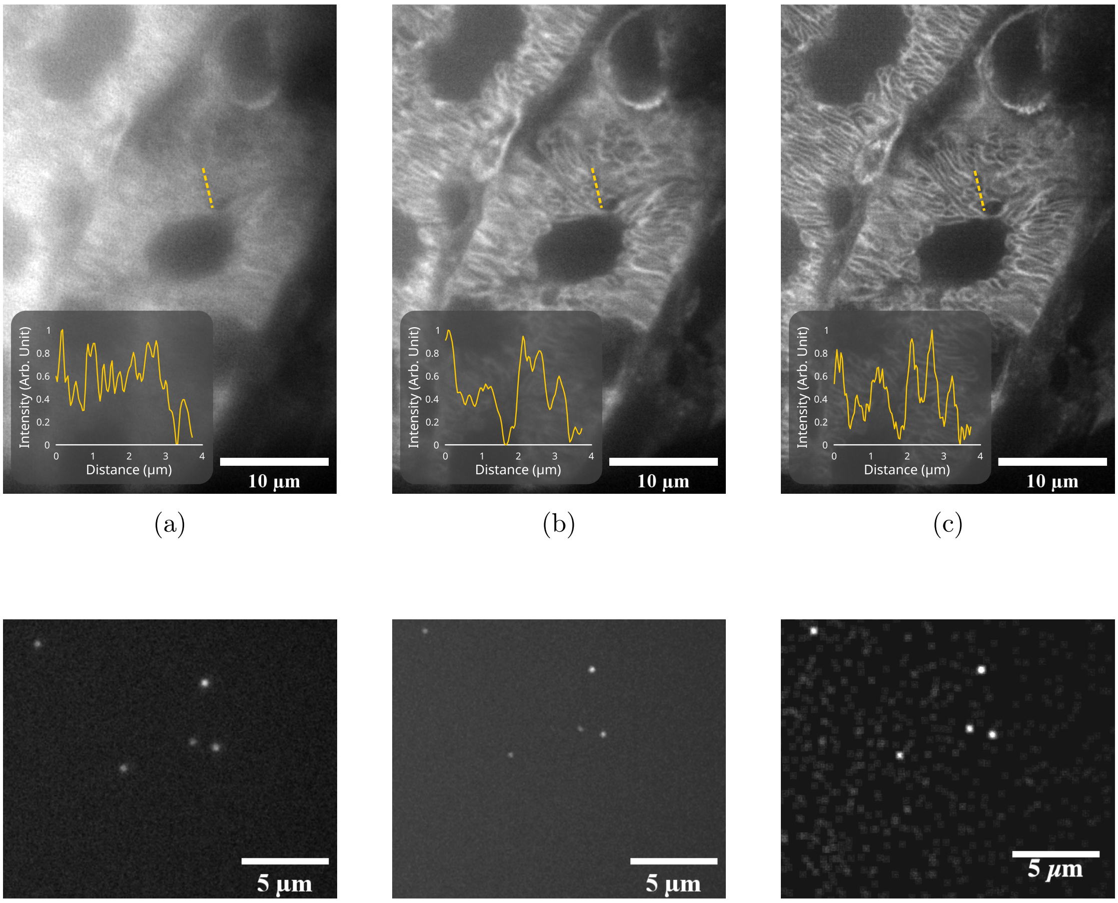
Save Location:

Figure 3.7.: The CairnFocal plugin ISM tab. This tab allows a user to perform multi-dimensional ISM acquisitions. First the user specifies the number of camera frames required to construct a single ISM image, the number of time points to acquire and the interval to wait between time point acquisitions. This tab doesn't connect to the CairnFocal Control program over the network interface so the number of camera frames per ISM image cannot be queried automatically. If enabled, the group of controls labelled 'Channels' allows the user to select the different µManager controlled colour channels to be used during imaging and set camera exposure times separately for each channel. Again, since this section of the plugin doesn't communicate over the network interface, controlling the colour channels using the FPGA interface is not directly supported. The 'Z Stacks' controls allow the user to specify the range of z stage positions to be used during acquisition. The top right 'Info' section provides some approximate information about how long the full acquisition will take and the final data size. The plugin saves the data for each ISM frame to a different folder to make subsequent processing easier. Data is streamed to disk to allow data sizes larger than available memory.

includes options to perform background adjustment, either by subtracting a constant value from each pinhole before summing, or by providing a background image that is used to subtract a local background from each pinhole image. In order to help in the selection of the R parameter, the plugin automatically displays the raw data with a box around each region that will be used.

Figure 3.8 shows a qualitative comparison between widefield, confocal and ISM (as described above). The top 3 images show the imaging techniques applied to a section of mouse kidney, the bottom 3 show the same techniques applied to 100nm beads. While the ISM has performed well on the mouse kidney, the bead data contains many square regions of non-zero values and the beads themselves also appear square. These artifacts are due to not being able to detect pinholes in regions of low fluorescence. The solution to this problem is to utilise knowledge of the images currently displayed on the DMD to predict the location of the images of the pinholes on the camera. Two techniques were investigated during this project. The first was to take a 'blank' ISM dataset using the same pinhole size and spacing as the actual data but using the microscope's transverse illumination. The ring illuminator illuminates the DMD sufficiently evenly and with sufficient intensity that the images of all the DMDs pinholes can be accurately detected using ThunderSTORM. Since the pinhole size and spacing is the same as in the actual ISM data, the pinholes localised in the blank dataset will also correspond to the locations of the pinholes in the actual dataset, even if those pinholes are not visible.

The second technique was to create a mapping between the DMD and the camera; knowledge of the *on* pixels displayed on the DMD can then be used to predict the location that would be detected by ThunderSTORM in the camera images. As discussed in Chapter 2, the principal difficulty in deriving such a calibration is that the non-linear nature of the transform caused by misalignments in the CairnFocal's correction optics and the 1.5× CMount magnifier in front of the camera. A secondary difficulty that presents itself is that, generally, only parts of the DMD are visible on the camera, meaning that in a typical blank ISM dataset, not all pinholes displayed on the DMD are visible in the camera images. The following protocol for producing a calibration dataset was developed. Firstly, the approximate bounds of the camera in DMD space are manually determined by displaying a rectangle on the DMD using the CairnFocal Control program. The user changes the size of the rectangle such that it is as large as possible while still entirely visible in the camera image. The size and position of the rectangle is then used in the creation of a set of calibration images. The calibration dataset consists of several images (typically 1000) where only a single, randomly selected, DMD pixel is in the *on* position. The rectangle determined in the first step is used to ensure that the enabled DMD pixel is



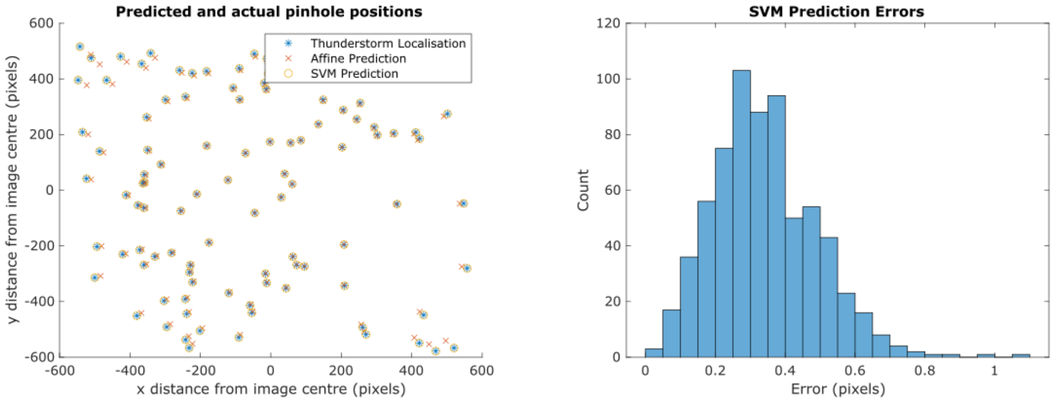
(a) (b) (c)

(d) (e) (f)

Figure 3.8.: Comparison of widefield, confocal, and ISM on mouse kidney, (a)-(c), and 100 nm beads, (d)-(f). When imaging the mouse kidney, widefield, (a), performs poorly, with very little detail visible within the sample due to out of focus light reaching the camera and scattering of light as it passes through the thick sample. Confocal (3 × 3 pinholes, 15 spacing in this case), (b), performs better, allowing significantly more detail to be resolved by rejecting out of focus light. ISM, (c), maintains the optical sectioning ability of the SDCM emulation, with the added benefit of improved lateral resolution and SNR improvement. The improvements associated with confocal and ISM are less apparent on the beads data, however. The small axial extent of the 100 nm beads and their lateral sparsity removes all benefits of using confocal, and as a result the confocal data (e) is of comparable quality to the widefield data (d), but with lower contrast. The ISM data, (f), is significantly worse, consisting of randomly placed square patches of non-zero pixels in regions where ThunderSTORM detected background noise as a pinhole location. This effect can be lessened by optimising ThunderSTORM's settings, however, this can cause it to miss pinholes that correspond to parts of the beads, a full solution requires the calibration and background correction processes described later in this chapter.

guaranteed to always be visible in the camera image. The location of each random pixel, in DMD space, is written to a CSV file. A single camera image is taken of each randomly selected DMD pixel using the ring illuminator and the location of each of the pixels in camera space is determined via ThunderSTORM. A calibration can then be derived by comparing the location of the pixel enabled on the DMD to the location detected in the camera image. This calibration will locate the centre of the DMD pixel which is assumed to be the location about which the pinhole image should be shrunk. This is slightly different to the result when using ThunderSTORM on the raw ISM data, where fluorophore density variations and sample aberrations may shift the detected pinhole location slightly.

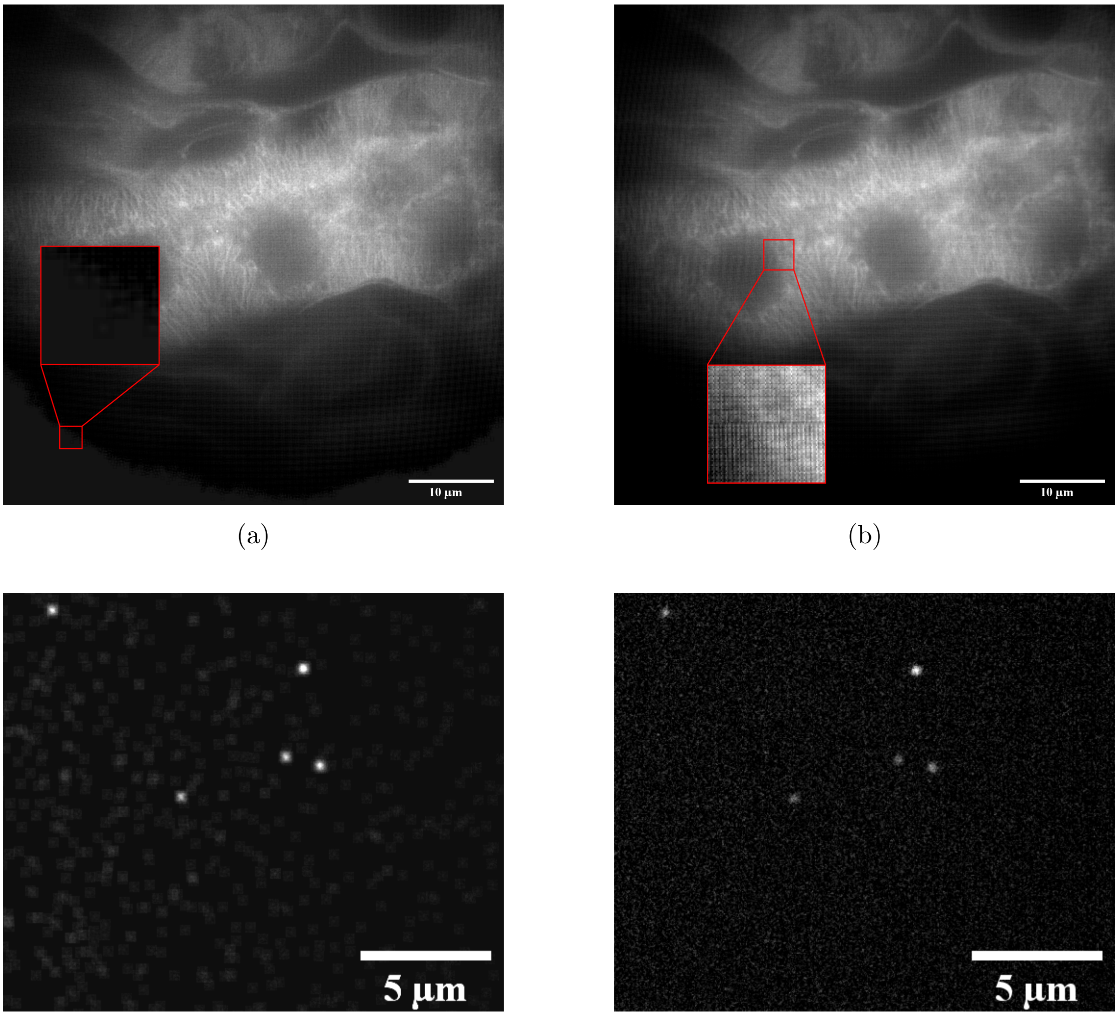
A MATLAB script to analyse these datasets and produce a calibration was developed. The calibration process is as follows: First, any frames where multiple ThunderSTORM pinholes were detected are discarded. Of the remaining pixel location pairs, some are held back to be used as validation points. Over sufficiently small regions, the mapping between the DMD and the camera is approximately linear. All pixel pairs that were detected within the centre 200 pixels of the image are used to calculate an approximate affine transform between the DMD and the camera. This affine transform is used to map all the DMD space locations into camera space (to a first approximation). Once approximately mapped into camera space, all the locations are converted into polar coordinates about the centre of the camera image. Since the majority of the distortions are radial, the θ value calculated by the affine transform should be approximately correct. Any location pairs for which the θ value of the affine-predicted location and the ThunderSTORM detected location differ by a large amount ($> 1\,\mathrm{rad}$) are assumed to have been incrorrectly localised and are discarded from further analysis. Two Support Vector Machines (SVMs) using 5th order polynomial kernels are then trained to map from the affine-predicted locations to the detected ThunderSTORM localisations. The SVMs are trained using MATLAB's `fitrsvm` function. The first SVM is trained to use the r and θ values of the affine-predicted locations to predict the r value of the ThunderSTORM localisation and the second was trained to use the same inputs to predict the θ values. Before training, the SVM inputs are duplicated twice, once with the θ values incremented by 2π and once with θ reduced by the same amount in order to reduce the effect of the discontinuity in θ as it transitions from π to $-\pi$ on the final output. The SVM predictions are worse than the affine transform predictions for small values of r so in the region $r < 50$pixels, a weighted average of the affine prediction and SVM prediction is used as the final output value. The weighting is adjusted such that as r increases, the SVMs prediction gets a higher weighting, in order to prevent a noticeable discontinuity at $r = 50\,\text{pixels}$. Figure 3.9 shows the accuracy of the predicted pinhole locations at several points during the calibration procedure on a typical calibration dataset and Figure 3.10 shows a qualitative comparison between the ISM reconstruction with and without calibration.



(a) (b)

Figure 3.9.: Accuracy of the SVM based calibration routine. (a) A plot of a random selection of pinhole locations as localised by ThunderSTORM, predicted by the linear affine transform and the full SVM based calibration routine. The CMount magnifier is the dominant source of the distortions, introducing pincushion distortions, which are caused by the magnification of the system varying with distance from the optical axis. As this is a radial distortion, the affine transform is accurate in the centre of the image and performs worse as distance from the centre increases. The SVM accounts for these distortions and produces predictions that are sufficiently accurate so as to be indistinguishable from the positions of the ThunderSTORM localisations at this scale. (b) A histogram of the errors associated with each pinhole prediction made by the full calibration routine. The majority of predictions are within 0.5 pixels of the true pinhole location, with only one prediction being out by more than 1 pixel.

To quantitatively assess the performance of the ISM procedure, the 200 nm beads used in the previous sections were again imaged. After ISM reconstruction, the lateral FWHM of the PSF was reduced to 227 nm ± 30 nm vs 314 nm ± 8 nm for widefield, a resolution improvement of $\sim 1.41\times$. To investigate the effect of deconvolution, 100 nm beads were imaged. In widefield, the FWHM of the lateral PSF was measured to be 315 nm ± 25 nm, ISM (without deconvolution) brought this down to 256 nm ± 32 nm. A PSF was generated using the Born and Wolf 3D optical model ($\lambda = 680$ nm), using the PSFGenerator plugin [130], and used for deconvolution in the Richardson-Lucy iterative deconvolution algorithm, via the DeconvolutionLab2 plugin [28]. After deconvolution,



(a)

(b)

(c)

(d)

Figure 3.10.: A comparison between the images achieved when reconstructing using the ThunderSTORM based localisation method, (a) and (c), versus using SVM calibration, (b), and 'blank' calibration (d). As described previously, the ThunderSTORM localisation method performs well in regions of relatively high fluorescence, but poorly in regions of low fluorescence. In (a), this manifests as a 'fuzzy' transition from the dark regions close to the bottom of the image to the grey region at the very bottom of the image where all the values are exactly 0. The reconstruction performs well throughout the rest of the image, however. Using SVM calibration, all pinholes are included in the reconstruction exactly once so this effect does not occur. Zooming in on the region shown in (b) reveals an artifact introduced by the SVM calibration along the line $\theta = \pi$. (c) is copied from Figure 3.8f. (d) shows that by using the blank dataset method for determining pinhole positions and applying local background correction the square artifacts present in (c) are completely removed.

the PSF was measured to be 178 nm ± 33 nm, a total increase in resolution of ∼ 1.77× that achieved with widefield. Figure 3.11 shows a comparison between the resolutions achieved with widefield, confocal, ISM and ISM with deconvolution.

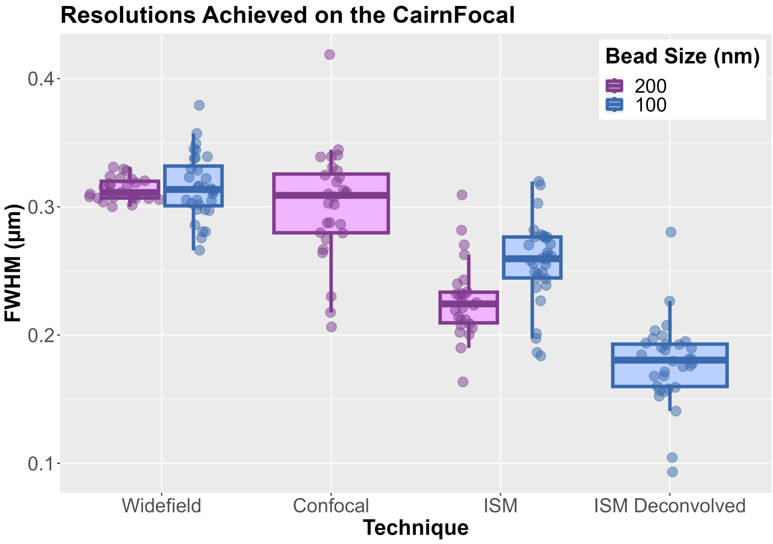


Figure 3.11.: Lateral resolutions achieved on the CairnFocal platform as measured on 200 nm and 100 nm beads. The values shown are from measurements in both the x and y directions. As expected, in widefield the average resolutions calculated using the 100 nm and 200 nm beads were almost identical, though it should be noted that the standard deviation of the 100 nm data was significantly larger, owing to the lower contrast-to-noise ratio associated with the smaller bead sizes (20.47 dB vs 12.35 dB). The confocal data shown is for a pinhole size of 3 pixels and spacing of 15 pixels. The confocal data shows a slight resolution improvement over the widefield (∼ 1.05×), though with markedly more variation due to difficulties detecting PSFs over the high backgrounds present. The ISM saw a larger resolution improvement, roughly in line with the predicted ∼ 1.41× better than widefield, with a larger resolution improvement seen on the 200 nm beads than the 100 nm. Deconvolution further reduced the size of the PSF to ∼ 1.77× better than widefield. Confocal data was not collected on the 100 nm beads due to high background causing difficulties with imaging. Deconvolution was not performed on the 200 nm bead data since the resolution achieved without deconvolution was approaching the size of the beads. An analysis of the statistical significance of these results is given in Appendix E.

3.4. Discussion

3.4.1. Widefield

Despite being the simplest of the 3 techniques performed, the widefield data (particularly that of the beads) demonstrates some the key challenges of using a DMD in this configuration. Firstly, the PSFs measured are larger than expected, ∼310 nm vs the ∼280 nm predicted by the Rayleigh criterion. This implies that the system is still somewhat aberrated, most likely due to misalignments in the correction optics. The existence of aberrations is also supported by the slight asymmetry in the PSF as the bead is moved in z, visible in Figure 3.6. While optical modelling performed in Zemax during the CairnFocal's development at Cairn Research indicates that the system should be capable of achieving diffraction limited performance when properly aligned, in reality the alignment procedure is not sufficiently well understood to reliably achieve optimal alignment. The difficulties in alignment are threefold: Firstly, the search space is large. In each of the two symmetric optical paths there are 4 mirrors, each with 3 degrees of freedom, the two curved mirrors in each path additionally have a course adjustment, and the CMount connectors onto which the cameras are attached contain a lens which can be moved relative to the camera for a total of 15 degrees of freedom per side. Secondly, no method has currently been developed to allow the position of individual components to be set independently of the other optical components (with the exception of the CMount lens which can, theoretically used to focus the camera at infinity). This means that an iterative alignment procedure is required to consider, at minimum, the 12 degrees of freedom associated with the fine controls on the 4 mirrors. To make matters worse, it's not immediately clear what the effect of misalignment of individual components would be. As an example, one might expect that changing the angle of the flat mirror closest to the DMD would simply move the image of the DMD left and right on the camera, however, in reality this also slightly changes the angle into the pair of curved mirrors, often introducing astigmatism which must be corrected by adjusting all three of the remaining mirrors appropriately. The final issue is that even when both sides of the CairnFocal have been aligned to give apparently close to optimal images individually, they seldom overlap one another adequately when digitally overlaid on top of one another. This is most likely due to reaching different locally optimal points in the optimisation spaces of each side, which are qualitatively similar. This causes issues for any techniques that attempt to use both sides simultaneously. Currently such techniques must rely heavily on a calibration between the two cameras, which as discussed in section 3.3.3, is a highly non-linear transformation that is difficult to model accurately.

The 1.5× CMount magnifier utilised alongside the 95B also likely contributes to a degradation in resolution. It was introduced to allow the 95B's 11 µm pixels to approximately reach the Nyquist sampling frequency when using a 60× objective and to comfortably surpass it with a 100× objective, however, it also introduces a considerable amount of pincushion distortion to the image (presumably it wasn't designed to support an ∼18.5 mm diagonal chip). When pincushion distortion is present, the magnification of the system will increase as the distance from the centre of the image increases making PSFs further from the centre appear larger. If a constant magnification of 1.5× across the field is assumed, this will tend to underestimate the resolution of the system. The prevalence of this effect is reduced, however, due to difficulties illuminating large areas of the DMD restricting the usable field size to regions relatively close to the centre of the image for fluorescence imaging.

As alluded to in the previous paragraph, illumination of the DMD is another area that causes difficulties when using the CairnFocal. Ideally, the DMD should be evenly and critically illuminated across the entire visible portion of the DMD. If the illuminated area on the DMD is too small then the field of view is restricted, if it is too large then the power density delivered to the sample will drop, reducing the maximum possible SNR. If the illumination isn't critical, then the shape of the illumination volume is changed to the detriment of confocal modalities. The DMD consists of 1024 × 768 mirrors of approximately 13.7 µm pixels, a total size of approximately 14.0 mm × 10.5 mm. The 95B has a square sensor of width 1200 pixels, with a pixel size of 11 µm; When the 1.5× CMount magnifier in place, this is equivalent to a square region of side length 8.8 mm in DMD space. The factor limiting the field size is therefore the sensor size. The end of the light guide used to couple the output of the LDI into the CairnFocal is circular, with a diameter of 1.5 mm. Since the light guide is circular, an ideal illumination scheme is not possible, either the field must be restricted to less than the area of the camera chip (prioritising power density at the sample) or the illumination must overfill the camera's field of view (maximising the size of the available imaging field). For the former illumination scheme, the required magnification between the end of the light guide and the DMD 5.87×, whereas for the latter, the required magnification is 8.30×. Using a 35.0 mm focal length lens, the illumination spot was measured to be approximately 7 mm on the DMD, implying a magnification of $\sim 4.67\times$ and that the curved mirrors are acting as a focusing element with an approximate effective focal length of 163 mm (close to the theoretical 166 mm used in [98]). Therefore, to under-fill the field optimally, a lens of focal length 27.3 mm is required, to overfill the required focal length is 19.3 mm. These focal lengths aren't unreasonable, suitable lenses to meet these criteria are available from retailers such as Thorlabs. It should be noted, that a different choice of camera, DMD to camera magnification or light guide would lead to different requirements for this lens. For example, a natural suggestion would be to replace the circular light guide with a square ended fibre in order to remove the trade-off between field size and power density, however, optical fibres typically have smaller cross sections than light guides. Thorlabs sells a single square ended fibre with a 150 µm width. Utilising this for illumination would require a magnification of 58.7× and a much more strenuous focal length of 2.73 mm. It should be noted that some manufacturers will supply larger square fibres, however, often only in quantities that would prohibit purchase by a single lab. The 35 mm lens used in this project was selected primarily because it was the shortest focal length lens on hand.

Construction of the illumination system revealed a measurement that could potentially help diagnose alignment inaccuracies within the correction path. Since the two curved mirrors act as a single focusing element with focal point placed on the surface of the DMD, the image of the DMD should be focused at infinity between the final curved mirror and the lens in front of the camera (this is what allows the filter cube to be placed at this location without introducing astigmatism into the final image). By a similar argument, the image of the end of the light guide should be focused at infinity by its collimating lens, in order to form an image of the end of the light guide on the DMD. To ensure this was the case, before attaching the illumination system to the CairnFocal, the position of the lens relative to the end of the light guide was adjusted so that an image of the end of the light guide was formed on a distant wall, however, upon attaching the illumination system to the CairnFocal, the image of the light guide was out of focus; the position of the lens required considerable further adjustment to bring the image into focus. This provides some evidence to suggest that, in their current positions, the mirrors are not acting to focus the image of the DMD at infinity (a hypothesis further supported by the fact that adding additional filters behind those already in the system introduced significant astigmatism as will be discussed later in this chapter). If this is the case, then it suggests that as well as attempting to bring an image of the DMD onto the camera after fixing the position of its lens, it may also be useful to simultaneously adjust so that an image of the light guide is focused onto the DMD. This observation also brings into question whether the magnification between the DMD and the camera really is unity (or 1.5× if the CMount magnifier is in place). While this question would need to be answered conclusively before making any absolute measurements of the size of imaged structures, the primary conclusions of this chapter, i.e., the relative performance of the implementations of widefield, confocal and ISM still hold even if the exact magnification is not as assumed.

	Laser Power (mW)		
Laser Wavelength (nm)	After Filter Cube	At Objective	Efficiency
405	0.961	0.070	7.28%
470	1.11	0.134	12.07%
555	5.26	0.543	10.32%
640	1.96	0.198	10.10%

Table 3.1.: CairnFocal Transmission Efficiencies

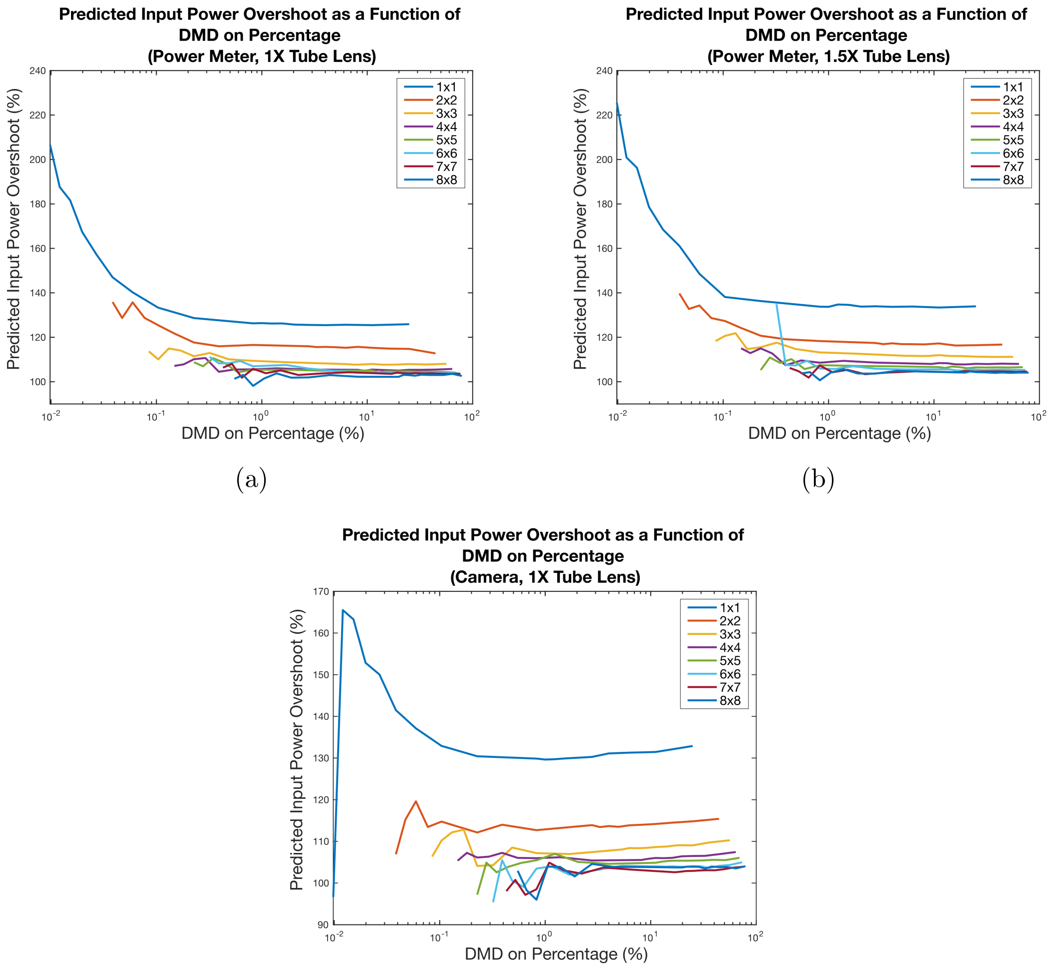
Another question that needs to be answered is how the CairnFocal performs as a SLM, poor SLM performance will result in poor imaging performance. As DMDs act as 2D diffraction gratings, incident light will not simply be reflected, as with a mirror, but instead diffracted into a number of orders, each emerging at different angles. Since any subsequent optical system will have a finite acceptance angle, diffraction gratings will inevitably suffer poorer efficiency than simple reflecting elements. Measurements of total laser power were taken at the output of the filter cube and the back aperture of the objective to determine transmission efficiency. The results, shown in Table 3.1, indicate that the system is only $\sim$ 10% across the visible range. This reduces the maximum available illumination intensity which could cause problems when imaging dim samples, however, in practice this is more than compensated for by the high power lasers provided by the LDI in most situations. It should be noted that these efficiencies don't necessarily reflect the collection efficiency of the system, as the large extent of the curved mirrors is used to capture more diffracted orders than the relay optics between the DMD and the microscope. The design of the correction optics is such that the large mirrors are capable of capturing multiple diffracted orders in order to recombine them to improve efficiency [102].

The second DMD parameter that may cause issues for some imaging modalities is contrast ratio, i.e., the ratio between the power transmitted by an *on* pixel vs an *off* pixel. Using a similar setup to that which was used for measuring the transmission efficiency, the Full-On, Full-Off (FOFO) contrast ratio at the back aperture of the objective was measured by comparing the measured value with all DMD pixels in the *on* state to the value when all the DMD pixels were in the DMD *off* state. The measured values were 16.3 mW for the *on* state vs 1.5 µW for the *off* state; a contrast ratio of $\sim 10000 : 1$. While 10000:1 seems to be higher than typically reported, there have been some reports of contrast ratios as high as 48250:1 [150]. It should also be noted that typically these measurements are based on using DMDs in commercial wall projector units, it's possible that the higher quality optical path typical in a modern research microscope provides a better contrast ratio than that which is typically achievable in commercial projector units.

While simple to calculate, the FOFO contrast ratio doesn't tell the whole story since the DMD is very rarely used in the all *off* position. In confocal, the DMD is used to illuminate small regions. As the pinhole size and spacing is varied, the structure of the diffraction grating seen by the illumination light changes and so too will the transmission properties of the DMD. Anecdotally, the contrast of smaller pinholes has appeared to be lower than that of larger pinholes when looking at illumination on the system's back camera. To investigate this possibility, the power meter was once again placed at the back aperture of the objective. The total input power was measured with the DMD pixels all in the *on* position. Power at the back aperture was then measured when using a DMD image of static pinholes with varying pinhole sizes and spaces. In order to compare the power measurements, they were divided by the percentage of the DMD that was in the *on* position when the measurement was taken. If the DMD behaved ideally, then every *on* pixel would transmit 100% of the power incident upon it to the sample and each *off* pixel would transmit 0%. In this case, dividing by the percentage of the DMD in the *on* position would result in the total input power measured previously. Any deviation from this value would therefore indicate imperfection in the transmission percentages and by extension the contrast ratio. Figures 3.12a and 3.12b show the results when using the 1× and 1.5× tube lenses in the microscope frame respectively. In both cases the powers detected are higher than predicted by the ideal DMD model. Since an *on* pixel can't transmit more than 100% of the power incident upon it, this must be due to off pixels transmitting some light to the sample and an effectively lower contrast ratio. As predicted, by this metric, the smaller pinholes perform worse, with the measured powers tending towards the ideal prediction as the pinhole size is increased. What was unexpected, however, was that the effective contrast ratio appears to decrease rapidly as the smaller pinholes are moved further apart. To rule out non-linearities in the detector, the power meter was replaced by a camera in front of a 10× air objective and the measurements repeated. Figure 3.12c shows the results with the 1× tube lens. The decrease in contrast ratio for small pinholes with large spacings is seen once again, however, the value for the largest spacings drops to below 100% of the total input power at the largest spacing, perhaps indicating the effect is a result of inadequacies in the chosen analysis method rather than a physical effect.

3.4.2. Confocal

The data shown in this chapter demonstrates an improvement in optical sectioning by making the CairnFocal emulate a SDCM. As with lateral resolution, the axial resolution in widefield is slightly worse than predicted by the Abbe diffraction limit at 680 nm (613 nm predicted vs 752 nm ± 28 nm measured). Ideally, confocal should improve the z



(a)

(b)

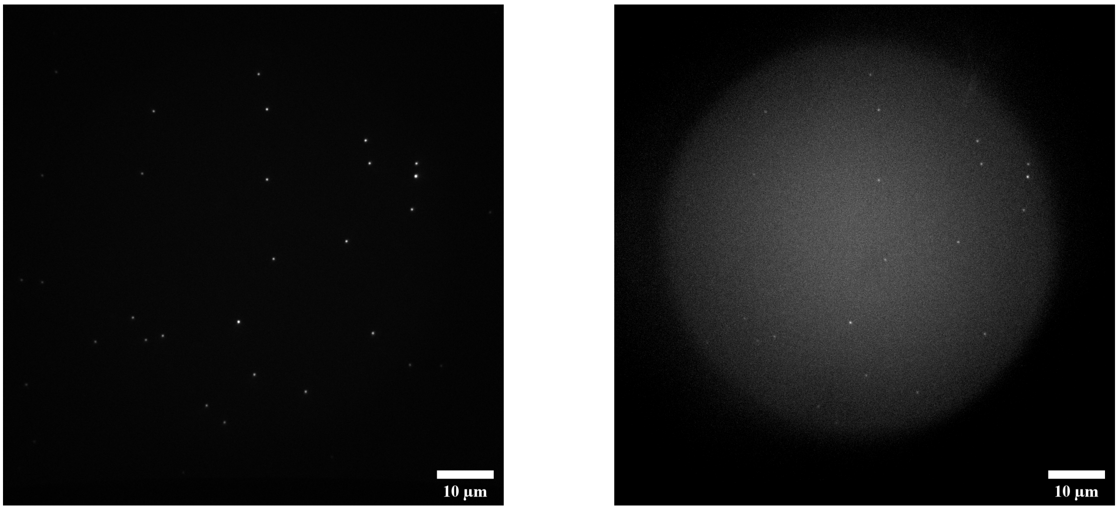
(c)

Figure 3.12.: Plots of excess light passed though the DMD, using a power meter, (a) & (b), and a camera (c) as detectors. All plots indicate an increase in the amount of excess light transmitted by the DMD when using smaller pinhole sizes, indicating the contrast ratio of the DMD is lower for smaller structures. The increase seen in the smaller pinhole plots as the on percentage decreases is as yet unexplained.

resolution by $\sim 1.43\times$ or to $\sim 525\,\text{nm}$, given the measured widefield z resolution [129]. While some of the settings resulted in improvements comparable to this limit (e.g., 3×3 pinholes with 9 pixel spacing measured a resolution of $509\,\text{nm} \pm 62\,\text{nm}$), the linear fit through the data indicated that the best resolution achieved was 624 nm, an increase of only $1.21\times$. While it is possible that this indicates less than optimal confocal performance, several difficulties in collecting the data could also go some way to explaining the apparently poor performance.

The primary difficulty in collecting the data was that of high background. Figure 3.13 shows a comparison between widefield and confocal beads data. While the beads are clearly visible in the widefield data, they are much harder to make out in the confocal data. The region illuminated on the DMD is also clearly visible, even in regions where there are no fluorophores. The background is a particular issue when small pinholes or large spacings are used (i.e., settings that should give better confocal sectioning). The reason for this is that, while an *on* pixel will primarily contribute signal if it corresponds to a region of the field with fluorescent sample, an *off* pixel will always contribute background due to illumination light reflecting back to the camera from the DMD. For a given exposure time, smaller pinhole sizes or larger spacings will result in any given pixel spending proportionally more time in the *off* position and contributing a higher background. Since both the background and the signal are proportional to the exposure time and illumination intensity, increasing either of these (which would generally improve SNR in widefield images) will not improve image quality when the background is the limiting factor. The low signal-to-background ratio in some images made it difficult for the automated analysis to correctly define the z resolution, which manifests as noise in Figure 3.5. This is also why the pinhole sizes explored in Figure 3.5 don't extended down to 2×2 and 1×1 pinholes; the signal-to-background ratio was too low to accurately define the z resolution.

One potential explanation for the visibility of the illumination region was that an optical component in the system was fluorescing under the high intensity illumination provided by the LDI. Since the illumination region appeared in focus, the fluorescing optical component would need to be placed in an image plane. The only optical component placed in an image plane is the DMD so this seemed the most likely to be causing the fluorescence, however, investigations with a spectrometer yielded no evidence of fluorescence. The fluorescence theory was made even less likely by the discovery that including multiple emission filters reduced the background significantly; if the background was caused by fluorescence then it would be expected that it would continue through multiple emission filters largely unattenuated. Unfortunately, the additional emission filters



(a)

(b)

Figure 3.13.: Example images of the same field of beads taken in widefield, (a), and confocal with 3 pixel pinholes, 15 pixel spacing, (b). Camera exposure and laser power were adjusted to provide the clearest image of the beads in both images. The confocal data is much lower contrast than the widefield, with the illumination spot clearly visible. Since the visibility of the laser spot is most likely laser light getting though the filters, simply increasing the exposure time will fail to increase the contrast of the resulting images.

also introduced significant astigmatism, likely because misalignment of the correction mirrors meant they were not placed in infinity space. More careful optimisation of both the excitation, emission and dichroic filters to better match the excitation and emission wavelengths of the beads allowed them to be imaged under the majority of settings of interest.

Another explanation for the visibility of the illumination region is that the excitation filter is not preventing all of the excitation light from reaching the camera. This is perhaps unsurprising, in a typical epi-fluorescent widefield microscope or CLSM, the majority of the illumination light that isn't absorbed by the fluorophores in the sample continues travelling out and away from the objective, with only scattered and reflected light making it back to the filters. In a SDCM and by extension the CairnFocal, the illumination light that is rejected by the pinhole array (or DMD) will be scattered in all directions, with a significant proportion travelling in the direction of the emission filters. Given this and the high laser powers typically required for SDCMs without matched microlens arrays, it's unsurprising that the inability for emission filters to block all light at unwanted wavelengths would be more apparent in these systems. This explanation also seems plausible since a similar background effect was seen during development of a SDCM that Cairn Research had collaborated on. The solution in that case was to place an adjustable aperture stop in a field plane in front of the camera. This allowed for the aperture to be closed down to reject light arriving at such an angle that it couldn't have originated from the objective and therefore must have been reflected from the pinhole array. In principal, a similar approach could be taken with the CairnFocal, however, this is yet to be attempted. Other SDCM implementations have deliberately tilted the disk so that the majority of the light is reflected away from the filters [44, p.229]. This also reveals an interesting advantage of the Yokogawa spinning disk system over SDCMs that don't have matched pinhole arrays that isn't often mentioned. Since the dichroic in the Yokogawa system sits between the two disks, the illumination light rejected by the pinhole array isn't scattered towards the camera, thus reducing the intensity on the emission filter to levels comparable to that expected in a widefield or CLSM system, and presumably reducing the probability of encountering background issues.

While difficulty in accurately determining the z resolutions achieved due to high background when using confocal patterns is probably a major contributing factor to the worse than expected improvement, it should also be noted that over extended durations ($\sim$ 10 min) considerable focus drift was apparent when not making use of the Nikon Perfect Focus system within the microscope body. When taking z stacks, Perfect Focus needed to be disabled and so it's possible that drift during acquisition has caused the distance between z planes in the output images to be inconsistent. Since smaller pinhole sizes and wider spacings between pinholes generally require longer exposure times, the effect of drift would be larger on data collected with these settings.

The plane fitted to the z resolution data shown in Figure 3.5 indicates that, on the CairnFocal, changing the size of the pinholes has a larger effect on the optical sectioning ability of the microscope than moving the pinholes further apart. To maintain optical sectioning performance after opening the pinhole by 1 pixel, the spacing between pinholes must be increased by 3 pixels. This conclusion is premised on the idea that the effects of both pinhole size and pinhole spacing on optical sectioning performance are linear. While this will be approximately true for a sufficiently small range of pinhole size and spacing values, it will not hold in general (this would imply that with a pinhole size of 8×8 when using the 100× objective lens, the system would achieve close to perfect optical sectioning, i.e., a 0 nm FWHM, with a spacing of 300 pixels, which is clearly impossible). Over the range of values tested, the fit appears to be subjectively plausible, however, it is possible that the noise that is particularly prominent on the 3×3 pixel data is masking the transition to non-linearity, which could also go some way to explaining the worse than expected performance of the confocal.

It's worth briefly mentioning work by the Jovin group done on a PAM system that formed the initial basis for the initial design of CairnFocal [106]. In this work, patterns were used where approximately 50% of the DMD pixels, selected at random, were placed in the *on* position. Data were collected on both the *on* and *off* cameras simultaneously and post processing was applied in order to reconstruct the equivalent conjugate and non-conjugate images. As mentioned in Section 3.4.1, aligning the *on* and *off* side images on the CairnFocal such that the two images could be directly subtracted or summed presents significant difficulties. Performing this post processing would therefore require a calibration between each of the two camera, probably using techniques similar to those outlined in Section 3.3.3. It's not clear, however, if this technique provides significant improvements over the standard SDCM emulation performed in this work.

3.4.3. Image Scanning Microscopy

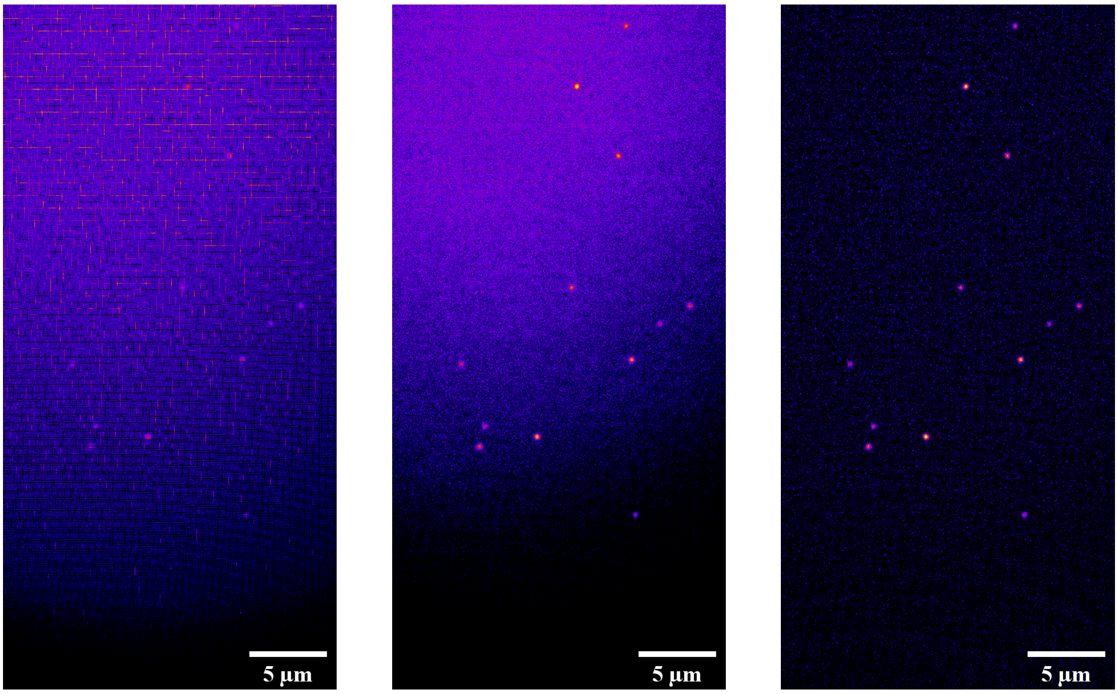
While the SDCM emulation performed described previously does provide a small improvement in lateral resolution, the improvement is so small that confocal is rarely described as a super-resolution technique. Section 3.3.3 outlined the results of implementing a version of the ISM super-resolution technique on the CairnFocal. ISM doesn't provide the dramatic resolution enhancements that are typically associated with other super-resolution techniques such as SMLM or STED [48], [63], [151]. Being a variant of SIM, resolution enhancements are limited to 2× the diffraction limit or approximately 120 nm vs the single digit nanometre resolutions that have been reported with SMLM [59]. For this project, the major advantage of SIM based techniques is that no special sample preparation is required, allowing the techniques to be used on any fluorescently labelled sample. ISM was chosen over traditional Gustafsson SIM as it plays to the strengths of the CairnFocal due to its ability to emulate SDCM. The DMD can be used to create the sinusoidal illumination patterns required for Gustafsson SIM, however, the presence of the DMD in the emission path interferes with the data collection, pushing some of the data to the *on* side and some to the *off* side camera. In theory it would be possible to reconstruct the SIM data from the images on the two cameras but this suffers from the same alignment problems as the Jovin confocal equivalent mentioned at the end of the previous section. It would be easier to collect the data using a camera on the back port of the microscope frame, however, in practice, there have been problems ensuring that the back port camera is conjugate with the DMD. Since the CairnFocal could already emulate a SDCM, implementing the data collection for ISM was simply a case of changing the synchronisation between the DMD and the camera.

This work has demonstrated the CairnFocal is capable of achieving a total resolution enhancement of $\sim 1.77\times$, roughly in line with values reported in the literature. The achieved resolution enhancement is slightly lower than the $2\times$ reported in the paper on the most similar technique, MSIM [52], slightly more than that achieved in the original ISM paper ($1.63\times$) and roughly in line with the theoretical maximum of the Fourier filtering approach quoted [48], and slightly more than the $1.7\times$ achievable on the Zeiss Airyscan [14]. Despite the differing processing steps, all three of these techniques and the original paper that ISM was based on [41] all agree that pixel reassignment without deconvolution should provide a resolution enhancement of $\sqrt{2}\times$ ($\sim 1.41\times$) and this was achieved on the CairnFocal when imaging the 200 nm beads. When imaging the 100 nm beads, the CairnFocal fell slightly short of this ideal, managing only a $\sim 1.23\times$ increase in resolution vs widefield. It was expected that the CairnFocal would be able to achieve the same resolution enhancement as MSIM given the similarities between the two techniques, failing to reach the $\sqrt{2}\times$ resolution enhancement from pixel reassignment is likely to have been a contributing factor to not matching MSIMs performance. The difference in resolution achieved by the three techniques mentioned above is primarily down to the final processing step, deconvolution. Fourier filtering was used in the ISM variant presented in [48], using a filter derived from a combination of the excitation and emission PSFs, whereas MSIM utilised Richardson-Lucy deconvolution using a measured PSF [52]. It's not clear what processing is performed on the Airyscan, [14] only makes reference to performing '... a linear deconvolution', but it is likely some variant of a Weiner filter using an estimated PSF. On the CairnFocal, a 680 nm PSF was used in Richardson-Lucy deconvolution as this improved the resolution without introducing noticeable distortions to the images. It's likely that using a measured PSF as in MSIM would have performed better. One potential difficulty with this is that the distortions introduced by the correction optics and the $1.5\times$ magnifier mean that the PSF is not constant across the field. In this project, 2D deconvolution was performed on each z plane individually due to difficulties in taking z-stacks on the system. Zeiss report an isotropic resolution of 140 nm achieved by instead performing a 3D deconvolution so it's possible that similar improvements over confocal's optical sectioning could be achieved on the CairnFocal by utilising a 3D deconvolution.

Another reason why the final resolution when measured on the 100 nm beads was lower than expected is that the resolution after pixel reassignment was also lower than expected. Oddly, pixel reassignment achieved almost exactly the expected resolution enhancement when imaging the 200 nm beads. While this is potentially due to the lower signal levels produced by the 100 nm beads, the microscope frame's $1.5\times$ tube lens was accidentally used when imaging the 200 nm beads but the 1× tube lens was used for the 100 nm beads. In the mathematical treatment of ISM in [41] and [48], the excitation spot is assumed to be scanned continuously through the sample (i.e., not at discrete steps). In [48] the illumination spot was generated using a normal laser scanner, theoretically capable of scanning the beam continuously (though the paper doesn't say what the step size actually used was). MSIM used a DMD with a smaller pixel size (10.8 µm) than that used on the CairnFocal with a 90× magnification between the DMD and the sample, since the illumination spots can only be translated by a single pixel, this corresponds to a step size of 120 nm between illumination locations. When imaging with the 1× tube lens, the CairnFocal's step size is 137 nm, whereas it is 91.3 nm when the 1.5× tube lens is used. In a CLSM system, the distance moved between scan positions determines the sampling frequency and must be less than that specified by the Nyquist criterion to avoid aliasing. In a widefield system, the pixel size determines the sampling rate that must meet the Nyquist criteria. In a SDCM system, the pinholes are swept continuously so once again, the camera pixel size determines the sampling frequency. In DMD confocal systems, the pinholes can only move discrete steps, the minimum size of which is determined by the DMD pixel size. It therefore stands to reason that both camera pixel size and the pinhole step size will have sampling criteria associated with them. Consider a hypothetical setup with a camera and magnification such that the effective pixel size exceeds the Nyquist limit for widefield. If the pinhole step size is set to several times the size of the pinhole then the final image would contain dark bands where the sample wasn't illuminated. Intuitively, it would make sense that both the CLSM constraint on illumination step size and the SDCM constraint on camera pixel size would need to be met to avoid aliasing on a DMD confocal system, though a more rigorous mathematical treatment is required to say for certain. If it's assumed that the DMD step size must meet the Nyquist sampling criterion associated with the illumination PSF ($\sim$ 107 nm) then the system would suffer some aliasing with the 1× tube lens but not with the 1.5×. Another consequence of the 1.5× tube lens is that the DMD patterns are projected onto a smaller region of the sample. Both the 200 nm and 100 nm bead data were taken using a pinhole size of 3×3 DMD pixels. With the 1× tube lens this corresponds to an illumination spot of 411 nm vs 274 nm with the 1.5× tube lens. The former is slightly larger than the widefield diffraction limit and the latter is slightly smaller. Since the illumination spots in the analysis of ISM are assumed to be diffraction limited, it's also possible that this is the cause of the sub-optimal performance of pixel reassignment on the 100 nm bead data. Unfortunately, since the 1.5× tube lens reduces both the illumination and detection efficiencies, it has so far not been possible to image the 100 nm beads with the 1.5× tube lens in order to determine if this would improve performance.

Figure 3.10 demonstrates the importance of accurately determining the location of all the pinholes within an image. When reconstruction is performed using pinhole locations determined by running ThunderSTORM on the unprocessed ISM data, the pinholes at positions in the field that don't contain any fluorophores won't be detected. Those regions will therefore contain no data after pixel reassignment. This in itself isn't a problem, in effect it increases the contrast of the final image as regions with no fluorophores will be completely black. It becomes a problem in regions of low intensity, however, where some pinholes in the region are detected but others are not. This manifests as increased noise in regions of low fluorescence and causes particular issues in imaging fine structures like beads or filaments. The solution to this problem is to ensure that all pinholes contribute to the final image even if they are not detectable in the ISM data. While this solves the problem, it introduces a variant of the background issue discussed in Section 3.4.2. Pinholes that would otherwise have been missed by ThunderSTORM and would therefore not contribute any background will now contribute background as they did when emulating SDCM, in effect lowering the contrast of the final images. Unlike the confocal case, however, a second effect degrades the resulting images further. The pixel reassignment process requires cutting out regions around each pinhole which get summed to produce the final image. The regions around neighbouring pinholes will overlap by an amount determined by the R parameter (see Section 3.3.3). Ideally, each pixel in the output image would receive contributions from the same number of regions, however, due to inevitable inaccuracies in the localisation of each pinholes, the regions will not perfectly overlap. Some pixels will therefore receive contributions from slightly too many regions and some from slightly too few, resulting in a distinct grid pattern being superimposed upon the final ISM image. As long as the localisation is sufficiently accurate and the regions are chosen to be sufficiently large that the pixel values at the edges decay to approximately the background value (though not so large that they encompass data from neighbouring pinholes) then this problem can be solved by subtracting the background value from each pixel before summing into the output image. If the imaging setup is such that the illumination region visible in confocal isn't present, then this can be achieved by subtracting a constant value equal to the camera offset. If the illumination region is visible, then the background value that needs to be subtracted will vary across the image. The easiest way to perform correction in this case is to turn the DMD off and take a number of images (around 100 will suffice) with the exposure time and laser power that was used during the ISM acquisition. The average of these images gives a good approximation to the background that will be encountered at each pixel in the ISM dataset. The drawback of this form of background correction is that the background data must be taken for every combination of exposure time, laser wavelength and laser power used in an acquisition. Figure 3.14 shows the result of pixel reassignment with the methods of background correction described.

Using techniques to infer the location of pinholes, rather than measuring their locations in the raw data, allows more accurate reconstruction in regions of low or no fluorescence. Running ThunderSTORM on the raw data detects pinhole locations to sub-pixel accuracy, which is required since the processing is typically performed on an image scaled up by 8×. Any inference techniques must be capable of locating pinholes to a similar degree of accuracy. Inaccuracies of one or two pixels will increase the prevalence of patterns in the reconstructions (though this may be fixed by using the background corrections described above) and reduce the final resolution. Larger inaccuracies additionally cause lower SNR and can even result in losing signal entirely. Two separate location inference techniques were used during the course of this work, taking a 'blank' dataset using the same confocal patterns used in acquisition but with the ring illuminator and calibrating the DMD to the camera using a calibration dataset. The former provides similar accuracy to simply running ThunderSTORM on the raw data but requires taking an extra dataset per pinhole size/spacing combination. Difficulties finding a calibration in the presence of the non-linear distortions introduced by both the correction optics and the 1.5× CMount magnifier result in lower accuracy pinhole locations. Additionally, with the calibration scheme described above, the inaccuracy tends to be location dependent. The model struggles to fit the regions around $r = 0$ and the line $\theta = \pi$, resulting in distortions in these regions even when accuracy elsewhere is high. The advantage of the calibration, however, is that only one calibration dataset needs to be taken instead of one per pinhole size/spacing combination. The calibration scheme above relies on an SVM using a polynomial kernel to learn to correct the non-linear distortions as it performed the best during optimisation using MATLAB's optimisation routines. An SVM using a Gaussian Radial Basis Function kernel [152] or a multi-layer neural network would likely have performed better given enough data, however, training these models on the 1000 images these tended to over-fit the training data and perform poorly. The polynomial kernel was chosen based on the dominant distortion being the pincushion introduced by the 1.5× CMount magnifier being roughly approximated by a low order polynomial [153]. Both techniques rely on knowing the image that was being displayed on the DMD for every image in the ISM dataset. Since the ISM patterns are generated in a deterministic order and the synchronisation ensures that one image of the DMD is taken per DMD image, knowledge of the DMD image corresponding to the first camera image and the pinhole



(a) (b) (c)

Figure 3.14.: ISM reconstructions of 100 nm beads, imaged with 3 pixel×3 pixel pinholes and 15 pixel spacing, reconstructed using different methods of background correction. (a) shows the result of pixel reassignment without any form of background correction. Since each DMD pixel spends more time in the *off* state than the *on* state, the background contributed by *off* pixels overwhelms the signal contributed by the *on* pixels and the beads are difficult to see in the resulting image. Since the background varies throughout the image, varying brightness and contrast settings fails to make the beads more visible across the whole field. Additionally, a distinct line pattern is visible, caused by slight variance in pinhole localisation. (b) Shows the effect of subtracting off a constant value from each pixel before summing. This improves contrast significantly and removes most of the pattern, however, cannot compensate for background variations caused by uneven illumination and so the illumination region is still visible. (c) shows the result of using an image of the DMD in the *all off* state to estimate the background at each pixel, in order to subtract off a local background. This technique shows the best contrast and removes the visibility of the illumination region.

size and spacing is sufficient to derive the images that were displayed on the DMD during every camera image. Since the DMD always starts displaying images at the beginning of a pattern, if the user starts an acquisition immediately after starting display of the ISM pattern on the DMD then it is usually safe to assume that the first image corresponds to the first DMD image. Care must be taken, however, when performing multiple acquisitions without restarting the DMD pattern in the CairnFocal Control program as the 95B will send a short output pulse on its expose out line at the end of acquisition. This pulse doesn't correspond to a camera image but is long enough to trigger the DMD to change to the next image. This situation occurs when using the CairnFocal µManager plugin to perform z-stacks or multi-colour imaging. A separate camera acquisition is started for each of the ISM images so the spurious output pulse will result in the start image advancing by one every ISM image. An offset option was introduced into the ImageJ reconstruction plugin to allow the user to account for this drift in start frame without needing to change the pinhole locations provided to the plugin.

ISM gives improved resolution when compared to widefield while maintaining the optical sectioning capabilities of confocal. It provides a similar resolution enhancement to SIM but is better suited to thick samples where the improved z resolution allows the resolution of structures deep within the sample that would otherwise be obscured by out of focus light. This is achieved with lower light intensities than are required for STED or many SMLM techniques. The main drawback, however, is that a large number of images is required per single ISM frame. While SIM achieves isotropic resolution enhancement with only 9 camera frames [49], implementation of ISM on the CairnFocal requires potentially hundreds of camera frames (the 100 nm and 200 nm beads data were taken with a pinhole size of 3 and spacing of 15 and therefore required 324 camera frames per ISM image). The large number of camera frames required has several drawbacks.

Firstly, since each camera frame is ~ 2.8 megabytes in size, multidimensional ISM imaging can easily produce large datasets, hundreds of gigabytes in size. For example, each beads dataset consisted of 41 z slices, resulting in a file size of just less than 38 gigabytes, if multiple colours had been used as well, this could easily exceed 100 gigabytes in a single dataset. Large datasets such as this put significant strain on data storage solutions, with many institutions poorly equipped to handle data of this size. This problem is only compounded by the requirement from some grant giving bodies for data collected to be stored for up to 10 years [154]. It should be noted, however, that while the raw ISM data is large, it is also relatively sparse, usually consisting of small regions of signal (the pinholes) separated by large regions of background. Lossless compression generally performs well on these sorts of datasets since background counts typically only vary by a few counts above and below the mean. Anecdotally, simply zipping the ISM datasets using the Windows zip utility reduced file sizes of the beads datasets to more manageable 17.15 gigabytes (down from 38 gigabytes). Some modern file formats, such as HDF5 provide, built-in support for lossless compression of data, allowing access to compressed datasets without the need to explicitly unzip files [155].

The large number of images required for each ISM frame also increases the time required to acquire each frame. This has two primary drawbacks, movement artifacts and drift. Since each area of the field is imaged at a slightly different time, any movement of the sample will result in subsequent camera frames detecting the sample at different positions (similarly to the artifacts introduces when imaging fast objects with a rolling shutter camera). Because of the particular patterns used, this introduces a 'wobble' to the reconstructed image, as in Figure 3.15. The second issue of sample drift caused particular issues when imaging the 100 nm beads. The system suffered from z drift that was detectable over the course of ~ 1 min. While this caused minimal issues for widefield and confocal images, where as long as the beads were refocused between z-stacks the drift was negligible, it prevented accurate measurement of the PSF in the z dimension, due to the resulting z steps being indeterminate. In principal, this could also have caused issues in defining the lateral PSF, however, in practice the beads were sufficiently small that their signal was only detected over a small number of pinholes which would be illuminated in a short period of time compared to the total ISM frame time. The issue of long frame times is theoretically entirely counteracted by the fact that the exposure of each frame can be made shorter than with other techniques. For example, if a confocal image, with a pattern consisting of 100 DMD images, required an exposure time of 100 ms, then the same total signal level should be achievable with a 1 ms camera exposure in ISM, due to needing to collect 100 images. In practice, however, the maximum frame rate of the camera usually provides a lower limit on the exposure time that prevents total exposure time parity with confocal acquisitions. An additional difficulty with this approach is that the camera's read noise in such an ISM acquisition would add in quadrature so even if the total signal level is the same, the noise level would be higher. In practice, pixel reassignment provides an increase in SNR and Richardson-Lucy deconvolution has the tendency to reduce noise in the final images, so this increase in read noise seldom has a noticeable effect.

3.5. Conclusion

This chapter has demonstrated the CairnFocal's ability to perform several different microscopy techniques primarily by varying the patterns displayed on the DMD and the synchronisation between the DMD and the sample. Analysis of the techniques have shown the system to have comparable performance to similar systems reported in the literature. The next chapter will build on this work by exploring the possibilities of adaptively switching modalities during acquisition in order to extend the lifespan of live samples during imaging.

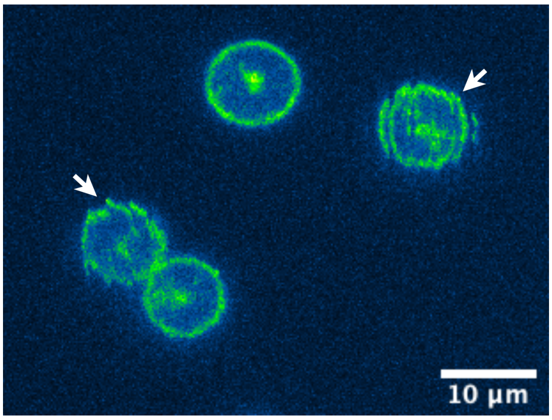


Figure 3.15.: ISM image of several human neutrophils. Due to the fact the image is built up from multiple individually illuminated regions, sample movement during the acquisition of a ISM frame has the tendency to introduce stripes into the reconstructed image. In this image, the two neutrophils labelled with arrows moved during the acquisition, whereas the other two remained relatively stationary and free of this particular artifact. The direction of the stripes indicates the direction the pinholes are initially moved. In this case, the pinholes have been moved vertically before they've been moved horizontally. Since the time between two consecutive camera frames is minimal, there is little movement between frames and the image is relatively along direction of the primary axis of the pinhole movement. In the perpendicular direction, the sample has more time to move, resulting is discontinuities along the secondary axis.

CHAPTER

FOUR

REACTIVE MODALITY SWITCHING

4.1. Introduction

The previous chapters outlined work undertaken to develop hardware and software to enable the CairnFocal to perform multiple imaging modalities. This chapter addresses the application of the resulting system to the imaging of live biological samples and further development to facilitate multi-modality imaging.

4.2. Materials and Methods

4.2.1. CairnFocal

The hardware setup was as described in the previous chapters. The multi-band pass filter set was used for all acquisitions to allow multicolour imaging. The FPGA trigger interface provided synchronisation between the various components of the system.

4.2.2. Acquisition Software

µManager was used for data acquisition and the CairnFocal control program used to upload patterns to the CairnFocal and configure the FPGA trigger interface. When one-off patterns not available in the CairnFocal control program were required, a short MATLAB script was used to upload the patterns in place of the CairnFocal control program (the FPGA trigger interface was not used in these acquisitions). The CairnFocal µManager plugin described in the previous chapter was extended to provide support for multi-modality acquisitions. Some of the data shown in this chapter were collected prior to the creation of the CairnFocal µManager plugin. In these cases, ISM was performed using custom µManager BeanShell scripts. Extensions were written in the Kotlin programming language, using the IntelliJ IDEA IDE (JetBrains, Prague, Czechia).

4.2.3. Samples

Live human neutrophil samples were provided by Dr. Lynne Prince and her PhD student Yin Xin Ho (Department of Infection, Immunity and Cardiovascular Disease, University of Sheffield). The samples consisted of live human neutrophils labelled with CellMask Deep Red membrane dye and infected with GFP expressing *Staphylococcus aureus*. During imaging, the samples were maintained at 37 °C using the UNO-T Stage-Top incubator (Okolab, Naples, Italy), mounted into the Nano-Z100 *z*-stage (Mad City Labs, Wisconsin, USA).

Live amoeba samples were provided by Dr. Jason King (School of Biosciences, University of Sheffield). Samples were intended to be dually labelled with GFP and Red Fluorescent Protein (RFP), however, the GFP labelling failed so epi-fluorescent imaging was performed using only the RFP channel.

4.3. Results

4.3.1. Neutrophils

Neutrophils were imaged as part of a separate PhD project investigating the mechanisms by which *S. aureus* is able to defeat the immune response provided by human neutrophils. The primary aim of imaging with the CairnFocal was to attempt to resolve any damage to the cell membrane caused by *S. aureus*. The secondary aim was to image for extended periods of time in order to detect long term changes in neutrophil behaviour and determine if *S. aureus* is capable of surviving inside the neutrophil. Super-resolution is required to resolve any potential damage to the cell membrane, however, imaging neutrophils in super-resolution presents a difficulty due to their light sensitivity. Super-resolution techniques typically require higher illumination intensities or longer total exposure times when compared to widefield, leading to increased phototoxic effects. Continual super-resolution imaging, as is often required on single-modality super-resolution microscopes, is likely to be ineffective on light sensitive samples such as human neutrophils. The CairnFocal's ability to switch between multiple modalities not only provides the ability to optimise the choice of imaging modalities to make appropriate trade-offs between illumination harshness, resolution and SNR, but also allows for these decisions to be remade

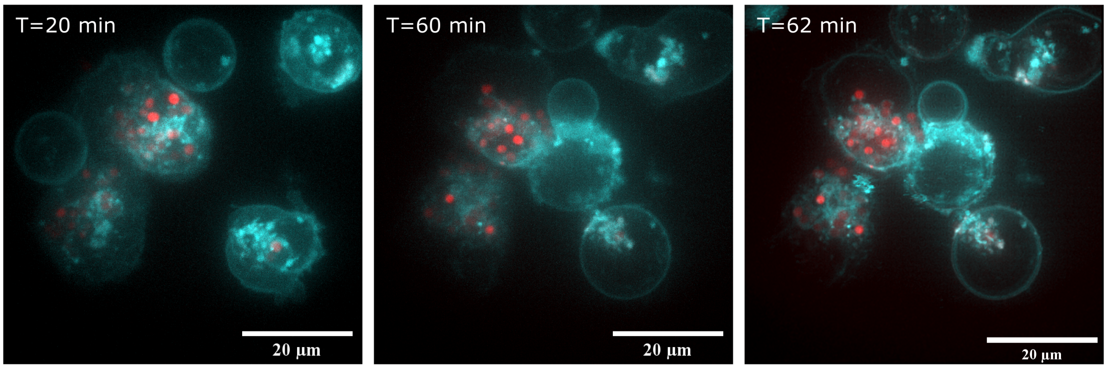


Figure 4.1.: Selection of images from a multi-modality time series acquisition of neutrophils. Initially, a 2-colour confocal z-stack was taken every 30 s for 1 h. After 60 min, the system was manually switched to ISM mode and 2-colour ISM z-stacks were acquired once a minute for 20 min. The neutrophils remained active throughout the confocal imaging period. Unfortunately, the data was acquired early on in the project and as a result the imaging, particularly the ISM was poorly optimised. The images shown are maximum value z-projections through the z-stacks acquired in each colour.

during acquisition as sample conditions change.

Figure 4.1 shows an application of this principal. Long term imaging was required in to track the behaviour of the neutrophils and in order to detect reproduction of the *S. aureus*, however, high-resolution images to detect any damage to the cell membrane by the end of the acquisition were also desirable. Continuous imaging with ISM would likely degrade the sample rapidly, preventing measurement of one or more of the features of interest. To address this issue, in the data shown, the neutrophils were imaged once every 30 s for 1 h in 2-colour confocal (3 × 3 pixel pinholes with 7 pixel spacing), using µManager's MDA. Once this time series acquisition was complete, the system was manually switched into ISM mode and a 2-colour ISM image was acquired every 1 min for 20 min using custom µManager BeanShell scripts. This allowed the sample to be imaged over an extended period, whilst still allowing super-resolution images to be collected when they would yield the desired information.

4.3.2. µManager Plugin for Modality Switching

While the preceding experiment demonstrated the basic premise behind multi-modality acquisitions, the execution required a large amount of user input. To explore the extent to which multi-modality acquisitions could be automated, the CairnFocal µManager plugin developed for performing ISM acquisitions was extended to perform modality switching.

The primary difficulty with the modality switching revealed by the neutrophil experiments, was that changing modality did not just require changing the DMD pattern and system synchronisation, but also other parameters such as illumination intensity and camera exposure time. While the DMD images and synchronisation (via the FPGA trigger interface) could be handled by the CairnFocal control program, the other system parameters could only be set from within µManager. To allow a single program to control both the DMD and the other system components, the CairnFocal plugin was extended to implement the CairnFocal Control client server protocol outlined in Chapter 2.

Figure 4.2a shows the GUI tab associated with the modality switching extension. Once the Modality Switcher connects to the CairnFocal Control program over TCP/IP, it automatically queries the available patterns. The user is then able to list the modalities that they wish to perform during the acquisition, along with the basic settings (e.g., exposure time, inter-frame interval, number of frames and z-stack options) provided by µManager's MDA mode. When the acquisition begins, the plugin performs each of the acquisitions requested by the user, automatically reconfiguring the µManager imaging parameters and sending requests to the CairnFocal control program to change the running pattern as appropriate.

Behind the scenes, the plugin uses the JeroMQ Java implementation of the ZeroMQ messaging protocols. Inter-thread communication and communication between the GUI and the acquisition code, is handled using Kotlin Coroutines. Unfortunately, the JeroMQ library, being written in Java, has no direct support for Kotlin Coroutines. Instead, it exposes a relatively low level API based on polling and blocking calls as opposed to the easier to use and more scalable asynchronous model employed by Kotlin Coroutines. Care must be taken when integrating JeroMQ code with Kotlin Coroutines code as JeroMQ has the restriction that sockets are not thread safe and cannot be moved between threads. In order to improve scalability without introducing code complexity, Kotlin Coroutines will automatically move work between threads based on the current work loads. To prevent this, a "`Client`" class encapsulates the JeroMQ sockets to provide coroutine-safe access. This class uses a single thread coroutine context on which a coroutine is launched that continuously polls both the JeroMQ updates socket for new messages and a Kotlin Coroutines `Channel` for requests from the modality switching plugin. When a message is received over the `Channel`, it is converted to a string matching the appropriate command in Table A.2 and sent over the Client's internal DEALER socket to the CairnFocal Control program. Parts of the plugin that wish to receive messages from the CairnFocal Control program must implement the "`ClientMessageListener`" interface, which provides a series of functions that will be called in response to the various messages that come from the CairnFocal Control program and also important events, such as when the connection to the CairnFocal Control program is established or lost. The `Client` class stores a reference to an implementer of the `ClientMessageListener` interface, when messages are received on the `Client's` internal SUB socket, any payload is parsed using the KotlinX Serialisation library and the corresponding function of the `ClientMessageListener` is called.

In order to allow code sharing between the Modality Switcher tab and a future variant of it, the `ClientMessageListener` interface is implemented by the "`DefaultAcquisitionController`" class, rather than by the Modality Switcher GUI class directly. On creation the `DefaultAcquisitionController` is passed two factory classes, one that can be used to create new objects representing specific acquisitions and one for creating messages that can be passed to the acquisition, as well as a GUI model object that can be used to update the state of the GUI. The `DefaultAcquisitionController` handles updating the GUI in response to all the common interactions with the CairnFocal Control program (e.g., connection, disconnection, heartbeats, etc.) as well as creation and cancelling of acquisitions in response to GUI events. Any messages from the CairnFocal control program that aren't handled by the `DefaultAcquisitionController` are converted to an acquisition specific message and passed onto the currently running acquisition if one exists.

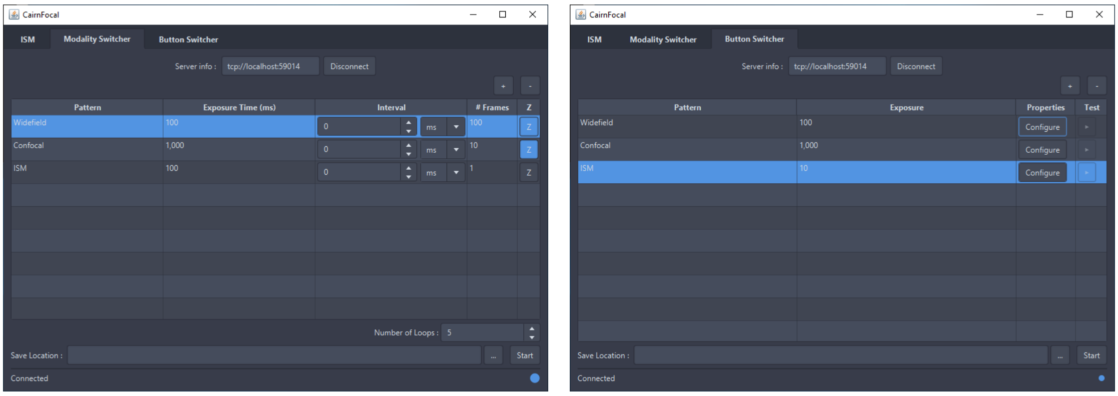
When created, the Modality Switcher acquisition object takes the list of modalities and constructs an object responsible for performing that modality's acquisition. In general, that consists of creating the datastore to use to store data as it arrives from the camera and requesting information about the synchronisation setup from the CairnFocal Control program (the most important information being the number of camera images required per image). The CairnFocal Control program is then instructed to run the required pattern, at the same time informing it of the current camera exposure. The the required number of camera frames are then captured before instructing the CairnFocal Control program to stop the pattern and, finally, saving the data to disk. This process is repeated in order to acquire the specified number of time points and z positions before moving onto the next modality. Multi-colour imaging is automatically handled via the FPGA trigger interface and information about which colour channels are to be imaged is part of the information requested about the pattern before it is run.

While the Modality Switcher section of the CairnFocal plugin provides a GUI for specifying acquisition protocols of the type used during the human neutrophil investigations, it also lays the groundwork for a more powerful variant of multi-modality imaging on the CairnFocal. The Modality Switcher requires the that the order and duration of each of the different modalities within the acquisition is known and specified *a priori*, however, often this isn't desirable. Since the Modality Switcher brings control of the entire system under one program, it is in principal possible to have the system switch modality in response to arbitrary events. As a proof of concept, the CairnFocal µManager plugin was once again extended, this time to support changing modality in response to GUI button presses. Similarly to the Modality Switcher, the Button Switcher allows the user to pre-specify a series of patterns, however, unlike the Modality Switcher, the patterns do not run immediately once acquisition starts or in order. Instead, no images are acquired until the user clicks the play button corresponding to a particular pattern. When the user clicks a play button, the plugin configures the µManager controlled devices and instructs the CairnFocal Control program to start the appropriate pattern. Data is collected continuously until either the user presses the stop button or the play button corresponding to another modality. The GUI associated with the Button Switcher is shown in Figure 4.2b.

4.3.3. Amoeba

Epi-fluorescent imaging presents similar challenges for investigation of amoeba as it does for human neutrophils, due to their similar sensitivity to light. Reliable, high resolution images of the amoeba's membrane as it undergoes phagocytosis were of particular interest to Dr. King, but since phagocytosis is an intermittent process, high illumination intensities associated with super-resolution imaging have the tendency to kill the amoeba before it occurs. Amoeba are an ideal sample with which to demonstrate the modality switching capabilities of the CairnFocal.

Figure 4.3 shows some preliminary results where multi-modality imaging was used to investigate phagocytosis in amoeba. To get a measure of the amoeba's sensitivity to the various imaging modalities, they were imaged continuously for several minutes using the three epi-fluorescent techniques described in Chapter 3. Even after 1 min of widefield imaging (1 frame every second, 50 ms exposure, and the lowest laser powers available from the LDI), the amoeba towards the centre of the illumination region displayed markedly reduced activity and changes in morphology, becoming more regular and circular in appearance. This was taken as evidence of significant damage to the amoeba. It was therefore determined that any epi-fluorescent images on the CairnFocal would be unsuitable for extended time-lapse imaging of amoeba. A Modality Switcher acquisition was set up to perform 1 min of brightfield imaging (using the system's ring illuminator) followed by a single ISM frame before switching back to brightfield, repeated for 10 min. No changes in activity or morphology were noticeable during the initial brightfield imag-



Server info : tcp://localhost:59014

Pattern	Exposure Time (ms)	Interval		# Frames	Z
Widefield	100	0	ms	100	Z
Confocal	1,000	0	ms	10	Z
ISM	100	0	ms	1	Z

Number of Loops : 5

Save Location :

Connected

(a)

Server info : tcp://localhost:59014

Pattern	Exposure	Properties	Test
Widefield	100	Configure	▸
Confocal	1,000	Configure	▸
ISM	10	Configure	▸

Save Location :

Connected

(b)

Figure 4.2.: (a) The Modality Switcher extension to the CairnFocal µManager plugin. The user connects to the CairnFocal Control C++ plugin using the TCP/IP address specified. Once connected the plugin retrieves the patterns currently available in the C++ program, allowing the user to specify the desired modalities by adding rows to the table using the '+' button in the top right. For each modality, the user can specify the exposure time, inter-image interval, number of frames to acquire and whether to enable z-stacks. Right-clicking on the 'enable z-stacks' button opens a dialogue box that allows the user to select the range of z positions to include in the z-stack. The number of loops control allows the user to instruct the plugin to repeat the modalities a set number of times. The save location field indicates where the plugin should stream data. Below that is the status bar, containing an area for status messages and a circular indicator that pulses whenever a heartbeat message is received from the C++ program. (b) is identical in most respects, though the table used to specify modality settings is slightly different. This table doesn't allow the user to specify inter-image interval, z-stacks or the number of frames to acquire. Instead, it provides a 'configuration' button which displays a dialogue allowing the user to specify µManager properties that should be set when the modality starts (e.g., laser powers) and a play/stop button. In this mode, when an acquisition is running, the user can press the play buttons associated with each modality to select which modality to image with.

ing, however, the amoeba reacted similarly to previous epi-fluorescent imaging tests in response to the collection of even a single ISM image.

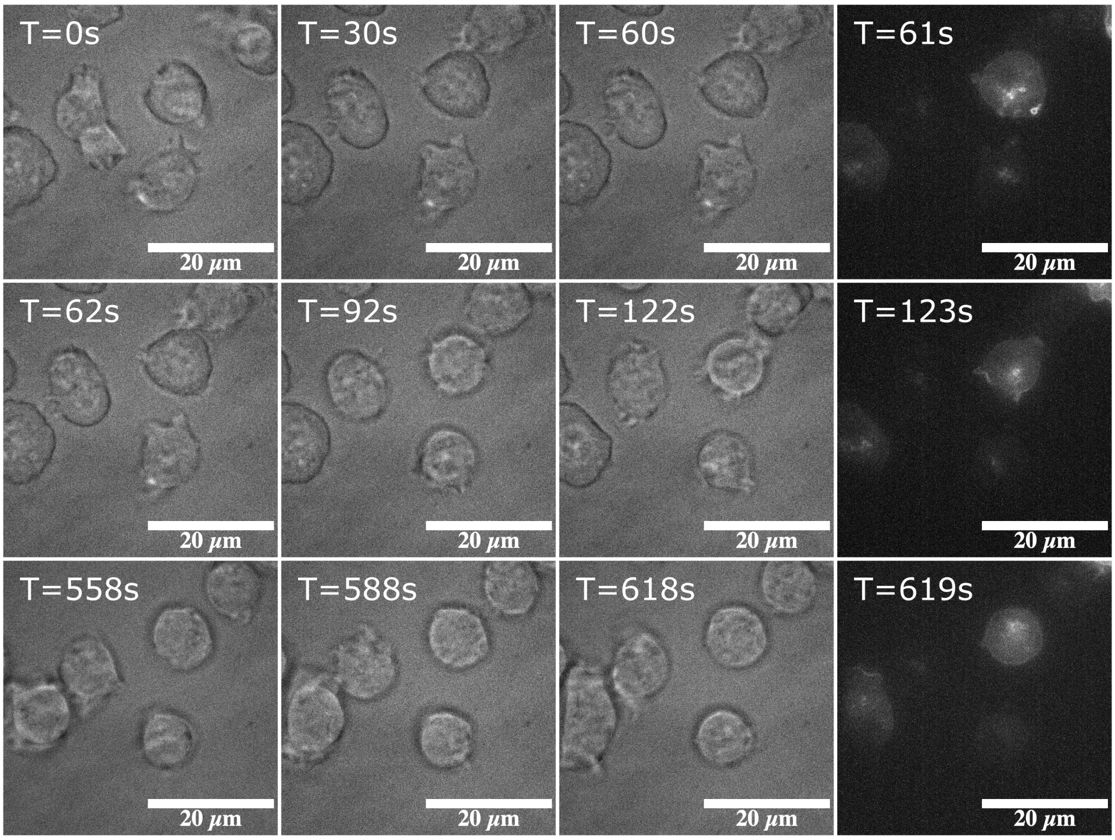


Figure 4.3.: A selection of images from a multi-modality time series acquisition of amoeba. The first 3 columns show images taken in the brightfield modality, the final column shows data acquired using ISM. The amoeba were imaged at a rate of 1 image a second in brightfield. After a minute of brightfield imaging, the system automatically switched into ISM mode to acquire a single image, before automatically switching back to brightfield mode for another minute. This process was repeated a total of 18 times. During the initial brightfield imaging (T = 0 s until T = 60 s), the amoeba remained active and assumed various morphologies, indicating little to no damage from the imaging. After the first ISM image, the amoeba became noticeably less active, with the amoeba at the centre of the frame (which received the highest intensity laser illumination during ISM imaging) tending to assume a constant circular morphology, which remained relatively unchanged throughout the entire acquisition, indicating significant damage to the sample from ISM imaging. The RFP labelling was such that no high resolution structures were present in the sample and as such, the ISM images were not reconstructed, with the images shown being maximum value projections through the raw ISM dataset. The brightfield images were taken with a 100 ms exposure time. The ISM data consisted of 64 camera frames taken with 5 ms exposures (320 ms total exposure).

4.4. Discussion

Multi-modality imaging of human neutrophils allowed long term imaging and resolution of fine structures on the same sample. Neutrophils were successfully imaged for several hours at a time in confocal before being imaged in super-resolution. Over the course of several experiments the long term confocal imaging allowed the observation of processes such as phagocytosis. One experiment even detected what appeared to be a neutrophil undergoing vomocytosis, whereby the neutrophil expels previously engulfed bacteria, a process as yet unreported in human neutrophils to the best of the author's knowledge. The ISM failed to detect any damage to the cell membrane and it's unclear if this is due to too low a resolution enhancement or because there was no detectable damage. It was, however, able to resolve septum in bacteria contained within a neutrophil's phagosome. The presence of a septum implies that the *S. aureus* was splitting, however, further data is required to determine if the bacteria were undergoing this process when engulfed by the neutrophil or if they were able to reproduce from within the phagosome. Additionally, a one off experiment was done as a proof of concept of an alternative multi-modality technique. Figure 4.4 shows the result of imaging using a pattern where $\sim 50\%$ of the DMD was used to perform widefield and the other half used to perform confocal. When combined with the ability to select regions of the DMD by drawing ROIs on images of the field, as described in Chapter 2, this could be used to perform more highly damaging imaging techniques on parts of the field, while imaging the remaining parts with a less damaging modality.

4.4.1. Limitations of the Study

As the neutrophil imaging took place towards the beginning of the project, it was unable to benefit from later developments such as the ISM calibration, the ISM deconvolution and the CairnFocal plugin. This presented difficulties in data acquisition and also resulted in sub-optimal ISM reconstruction. Disruption due to the COVID19 pandemic and the subsequent graduation of the student who performed the sample preparation meant that the full capabilities of the system couldn't be applied to the investigation of the interaction between human neutrophils and *S. aureus* during the course of this project. The subsequent development of the CairnFocal µManager plugin greatly simplifies the use of the system and reduces the barrier to entry for any future students intending to continue this work.

The amoeba imaging similarly demonstrated the utility of modality switching. By imaging with the LED ring illuminator rather than using epi-fluorescence, the amoeba

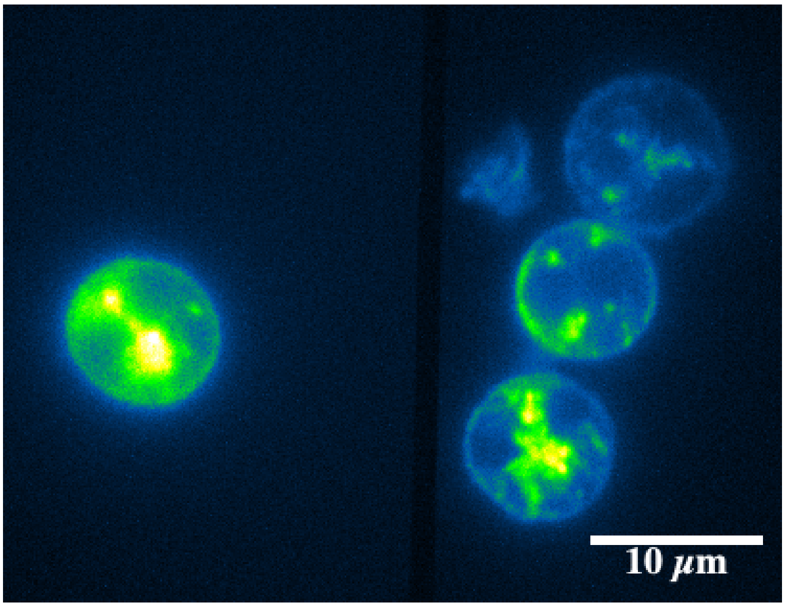


Figure 4.4.: Human neutrophils imaged under widefield and confocal simultaneously. The left hand side of the image was imaged using a widefield pattern, whereas the right was imaged using a confocal pattern. The DMD images consisted of a regular confocal pattern on the right hand side with the left hand side off, followed by a single image of a widefield pattern on the left hand side with the right hand side off. The whole pattern was displayed within a single camera exposure, resulting a simultaneous widefield and confocal image. The widefield pattern is displayed for a single DMD frame after the confocal pattern to prevent scattered widefield light passing through the pinholes in the other part of the field and to reduce the effective exposure time on the left hand side of the field such that the longer exposure time required for confocal would not saturate the image. The black bar down the centre of the image was introduced to make it easier to distinguish the two regions.

could be imaged for arbitrarily long times without them being detectably damaged. By switching to ISM, super-resolution images could be acquired at the expense of damage to the sample. In principal, an experiment to investigate phagocytosis in amoeba could currently be performed, whereby images are taken using the ring illuminator until the amoeba begins to engulf an object within the sample (e.g., a bead or bacteria) at which point the operator could instruct the microscope to switch to ISM, capturing the event in super-resolution. Unfortunately, time constraints prevented this experiment from being performed during this project.

One thing that was clear from the preliminary amoeba data collected using the Modality Switcher mode of the plugin, was that epi-fluorescent imaging of any kind, even widefield with the lowest available laser powers, damaged the samples quickly. Ideally, this experiment would be performed using epi-fluorescent widefield rather than the ring illuminator since this provides higher contrast images and would allow for specific labelling of proteins involved in phagocytosis. It's possible that non-damaging epi-fluorescence imaging could have been achieved by reducing incident laser powers using a neutral density filter in the excitation path. A higher contrast, label-free, imaging technique may be preferable to ensure minimal damage to the amoeba. As described in Chapter 3, the system is currently capable of performing phase contrast and darkfield. Out of these two label free techniques, phase contrast is preferable as darkfield requires low NA objectives which would reduce the resolution achievable with ISM.

4.4.2. Future Work

The extensions to the CairnFocal µManager plugin enable the system to automatically switch between modalities in response to certain events, a pre-specified number of images being acquired in the case of the Modality Switcher and an operator pressing a button in the case of the Button Switcher. Theoretically, since no mechanical components need to be moved in order to change modality, the system should be able to switch modalities quickly, requiring only the time it takes to change the orientation of the mirrors on the DMD (44 µs). In reality, the switching takes significantly longer than that theoretical minimum due to several factors. Firstly, the quoted 44 µs is the time it takes to switch to a new frame within a single pattern currently uploaded to the DMD and it's likely that switching between DMD patterns will require a longer time due to USB communication overheads. Updating the settings on the FPGA trigger interface can also take a long time (of the order of seconds) if several settings need updating to switch to the new modality. This is primarily due to the low baud rate used, 9800 baud, which was initially chosen to improve stability during development. It's likely possible to up this rate to a higher value in order to speed up reconfiguration time. Currently, each setting is updated using an individual serial command, introducing support into the SDK for batched updates, where several settings are updated with a single serial command, would reduce the overheads associated with initiating serial transfers, again speeding up reconfiguration. At present, the CairnFocal Control program caches the current settings and only issues commands to update a setting if it needs changing. This reduces the time it takes to reconfigure the trigger interface to perform similar trigger patterns, however, switching between two dissimilar patterns will still incur a large reconfiguration time. To address this, the FPGA trigger interface could be extended to provide support for multiple configurable presets that could be switched between by issuing a single serial command. Updating the settings of the devices controlled by µManager also takes some time. The LDI also uses a low baud rate serial connection for configuration, but typically it's only necessary to update the laser power settings so this typically doesn't take long. Finally, the data must be saved to disk and the camera's acquisition restarted.

As a measurement of typical switching time, the 95B was removed from the microscope and a lens used to form an image of a phone's stopwatch app on the sensor. The Modality Switcher was set up to repeatedly swap between widefield and ISM modalities. The difference in the time shown on the stopwatch in the last image before a switch and the first image after a switch was recorded for both a single colour and 4 colour setup. The switch time for the single colour experiment was 1.236 s ± 85 ms and 1.503 s ± 67 ms for the 4 colour setup.

This work has demonstrated some of the benefits of multi-modality imaging for reducing phototoxicity in live cell imaging. The Button Switcher allows a trained operator to decide when to switch between a set of pre-defined modalities. This concept could be further extended to perform a form of 'smart microscopy', whereby the plugin analyses data on the fly in order to determine when to switch in a fully autonomous manner. One can imagine an experiment where a machine learning model is trained to predict the start of a process of interest from, for example, brightfield data, the output of the model being used as the trigger to switch to super-resolution modalities. The drawback of this technique would be that each new biological system or process would require training a new model, however, this could still be useful in situations where large amounts of data are required on the same process under slightly different conditions, as is common in drug discovery.

4.5. Conclusion

This chapter has described the application of the modalities developed in the previous two chapters to two light sensitive biological systems, that are typically too sensitive to image for long periods with super-resolution modalities. By dynamically switching modalities during acquisition, samples could be imaged over long periods of time before switching to ISM in order to get high-resolution information. Development of software to allow users to quickly switch modalities at the click of a button was described. This software also provides the platform on which a fully autonomous, modality switching, microscope could be built.

CHAPTER

FIVE

DEVELOPMENT OF SMFBOX AND EI-FLEX

5.1. Introduction

While the primary focus of this thesis is the development of the CairnFocal to allow it to perform super-resolution and multi-modality imaging of biological samples, some time during the PhD was also spent collaborating with Dr. Tim Craggs' group (Department of Chemistry, University of Sheffield) to further develop their confocal smFRET microscope, the smfBox. The work formed part of a 2020 publication [84] where both the software and the microscope designs were made open source and freely available to the academic community.

In late 2021, Dr. Craggs span out a company, Exciting Instruments, with the aim of commercialising the smfBox. The software developed during this project constituted a significant portion of the intellectual property that was licensed by the University of Sheffield to Exciting Instruments in order to facilitate the commercialisation process. A 3 month placement was undertaken to further develop the software.

5.1.1. Hardware Overview

The smfBox is a fixed-spot confocal microscope, designed to perform solution based smFRET experiments, which can be constructed with relatively little specialist knowledge for approximately £40,000. 515 nm and 638 nm laser illumination from a LightHUB-2 (Omicron-Laserage Laserprodukte GmbH, Dudenhofen, Germany) is collimated via the collimator provided with the LightHUB, and coupled into the back aperture of a 60x, 1.35 NA, Super Apochromat, Oil immersion objective lens (Olympus Life Sciences, Essex, UK) via a ZT532/640rpc dichroic filter (Chroma Technology Corporation, Vermont, USA) in order to form a diffraction limited illumination volume. The width of the beam can be controlled using an adjustable iris in order to under fill the back aperture of the objective and form a larger illumination volume. Back reflection from the cover slip-sample interface is imaged onto a DCC1545M camera (Thorlabs Inc., New Jersey, USA) and used for determining focus. 10% of the laser light is directed via a 90/10 beam-splitter (Chroma Technology Corporation, Vermont, USA) and a 63.5 mm focal length achromatic lens (Edmund Optics Ltd., York, UK) onto a DET10A2 1 ns rise time photodetector (Thorlabs Inc., New Jersey, USA) in order to measure the alternation and rise/fall times of the lasers. Focus is controlled using a FOC500 piezo z-stage (Piezoconcept, Lyon, France) to adjust the position of the objective.

Emitted light collected by the objective is focused onto a 50 µm pinhole (Newport Corporation, Irvine, California), using a 50.8 mm focal length achromatic lens (Edmund Optics Ltd., York, UK). A 63.5 mm focal length achromatic lens (Edmund Optics Ltd., York, UK) is used to form an image of the pinhole at infinity. An NC395323 - T640lpxr dichroic mirror (Chroma Technology Corporation, Vermont, USA) is used to spectrally separate the emission light into ‘red’ and ‘green’ emission channels. An image of the pinhole is then focused onto the detector using a 75 mm focal length achromatic lens in the red channel and a 100 mm achromatic lens (Thorlabs Inc., New Jersey, USA) in the green. In both the red and green channels, an SPCM-AQRH-14 APD (Excelitas Technologies, Denbighshire, UK) is used for single-photon detection. The red channel uses a FF01-679/41-25 emission filter (Semrock, New York, USA) and a FF01-571/72-25 emission filter (Semrock, New York, USA) is used in the green channel.

A PCIe-6353 NIDAQ board (National Instruments, Texas, USA) is used for timestamping of output pulses from the APDs, measuring the output voltage of the DET10A2 and supplying the TTL signals used to alternate the lasers.

5.1.2. LabVIEW Software

Prior to the start of this project, control of the system was performed through a collection of 3 LabVIEW programs, one for data acquisition, one for alignment and one for focus control. While the programs proved sufficient for the majority of data acquisition tasks, they fell short on usability and presented issues for open sourcing and extensibility. As discussed in Chapter 2, being written in LabVIEW meant the programs couldn’t be directly hosted on GitHub in a form that could be reviewed or extended by end users. Additionally, while it is possible to package LabVIEW programs into executables that can be run without requiring a LabVIEW license, a lab replicating the smfBox wanting to customise the behaviour of the programs in any way would require a full LabVIEW license. For these reasons, a rewrite of the operational software that didn't rely on LabVIEW was required. This also presented an opportunity to address some usability issues by combining all three pieces of software into a single program, changing the output format and streamlining the user experience.

5.2. Materials and Methods

The software was written in C++ using the Microsoft Visual C++ 2017 compilers. The GUI was written using Qt 5.12 (Qt Company, Espoo, Finland). Development used the Qt Creator IDE (Qt Company, Espoo, Finland). Communication with the NIDAQ board was via the DAQmx C SDK (National Instruments, Texas, USA). The C SDK shipped with installations of the ThorCam software was used to interface with the focusing camera. Data were saved to the HDF5 file format using version 1.10.4 of the C++ SDK. During development, builds were regularly made available to members of the Craggs lab, who would compare data generated with the new software to that produced with the LabVIEW software to ensure correct operation.

5.3. Results

5.3.1. Software Structure

Figure 5.1 shows an overview of the structure of the smOtter software. The software is written as a Qt Modelling Language (QML) application. In QML applications, the GUI is defined in a variant of JavaScript, allowing both layout specification and implementation of GUI logic. Qt objects, written in C++, can be exposed to the QML code at runtime by registering the types with the QML engine and providing pointers to instances of those classes at program startup. The signals, slots and properties of the Qt object are then accessible as members of the corresponding JavaScript objects in QML. In order to separate the GUI code from the application code as much as possible, three classes are used to mediate between the GUI and application code, `NIDataSource`, `ThorCamSource` and `ZStage`. `ThorCamSource` and `ZStage` allow the GUI to communicate with the focus camera and *z*-stage respectively, whereas the `NIDataSource` is used to initiate and interact with acquisitions.

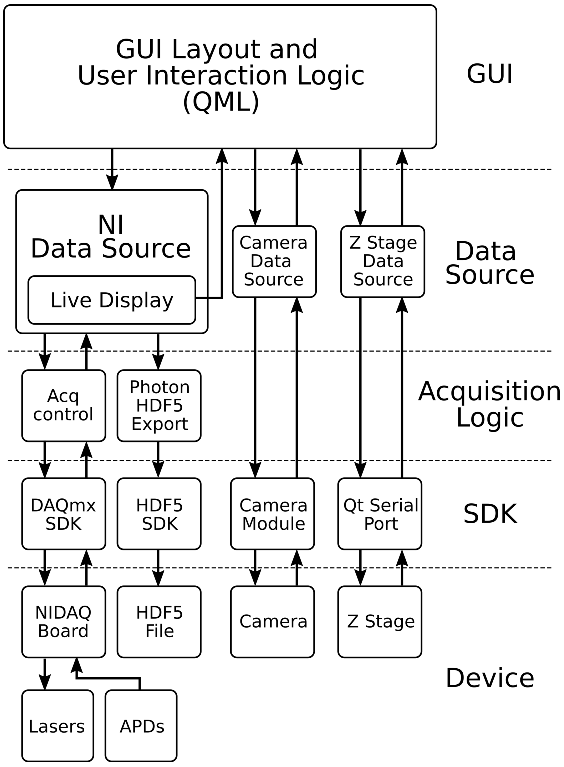


Figure 5.1.: The structure of the smOtter program, with data and control flow between sections of the program indicated by arrows. The program is arranged hierarchically, with each component only interacting with its parent and child components. At the same time, the structure is roughly split into several layers. The GUI layer is written in QML and is responsible for specifying the layout of the GUI and implementing the logic associated with user interaction, e.g., button presses. The next layer down is the Data Source layer. This layer is responsible for mediating between the GUI layer and the main program logic, as well as handling complex GUI updates, e.g., plotting graphs. The Acquisition Logic layer is responsible for controlling acquisition and ensuring that data is written to disk as the experiment progresses. The SDK layer represents the the interfaces between the smOtter program and the various devices comprising the microscope. While most of these are libraries external to the smOtter software, the Camera Module SDK was developed during the course of this project and will be discussed later in the chapter. The final layer represents the physical devices present in the system.

5.3.2. Data Acquisition

In smFRET with ALEX, the donor and acceptor lasers are enabled alternately [76]. When a photon is detected, its arrival time can be used to determine which laser was on when the photon was emitted. This information combined with the detector that detected the photon can be used to determine if the photon was emitted as a result of regular fluorescence (either donor fluorophore emission under donor excitation or acceptor fluorophore under acceptor excitation) or as a result of FRET (acceptor emission under donor excitation); donor emission under acceptor excitation is assumed to be noise and ignored. In order to be able to collect this information, the laser alternation and timestamp clock must be synchronised with respect to one another.

The PCIe-6353 board used by the smfBox has 4 configurable counters that can be used in signal generation or data acquisition as well as a multitude of digital outputs that can be used for generating signals that can be used to alternate the lasers [156]. The Omicron lasers used on the smfBox directly support alternating the lasers at frequencies suitable for smFRET experiments (typically 10 kHz is used on the smfBox) by supplying them with appropriate 5 V TTL signals [157]. The software creates a two-channel digital output task on the NIDAQ board, setting the update frequency to 1 MHz and supplying 100 samples per channel, where a value of 1 indicates the corresponding laser should be on and a 0 indicates it should be off. This ensures that laser alternation can be controlled with a resolution of 1 µs. Two extra channels are added to the digital output in order to control the gate inputs on the APDs. The APD gates are pulled high during acquisition and low after acquisition ends in order to disable the APD and reduce the probability of damaging it with high light intensities when changing sample.

The outputs of the APDs are each connected to one of the programmable function input pins on the PCIe-6353, allowing it to be connected to the inputs of one of the general purpose counters. In order to generate a timestamp for each photon, each timer is configured to count the internal 100 MHz timebase of the NIDAQ board, resulting in its value incrementing once every 10 ns. The APD output is routed to the 'readout' input on each of the counters. The current value of the counter is read into a buffer on the PCIe-6353 on the rising edge of the readout signal. Since the value in the counter increments every 10 ns, the buffer will be filled with the arrival times of photons in units of 10 ns. The data in this buffer is automatically transferred to the host computer where it is further processed.

By default, the DAQmx SDK applies no synchronisation between the counters and the digital output task used to control the laser alternation. As a result, the timestamps cannot be used to accurately determine which laser was enabled when the photon was detected. To enforce synchronisation between the various tasks in the system, it is necessary to define a common start signal for all the tasks. In the smOtter program, the counter tasks are instructed to use the board's digital output start trigger pin (`/do/StartTrigger` in the SDK) to trigger the start of their acquisition. The photon counting tasks are started before the digital output task so they are awaiting the digital output start trigger signal when it fires. Since all the tasks are running on the same board, using the same physical clock signal, synchronisation of the start of the tasks is sufficient to ensure synchronisation throughout the entire acquisition.

The setup as described thus far suffers from a severe limitation that must be addressed, namely that the counters on the PCIe-6353 are only 32-bits wide. When counting the 100 MHz timebase, this causes the counters to rollover once every ∼43 s. This is insufficient for smFRET experiments, which typically need to run for over an hour to collect sufficient data for reliable analysis [84]. In principal, its possible to cascade two counters on the board, such that when one counter rolls over, the other increments its value, effectively combining them into a single 64-bit counter. The number of counters on the PCIe-6353 is limited however, and it is intended that the smfBox could be extended to support up to 4 APDs, allowing measurement of other quantities, for example, polarisation. The 64-bit extension was therefore performed in software. Since timestamps for each channel arrive in order, a timestamp with a lower value than the one proceeding it indicates that at least one rollover has occurred. As long as at least one photon is detected within any 43 s window, it is guaranteed that no rollover has occurred unless a decrease in timestamp value is detected and that a decrease in timestamp value implies only a single rollover has occurred. The SPCM-AQRH-14 APD units used on the smfBox have a rated dark current of between 50 and 100 counts per second [158] and so this condition is guaranteed to be met even with no signal present.

While photon count rates during an smFRET acquisition are low, they are typically much higher when performing alignment or during FCS acquisitions, with the upper limit provided by the SPCM-AQRH-14 APDs, which are rated to produce up to 35 million counts per second. The DAQmx SDK automatically handles copying the timestamps to a buffer on the host computer, however, if this buffer isn't emptied sufficiently quickly, it can overflow, causing the SDK to stop recording new timestamps. Care must be taken to ensure that other program operations (e.g., updating the GUI, saving data to disk, etc.) don't impede the emptying of this buffer. To this end, smOtter continually reads photon timestamps from the buffer in a dedicated thread. This thread also performs the 64-bit extension and categorisation of the photons. While this helps prevent buffer overflows, it introduces questions about how best to share data between the acquisition thread and the GUI thread to allow users to view incoming data in real time. Reading and writing to state from multiple threads simultaneously can result in data races. Locks and mutexes are often used to serialize access and prevent data races, however, this can have the unintended side effect of blocking threads while they wait to acquire a lock on the shared state, reducing the amount of work being performed concurrently and the overall performance of the program. In order to prevent buffer overflows, the time that the GUI thread holds a lock on the mutex used to protect access to the analysed photons must be kept short. To achieve this, data access is limited to movement of analysed photons into the shared state by the photon counting thread and movement out of the shared state by the GUI thread. Conceptually, the shared state consists of a doubly linked list of photon timestamps (`std::list` from the C++ standard template library). The photon counting thread preforms all its analysis on local variables, resulting a local doubly linked list of new photon timestamps. It then acquires the lock on the shared state and uses the `splice` method available on instances of `std::list` to move the photons to the shared state. The `splice` method efficiently moves the photon data from the local list to the shared list without copying any data values; it requires copying only 4 pointers, as shown in Figure 5.2. The GUI thread performs the same operation in reverse, `splice`-ing photon timestamps from the shared state into its own local copy, which it can then process without interfering with the execution of the photon counting thread. This ensures that the critical sections associated with photon timestamp transfer between threads only consist of 4 pointer reassignments each.

In addition to the above, the software is also capable of recording from an analogue input pin on the NIDAQ board in order to measure the laser power recorded by the DET10A2 photodetector. In principal, this should allow the continuous monitoring of laser powers during an experiment, either to indicate laser power fluctuations or potentially to correct out effects due to laser power fluctuations over the course of an experiment. In practice, the DET10A2 produces an unamplified current signal which is low intensity and sensitive to impedance mismatch at the NIDAQ, which prevents its use for this purpose. A transimpedance amplifier is required to convert the current signal to a voltage that can be read by the NIDAQ board. A simple resistor can be used for this purpose, however, using a $50\,\Omega$ resistor to correctly match the impedance at the NIDAQ results in such low signal levels that the board barely registers a change between the laser's on and off states. Using a larger resistor value results in higher voltages, however, reflections due to the impedance mismatch quickly reduce the bandwidth of the circuit to the point that the signal is too heavily distorted to be useful. An intermediate value of $\sim 10\,\mathrm{k}\Omega$ provided the largest signal levels possible without so much distortion that laser

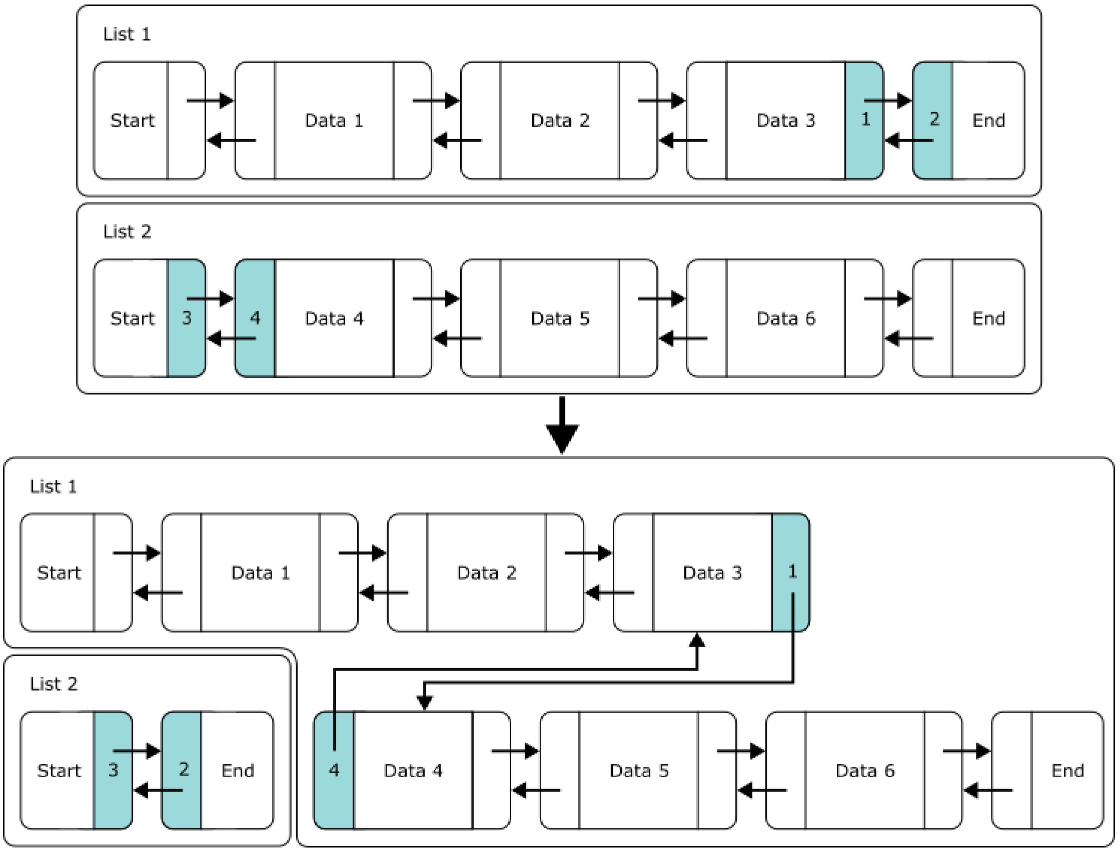


Figure 5.2.: Illustration of the splice operation on lists. Two doubly linked lists are shown, initially containing 3 data elements each. Each data block has 3 sections, from left to right: the backwards pointer, the data and the forwards pointer. The forwards pointer points to the next element in the list and the backwards pointer points to the previous element, allowing for bidirectional iteration through the list. The start and end blocks contain no data, only pointers to the first and last elements of the list respectively. By changing the values of the four pointers highlighted (P1 → D4, P2 → S2, P3 → E1 and P4 → D3) the data from List 2 may be transferred into List 1. The nodes of the list remain in the same locations in memory so the transfer will always take the same amount of time, irrespective of the number of data elements to be transferred.

alternation couldn't be detected, however, even these levels were too low to make any any measurement of laser power fluctuations.

5.3.3. Data Output

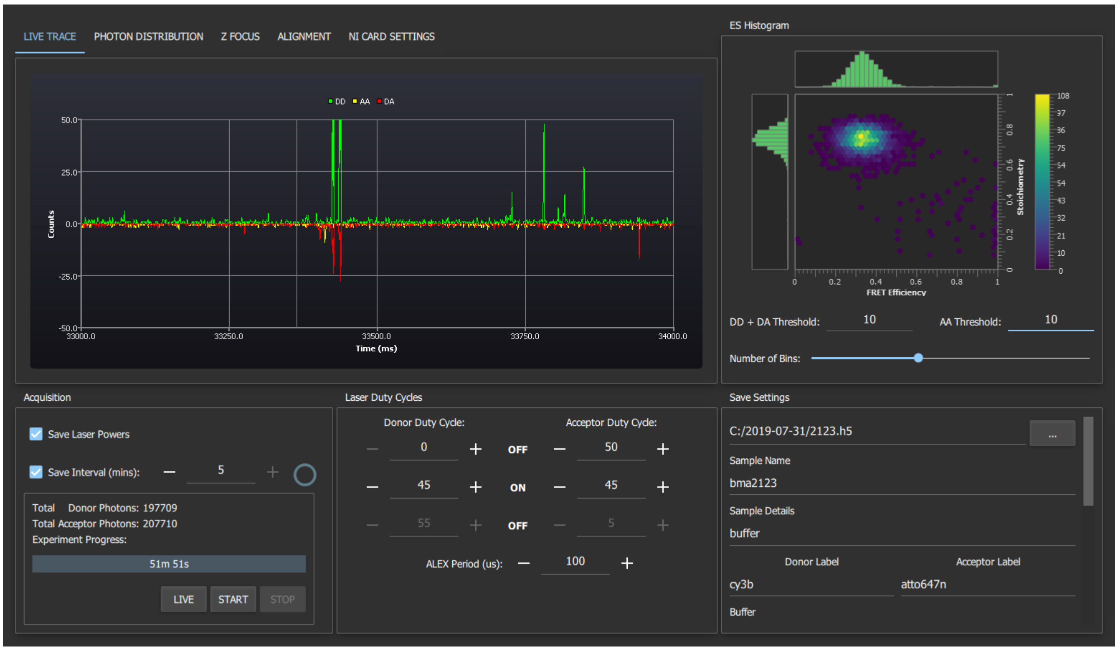
The original LabVIEW software provided the output of the system as a Comma Separated Value (CSV) file containing the timestamp and detector information for each detected photon. The smFRET analysis was performed using the open source, python based analysis package FRETBursts, which expects input to be in the open source file format PhotonHDF5 [75], [159]. As its name suggests, PhotonHDF5 is a file structure imposed on the HDF5 file format to allow it to both data and common metadata required in the analysis of single photon counting experiments, such as smFRET. In order to load the data into FRETBursts, it first needed to be converted from CSV into PhotonHDF5.

To improve the system's workflow, smOtter outputs directly into a PhotonHDF5 file. The user is required to enter all the PhotonHDF5 metadata relevant to smFRET experiments, which is written to the file upon creation. Writing directly to PhotonHDF5 has some other benefits, namely writing speed and compression. Since CSV is a human readable file format, all the numeric data must be converted to a string representation before it can be saved. Numeric data in an HDF5 file is stored in its binary representation so no serialization is required. While metadata is written to the file at creation time, the photon data is streamed to disk during acquisition. This requires the HDF5 dataset objects representing the timestamps and the detectors to be dynamically expanded as new data is saved. In HDF5, expandable datasets require the data to be "chunked" (i.e., the HDF5 library is not required to store it contiguously within the file). This is also the requirement for HDF5 datasets to be losslessly compressed, therefore, smOtter is able to automatically apply compression to reduce file sizes. Since the HDF5 libraries provide uniform access to both compressed and uncompressed data, the end user is able to enjoy the benefits of smaller file sizes without the need to manually compress and decompress the final files (e.g., using the operating system's zip utility). Due to the relatively low photon data rates during smFRET experiments, smOtter is able to apply the maximum compression level without degrading performance noticeably. On a selection of real smFRET data taken by members of the Craggs Lab, compression resulted in a compression ratio of ∼2.2:1 on the timestamps dataset and ∼4.9:1 on the detectors dataset. The compression typically performs better on the larger FCS datasets, generally achieving a compression ratio of ∼3.2:1 on the timestamps dataset and ∼52.6:1 for the detectors. The largest compression ratios seen were 3.67:1 for the timestamps and 52.629:1 for the detectors, which had the effect of reducing the file size from just over 2 gigabytes to just over 500 megabytes.

5.3.4. Graphical User Interface

Figure 5.3 shows the main view of the smOtter software, which is loosely based on the layout of the original LabVIEW. All interaction with the system can be performed from this screen, which is divided roughly into 5 regions. The user can configure the length of an experiment, set how regularly the software should save data to disk as well as start/stop live mode and acquisition modes from the acquisition panel in the bottom left. In acquisition mode, the software initiates an acquisition as described in the previous section, updating the various graphs in real-time. The acquisition panel also shows an indication of the total number of photons detected and the time remaining until the end of the experiment. The circular widget next to the save interval setting fills up to indicate the time until the program next saves to disk. Live mode does the same but continuously and without saving to disk. The user must explicitly stop a live acquisition by pressing the stop button. The bottom centre panel allows the user to select the duty cycle and alternation period of the laser. The bottom right panel allows the user to select the save location and enter all the metadata required for the PhotonHDF5 that can't be derived from the other system settings. While a lot of this metadata is optional in PhotonHDF5, smOtter treats it at mandatory and will not allow the user to start an acquisition without filling it in. The top right panel shows a 2D histogram of the FRET efficiency and stoichiometry values calculated from bursts of photons. As photon arrival times are streamed from the NIDAQ board, they are binned into 1 ms intervals. In a particular millisecond interval, if the number of photons under donor excitation passes a user defined threshold and the number of photons under acceptor excitation passes a second user defined threshold, it is treated as a burst. The FRET efficiency and stoichiometry of the photons within the burst are calculated using equations 1.31 and 1.32 respectively, and the value at the corresponding histogram bin is incremented by 1. The histogram widget was designed to mimic the hexagonal 2D histogram plots available in the matplotlib Python package [160]. The slider below the graph allows the user to control the size of the bins.

The top left panel shows a different set of controls depending on which of the tab buttons at the top of the screen is selected. Figures 5.4-5.7 show 4 of the options. The Live Trace view, Figure 5.4, shows a real time plot of the number of photons arriving, binned into 1 ms intervals. Donor photons are displayed as a green trace, increasing towards the top of the graph. Acceptor photons are displayed as two traces increasing towards the bottom of the graph, acceptor photons under acceptor excitation being



DD + DA Threshold: 10

AA Threshold: 10

☑ Save Laser Powers

☑ Save Interval (mins): 5

Total Donor Photons: 197709
Total Acceptor Photons: 207710
Experiment Progress: 51m 51s

Donor Duty Cycle:		Acceptor Duty Cycle:
0	OFF	50
45	ON	45
55	OFF	5

ALEX Period (us): 100

C:/2019-07-31/2123.h5

Sample Name: bma2123

Sample Details: buffer

Donor Label: cy3b

Acceptor Label: atto647n

Buffer

Figure 5.3.: The main view of the smOtter GUI. This screen was inspired by the original LabVIEW acquisition software. The redesign of the LabVIEW software's GUI attempted to modernise the design and reduce clutter in order to simplify the user experience. The view is split into 5 sections. The top right section (labelled ES Histogram) is a custom component replicating the matplotlib 2D hexagonal histograms produced as an output of analysis by the FRETBursts analysis package. The horizontal axis of the 2D histogram is FRET efficiency and the vertical axis is Stoichiometry. Bursts surpassing the user defined thresholds are added to the histogram in real time. The bottom left section (Acquisition) allows the user to specify the experiment parameters, start/stop acquisitions and displays the time remaining in the current acquisition. The bottom centre section (Laser Duty Cycles) allows the user to configure the laser alternation settings. The bottom right section (Save Settings) allows the user to select the data output location and enter extra metadata required for the output PhotonHDF5 files. The top right section displays a variety of components depending on which of the tab buttons at the very top of the page is selected, they are discussed in more detail below.

displayed in yellow and acceptor photons under donor excitation in red. The y-axis scale can be adjusted using the slider on the left hand side of the view (not shown).

The Photon Arrival view, Figure 5.5, is primarily used for diagnostic purposes. It consists of two linked graphs, the top indicating the selected alternation period and the bottom a histogram of the photon arrival times relative to the alternation period of the laser. The histogram uses 1 µs bins and colours the bars using the same colour scheme as

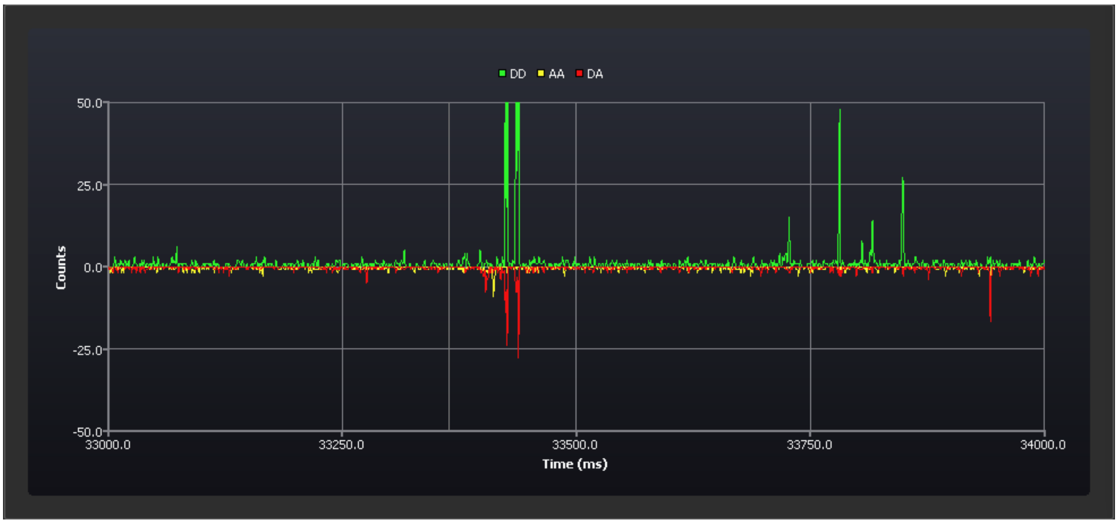


Figure 5.4.: The live trace component shows a close to realtime view of the photons detected on both APDs and their categorisation. Each of the 3 coloured traces shows the number of photons of a particular category detected within a particular millisecond time interval. The green trace shows the number of donor photons under donor excitation, the yellow acceptor photons under acceptor excitation and the red acceptor photons under donor excitation, i.e., FRET photons. Donor photons under acceptor excitation and photons detected when no lasers were enabled are not shown. A grey vertical line sweeps from left to right to indicate the current time. The red, yellow and green traces are overwritten as the grey current time indicator passes over them. In later versions of the program, the vertical scale may be adjusted using a slider on the left hand side.

the Live Trace graph, with the addition of grey 'noise' photons which indicate photons that have been detected but do not correspond to fluorescence (e.g., donor emission under acceptor excitation or detection when both lasers are off). The photon arrivals graph can reveal problems with laser stability as well as issues with turn on/off times of the lasers and light tightness.

Operators use the Z Focus view, Figure 5.6, to adjust the position of the objective relative to the sample to ensure that the front focal point of the objective is placed within the sample. The refractive index mismatch between the cover slip glass and the sample solution results in reflection of some of the illumination light back towards the objective. This back reflection is imaged onto the focusing camera, appearing as a circle. The focusing procedure consists of adjusting the extension of the z stage until the spot is as small as possible, indicating that the cover slip-sample interface is in focus. The user then moves the z stage ~20 µm up (i.e., the objective is brought ~20 µm towards

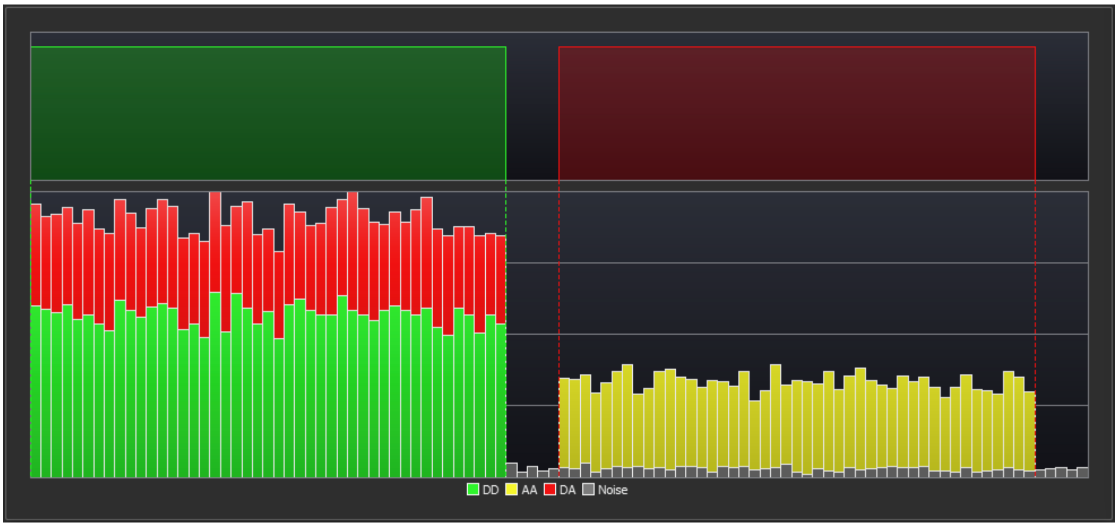
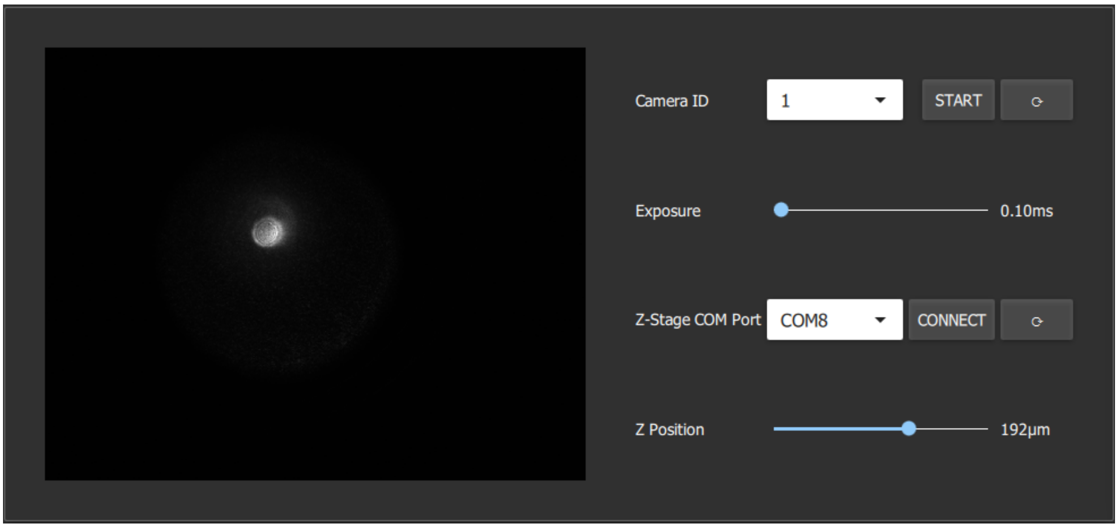


Figure 5.5.: The photon arrivals graph is a custom component showing a graphical representation of the currently selected laser alternation settings and a histogram of all the arrival times relative to the laser alternation. As in the Live Trace, green histogram bars indicate donor photons under donor excitation, the yellow acceptor photons under acceptor excitation and the red acceptor photons under donor excitation. Unlike the LiveTrace, donor photons under acceptor excitation and photons detected with no laser enabled are shown in grey. In the top section of the widget, the green area represents the donor excitation period and the red the acceptor.

the cover slip) to ensure that the illumination volume is well within the sample.

The initial design of the smfBox used a DCC1545M camera for focusing. This camera uses the UC480 SDK from Thorlabs which was used to develop smOtter. In the intervening time between the initial smfBox designs and the release of the designs to the wider academic community, Thorlabs discontinued the DCC camera range, replacing it with the ‘Zelux’ range. Unfortunately, the UC480 SDK isn’t compatible with the Zelux camera range and the replacement SDK isn’t compatible with the DCC range. The program was required to support both the old camera used by the smfBox and the new version that would need to be bought when building a new smfBox. A simple solution to this problem would have been to create an abstract class representing a common interface for interacting with a camera. An implementation of this interface could then be written for both cameras, allowing the program to select the relevant implementation at runtime. This would solve the immediate issue, however, suffers from a potential drawback: the program would be required to link against both SDKs, requiring both to be installed on the client’s computer for the program to run. This may cause an issue if Thorlabs



Camera ID 1

Exposure 0.10ms

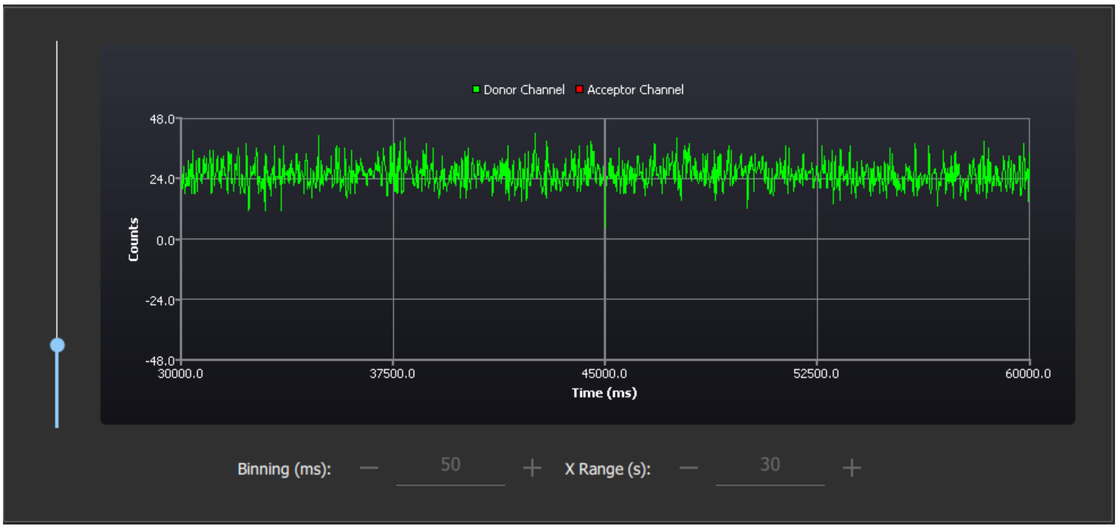
Z-Stage COM Port COM8

Z Position 192µm

Figure 5.6.: The z-focus view allows the user to adjust the objective height to ensure that the confocal illumination volume is placed well within the sample. The region on the left half of this view is the image of the back reflection from the coverslip-sample interface as seen by the camera. The dropdown menus on the right hand side allow the user to select the identification number of the camera in case more than one is connected to the computer (top) and the serial port corresponding to the z-stage controller (bottom). The values available from each dropdown can be refreshed using the refresh buttons to the right of the dropdowns in the case that the program was started before the devices were connected to the computer. The top slider allows the user to set the exposure time of the camera and the bottom adjusts the position of the objective, with the objective moving closer to the coverslip as the slider is moved to the right.

stopped shipping the old SDK with the ThorCam software in future releases or if cameras from other manufacturers needed to be supported in future. A solution, inspired by the structure of µManager, was implemented instead. Rather than the interface being described by an abstract class, it was instead described as a collection of C functions, which are listed in Appendix F. To provide support for a particular camera, a separate C++ project can be created. The project is required to produce a DLL which provides implementations for all the functions outlined in Appendix F for the camera of interest. At runtime, the smfCameraModules runtime library can scan a directory containing the DLLs for all the supported cameras and attempt to load each one. If dependencies for a particular DLL aren't found then loading of that DLL will fail but it will not crash the smOtter program. Camera module DLLs were written for the DCC, Zelux and the CellCam cameras.

Alignment is performed using the Alignment view (Figure 5.7). This view is conceptually similar to the Live Trace view, however, doesn't differentiate between acceptor photons arriving under donor and acceptor excitation, has a larger range of values for the y-axis scale and allows for user specification of the binning and x range settings. When placing a bright fluorescent sample on the smfBox, a user can use the alignment view to see the effects on detected signal intensity as a result of manoeuvring the various components of the system.



Binning (ms): — 50 +
X Range (s): — 30 +

Figure 5.7.: The alignment view is used during the alignment of the smfBox. It shows similar information to the Live Trace view, however, doesn't distinguish between acceptor photons under acceptor excitation and acceptor photons under donor excitation; both are combined into the red trace (not visible). Additionally, the vertical axis range is larger to account for the higher photon rates typical during alignment and the user can select both the binning and horizontal axis range.

There are also tabs for configuring the NIDAQ board and an about page (not pictured). The NI Card Settings view allows an end user to select the physical ports connected on the board as well as several other settings. This allows labs building their own smfBox to make use of smOtter even if the exact parts used aren't identical. The about view shows information about the software and its development.

5.4. Discussion

5.4.1. smOtter and the smfBox

The smfBox and smOtter software were published in a 2020 paper [84], which demonstrated the system's ability to perform smFRET measurements and the smfBox's efficacy in performing smFRET experiments had been previously demonstrated in a 2018 study comparing multiple commercial and custom built smFRET microscopes [86]. The smOtter program provides a viable alternative to the original LabVIEW control software, incorporating several workflow and usability improvements. The software quickly replaced the LabVIEW programs as the preferred method of interacting with the system within the lab and is used when onboarding all new users and collaborators. During a break to the PhD, the software was extended to support the spin-out of Exciting Instruments (see Section 5.4.6 for an overview of the work undertaken).

The choice of C++ for the project allowed the fine grain control of memory management required for reliable high-speed data acquisition whilst providing the higher level language constructs required for robust application development. The main drawback of C++ is that C++ coding ability is a relatively niche skill compared to more popular languages like Python [136], [161]. Unfortunately, Python's garbage collector doesn't provide the control over memory management required. Garbage collectors generally need to pause the application program in order to clean up memory, such a pause at the wrong time during an acquisition would likely cause the timestamps buffer to overflow. Similar issues with garbage collection would likely prevent the use of JVM or .NET languages, however, their superior performance versus Python may have gone some way to offset this issue. Another language currently gaining popularity in the programming community is Rust [116]. Rust's memory model is similar to that of C++'s and would provide sufficient control to prevent buffer overflows and the additional safety features provided by the Rust compiler may have helped improve program stability during the development process. At the start of the project, there were few feature complete GUI frameworks written in Rust. Interacting with wrapped C/C++ GUI libraries and writing a wrappers for the NIDAQ and camera libraries would likely have slowed development significantly and negated many of the safety benefits provided by Rust.

The Qt framework provides a feature complete GUI toolkit including several components often missing from other toolkits (e.g., graphing capabilities) as well as several integrated utility components (e.g., serial ports) [162]. Qt has existed since the 1990s, is open source, and has been employed across a large range of sectors. This means that a wide array of tutorials and help is available online that would allow labs with the prerequisite C++ knowledge to quickly familiarise themselves with toolkit in order to extend smOtter if required. Implementing the GUI as a QML project allowed the implementation of a modern looking GUI and provided an in-built dark theme that can help in reducing eye strain in dark lab environments. The program was always intended to be made open source and was eventually released under the Massachusetts Institute of Technology (MIT) license so there were no concerns about Qt's licensing initially, however, Qt's licensing did cause an issue for the commercial version of the software. The majority of Qt's components are licensed under the GNU Lesser General Public License (LGPL) V3 license so it is possible to keep application source code that uses these components closed source without breaking the terms of the license. Unfortunately, some of the components, most notably the graphing components, are licensed under the more restrictive GNU General Public License (GPL) license, which prevents any application code from being kept closed source. Exciting Instruments have been required to pay for a commercial license for Qt in order to ship their variant of the software. The structure of the software attempted to enforce a clear dividing line between the GUI components and the non-GUI components (e.g., acquisition, data export, etc.). As a result it should be relatively simple to replace the Qt components of the software with an alternative GUI toolkit without interfering with the main operation of the program.

The method of passing photon timestamps between threads presented in Section 5.3.2 was designed to handle photon data rates typical to smFRET experiments. While it performs well under these conditions, attempts to use the software to perform FCS measurements where the data rates are considerably higher, revealed the limitations of the approach. It is widely understood in the C++ community that the data layout of `std::list` is poorly optimised for modern CPU architectures. The reason for this is that modern CPUs make heavy use of caching to improve data throughput. If a value is present in the registers of the CPU it can be operated on immediately, if not the value must first be transferred from memory into a CPU register, while this transfer is happening the CPU core can't perform calculations[1]. To help mitigate this issue, modern CPUs employ several layers of caching to store recently loaded values physically closer to the CPU registers to reduce data access times by ensuring the data doesn't need to be moved all the way from main memory. Since processing arrays of contiguous data is so common, upon a cache miss, data is loaded into the cache from main memory in blocks. Processing data stored in a contiguous data structure (e.g., `std::vector` in C++) will

[1] This is a slight oversimplification. Modern CPU architectures utilise techniques such as operation pipelining and instruction reordering to allow the CPU to perform some calculations while awaiting a memory transfer, however, the intuition that awaiting memory transfers increases the time it takes for the CPU to perform calculations still holds

require less data access time than non-contiguous data structures (e.g., `std::list`) since many of the values will already be pre-cached.

While this effect was definitely present and limited the maximum photon rates that could be detected without the timestamps buffer overflowing, the primary difficulties were experienced at the end of acquisition, where the program would hang for several minutes before allowing another acquisition to take place. The problem is caused by two factors, firstly, a lessor known issue with memory allocation and deallocation when using `std::list` and secondly, the high compression level used. Unsurprisingly, higher compression levels take longer to process and with the data rates typical of high concentration FCS, smOtter will not be able to compress all the data and save it to disk before the next set of data arrives. This results in a backlog of data that still needs saving to disk at the end of acquisition and the program stalls while waiting for the backlog to clear.

The C++ standard effectively requires that an individual memory allocation be performed for every element of a `std::list`, as opposed to a `std::vector` which is expected to allocate at least enough space to store all its elements in a single allocation. Since each dynamic allocation requires an operating system call, performing many small allocations tends to be far less efficient than performing one large allocation and by extension, creating a `std::list` of size N tends to take longer than creating a `std::vector` of the same size. This effect also reduces the efficiency of the 64-bit timestamp extension code but not enough to cause a problem at typical FCS data rates. The same effect occurs when the list must be deallocated at the end of the experiment. Since the data isn't removed from memory during acquisition (see Section 5.4.4) this requires one deallocation for every detected photon. A hybrid approach utilising both `std::list` and `std::vector` can be used to improve the performance of this method, while still benefiting from the short critical sections resulting from the use of `std::list`. This approach was taken when writing the FCS extensions to smOtter for Exciting Instruments. The commercial version of the software can handle the maximum data rates of the APDs without the timestamps buffer overflowing and without hanging at the end of experiments with high photon rates.

5.4.2. Photon Timestamping

The NIDAQ boards provide a cheap and relatively easy way to both generate the required control signals and timestamp the output pulses from the APDs, however, they're not perfectly suited to the task. As mentioned previously, the counters used to timestamp photon arrival on the NIDAQ boards are only 32-bit, restricting the maximum length of an experiment to $\sim 43\,\mathrm{s}$ when counting the 100 MHz timebase of NIDAQ board, so a larger counter size is required. The next largest standard integer size available on most modern CPU architectures is 64-bit, which allows a maximum experiment length of ~ 5846 years. It is possible to daisy-chain the counters on a NIDAQ board to create a counter with an effective size of 64-bits, however, doing so would use all available counters on the board. Extension in software has two drawbacks. Firstly, it increases the amount of processing that needs to be applied to each photon timestamp as it arrives and by extension the chance of overflowing the timestamps buffer. Secondly, it relies on receiving at least one photon every 43 s to correctly detect rollover of the 32-bit counter, though in practice, the dark counts of the APDs will ensure that this is always the case. While the smfBox is designed to perform 2 colour smFRET experiments, requiring only 2 APDs, there are experiments that would require more detectors. For example, by introducing a polarising beam splitter before the emission dichroic and introducing a second identical pair of APDs and accompanying optical components, it is possible to simultaneously perform fluorescence anisotropy measurements [163]. Alternatively, smFRET experiments using more than 2 fluorophores can be performed. In the case of 3 fluorophores, a third APD could be introduced to specifically detect fluorescence from the third fluorophore [164]. Both these cases would require a second NIDAQ board if the extension to 64-bit was performed in hardware by daisy-chaining counters but only require software additions when extension is performed in software.

In principal, it should also be possible to perform experiments that require more than 4 counters by synchronising multiple NIDAQ boards. It would require sharing two signals from the primary board to the secondary board, the start trigger (`/do/StartTrigger`) and the 100 MHz clock. The counters on the primary board would be set up the same as in the single board version. The start trigger and clock signals of the primary board would be internally routed to two of the Programmable Function Input (PFI) pins where they would be externally connected to two PFI pins on the secondary board, preferably via coax cable to improve signal integrity. The counters on the secondary board could then be set to use the signals from the primary board via the appropriate PFI ports.

An extra counter could also be used to help solve one of the more difficult issues with handling live streams of photon arrival times. The majority of the realtime analysis performed in order to generate smOtter's displays require instantaneous photon rates. Since this data is not usually available, the number of photons arriving in a particular time interval (usually 1 ms) is used as a proxy. This presents no difficulties when analysis is performed once all data have been acquired, however, when performing the analysis in realtime, two extra questions must be answered: *will there be any more photons placed* *in this bin in future?* and the closely related *is this bin empty because the data has not arrived at the computer yet or because there were no photons in that time period?*.

The fact that the photon timestamps for a given channel are provided by the DAQmx SDK to application software in chronological order allows the program to answer the questions some of the time, if a photon has been placed in the n^{th} bin, then bins in the range $[1, n)$ are guaranteed to be complete. Unfortunately, no guarantees can be made about the completeness of the final bin into which a photon has been placed, even if the application exactly on a millisecond boundary, any photons detected by the NIDAQ board may not have been transferred to the computer yet. Analyses such as that performed to populate the ES histogram, are able to avoid this issue by only processing up to the $(n-1)^{\mathrm{th}}$ bin. This is because their analysis only results in a visible change when the number of photons within a bin passes a non-zero threshold. For figures such as the Live Trace, which are required to update continuously, even when no photons have been detected in a particular bin, things are more complicated. Taking the minimum number of counts per second to be 50 (the approximate minimum rated dark counts for the APDs used on the smfBox) the expected interval between counts would be 20 ms, though could vary anywhere between 10 ns and 92 ms with 99% probability. If the live trace were to use the last detected photon time as an indicator of the current time relative to the start of the experiment, then the current time marker would increment by random intervals whenever it was updated if the photon rates at the detectors were low. To be able to render these components, the software needs a steadily increasing measurement of the current time relative to the start of the experiment. Currently, smOtter uses the operating system clock for this purpose, however, this isn't an ideal solution. The primary issue is that the photon arrival times are measured against the 100 MHz clock on the NIDAQ board, whereas the GUI updates are measured using host computer's processor clock. These two clocks aren't synchronised in any way so their measured times will inevitably drift apart over the course of an experiment. Drift whereby the NIDAQ clock runs faster than the CPU clock doesn't cause an issue, however, drift in the opposite direction causes major graphical glitches whereby the photon graph lags the current time indicator with the lag increasing over time. The mitigation currently employed in smOtter is to deliberately run the Live Trace display with a 500 ms lag, however, since the drift will generally increase the discrepancy between the two clocks, sufficiently long experiments will still suffer the graphical glitches. The DAQmx SDK doesn't provide any way to directly query the time since the start of a task as measured by the NIDAQ's clock, however, a spare counter could be set up to count the 100 MHz timebase in 'on demand' readout mode. The software could periodically read this counter to determine the current experiment time according to the NIDAQ board, thus providing the software with a measurement of experiment time that does not drift relative to the photon timestamps.

While sufficient for smFRET measurements, the timestamp resolution is too low to determine information about fluorescence lifetime, which is typically of the order of nanoseconds for fluorescent labels used in microscopy [165]. Fluorescence lifetime measurements typically require a temporal resolution of the order of tens of picoseconds in order to resolve small differences in lifetimes due to differing environments surrounding fluorophores in the sample [90]. The timestamping topology used in the NIDAQ boards uses a counter which counts the rising edges of the board's internal clock in order to determine the arrival time of the photon. While robust and easy to implement, this topology's resolution is limited by the system clock used by the FPGA. To achieve a a resolution of even 1 ns using a counter based topology would require a clock frequency of 1 GHz, far higher than supported by even high end FPGAs [166]. Achieving subnanosecond resolutions requires fundamentally different circuit topologies such as that available in the HydraHarp 400 (PicoQuant, Berlin, Germany), which can provide a resolution of 1 ps. Given the expense of the HydraHarp, the question arises as to whether it would be possible to develop custom circuitry to perform these measurements on the smfBox in order to keep the cost of the system as low as possible. In [166], the authors discuss several common methods for implementing high resolution TDCs on FPGA platforms. Most of the topologies rely on exploiting deviations from the ideal performance of theoretical digital circuits (e.g., propagation delays, rise/fall times, etc.), which are generally difficult to control on general purpose FPGA hardware, making many of the topologies better suited to Application Specific Integrated Circuit (ASIC) implementation. The Tapped Delay Line (TDL) architecture, however, is relatively amenable to implementation on FPGA, and several implementations have reported resolutions of the order of tens of picoseconds or less. In [90], the authors report a resolution of 30 ps using a TDL architecture implemented on a KC705 Evaluation Kit (Xilinx, California, USA) which can be purchased for ~\$2,500. The implementation is fully open source so could presumably be extended to provide the laser control signals required for performing smFRET experiments. Another potentially interesting avenue of investigation would be the use of commercial TDC chips, such as the MAX35101 (Maxim Integrated, California, USA). The MAX35101 provides a resolution of 20 ps and costs less than £10, however, it may not be capable of handling the high data rates associated with photon timestamping datasets. Irrespective of the TDC circuitry employed, measurement of fluorescence lifetime would require the addition of pulsed lasers to the system, likely increasing build cost significantly.

5.4.3. Laser Alternation and Stability

Typically, when performing smFRET with ALEX, the laser alternation is symmetrical, i.e., the donor and acceptor lasers are enabled for the same amount of time [167]. The smOtter software defaults to a symmetrical alternation profile of 45 µs in each channel, allowing for a 5 µs spacing to let each laser to turn off fully before the next laser is turned on with a 100 µs alternation period. A shorter alternation period would improve the ability of the offline processing to account for the effects of photobleaching as the molecule drifts through the illumination volume. The lasers, having a fixed rise/fall times of the order of a microsecond [157], provide the practical lower limit of alternation period. Since overlapping of laser illumination presents complications for analysis, each laser must turn off before the next laser can be turned on. As the alternation period is decreased, the proportion of the alternation cycle spent with both lasers off must increase, reducing the total illumination time of each molecule and therefore the total data collected. The Omicron lasers used on the smfBox have a second 'analogue alternation' input that could in principal be used to reduce the laser's rise/fall times, however, this would come at the cost of a much lower extinction coefficient of 1000:1 as opposed to the infinite extinction coefficient achieved by completely turning the lasers on and off. 100 µs provides a good compromise between these two factors.

When performing ALEX, illumination with the donor laser provides the information about FRET efficiency, the acceptor illumination is only used to determine if the molecule was successfully labelled with both fluorophores [76]. It stands to reason then that there may be some benefit to employing asymmetric ALEX, prioritising donor excitation over acceptor excitation within the alternation cycle. In [84] it was shown that varying the relative duty cycles of the two lasers may provide an avenue for optimisation of smFRET experiments. To allow the end user to optimise their experiments, smOtter allows the user to change both the alternation period and laser duty cycles, however, introduces some restrictions on the allowable values to ensure correct operation. The NIDAQ boards require application software to provide digital output waveforms as a series of samples. When creating the digital output task the application specifies the output rate, i.e., the reciprocal of the time each digital output value should remain on the output pin. When interacting with smOtter, the user supplies the desired duty cycle of each laser in whole percentage points, smOtter uses this information and the 'laser control resolution' from the NI Card settings tab to configure the data output. The laser control resolution determines the output rate and therefore the allowable values of the alternation period.

Specifically, two conditions must hold:

$$100T_{\text{LCR}} \leq T_{\text{ALEX}} \tag{5.1}$$

and

$$T_{\text{ALEX}} \bmod (100T_{\text{LCR}}) = 0 \tag{5.2}$$

where T_{ALEX} is the alternation period and T_{LCR} is the laser control resolution. Equation 5.1 ensures that the alternation period is sufficiently long for at least 100 samples and Equation 5.2 ensures that all 100 samples can be made to fit evenly into a single alternation cycle, potentially by repeating each sample multiple times. One final restriction

$$T_{ex} \bmod T_{\text{ALEX}} = 0 \tag{5.3}$$

where T_{ex} is the total experiment duration, ensures that the final alternation period is not cut off by the end of the experiment. No restriction is placed on the laser control resolution, however, care should be taken that it results in a sample rate that is an integer division of the NIDAQ board's 100 MHz clock to ensure that the waveform can be accurately produced. The software also allows the user to overlap the excitation, which can be useful for performing some techniques (e.g., Pulsed Acceptor Excitation), however, it should be noted that in this case, the realtime processing will assume all acceptor detector photons are a result of FRET in regions where the two lasers overlap.

As mentioned in Section 5.3.2, smOtter is capable of continuously sampling the output of the system's DET10A2 1 ns rise time photodetector with the aim of monitoring laser power fluctuations during the course of an experiment. Utilisation of this data has presented two issues thus far, firstly, the signal from the DET10A2 is unamplified and therefore difficult to measure on the NIDAQ boards. Secondly, the rate at which the data must be sampled increases the output file sizes significantly.

To reliably read from the DET10A2, its signal must be amplified. Section 5.3.2 outlined attempts to use a single resistor for this task, which failed, either due to a low current to voltage conversion gain or impedance mismatch, depending on the particular resistance used. An active transimpedance amplifier is therefore required to allow the NIDAQ board to read the output of the detector. The rated 1 ns rise time implies a bandwidth of 500 MHz would transmit the signal practically undistorted [168]. To calculate a rough requirement for the amplifier's gain, the the DET10A2 has a responsivity of approximately $0.239\,\mathrm{A\,W^{-1}}$ at 515 nm (the responsivity at 638 nm is higher), assuming a 1.5 mW incident power (15 mW though the 90:10 beamsplitter) the output of the DET10A2 would be ~360 µA. Amplifying this to a reasonable level for NIDAQ measurement, for arguments sake 1 V, would require a gain of ∼3,000. While certainly not an impossible task, a 500 MHz bandwidth, 3,000 gain amplifier would require multiple stages consisting of high bandwidth, low noise amplifiers. Additionally, since the DC level of the signal is important, stages could not be AC coupled and would therefore likely require adjustable biasing to prevent DC bias from prior stages being amplified. There are a couple of observations about the system that significantly reduce the bandwidth requirements, however. Firstly, the analogue inputs on the NIDAQ boards only have a 100 MHz bandwidth so the full 1 ns rise time of the detector can't be measured. Secondly, the maximum frequency present in the signal of interest is likely to be determined by the rise time of the lasers rather than the rise time of the detector. The rise time of the lasers when using the laser enable input is rated at '<2 µs' [157], if we assume that the rise time of a laser is unlikely to be less than 1 µs then a bandwidth of 500 kHz would be sufficient to pass the signal undistorted. A bandwidth of 500 kHz drastically reduces the gain bandwidth product requirements on the operational amplifiers. Since the smfBox was initially designed, Thorlabs have released amplified versions of the DET10A2. The PDA10A2 is the drop-in, amplified, replacement for the DET10A2 with a fixed amplification of $5\,\mathrm{kV\,A^{-1}}$ into a 50 Ω load. Using the incident power and responsivity calculated previously, this would result in a signal of 1.79 V. The PDA10A2 is rated with a lower, 2.3 ns rise time, however, this is still faster than can be detected with the 100 MHz bandwidth of the NIDAQ boards and the rise time of the lasers.

In order to get an accurate measurement of the laser power during an alternation cycle, it's desirable to sample the power multiple times while the laser is on, however, doing so increases the amount of data generated. By default, the software samples the voltage at 100 kHz or, alternatively, once every 10 µs. Since the analogue input task is also synchronised to the laser alternation, this ensures exactly 4 samples of each laser's power per alternation period. Analogue input tasks produce 64-bit floating point data so continuous measurement of the laser powers produces ∼0.8 megabytes of data per second. Over the course of an hour long experiment, this equates to ∼2.88 gigabytes of data, substantially more than the ∼10 megabyte file sizes typical without laser power measurement. The data is compressed before saving, so the size on disk will be smaller, however, since this has only ever been tested with the unamplified output of the DET10A2, the exact compression ratios that can be expected are unknown. A potential way to reduce the data size associated with laser power measurements is to only save measurements periodically. Recording 1 s worth of alternation periods every 1 min would reduce the resulting data size by a factor of 60, equivalent to ∼48 megabytes uncompressed over a 1 h experiment. Further reducing the amount of time sampled or the time between bursts of samples would allow increasing the NIDAQ's sampling rate to better resolve each alternation cycle whilst still maintaining reasonable alternation cycles. From a software perspective, the analogue input task could be left running continuously even when data isn't being saved, allowing for realtime monitoring of the laser alternations to warn the user of any detected laser power fluctuations.

5.4.4. Data Output

One of the major workflow improvements provided by smOtter over the LabVIEW software is the direct export to PhotonHDF5. When using the LabVIEW acquisition software, the data is written to a CSV file, which must be manually converted to PhotonHDF5 before it can be consumed by FRETBursts. During the course of development, it was decided that smOtter would consider all applicable PhotonHDF5 metadata fields mandatory. The software won't allow the user to start an acquisition until all metadata has been provided. The reason for this decision, is that while not required for analysis of the data, information such as the sample contents, date/time, laser powers, experimenter name, etc., are important for record keeping and open science. Often this data is written physical lab books, in non-standard locations and formats (e.g., file names) or omitted entirely, making it hard for future researchers to make use of the data. While, sometimes irritating, requiring this data ensures that best practice is always followed and ensures that the information is always available if needed.

While PhotonHDF5 is an open standard, specified as a series of data an metadata fields that should be contained within a regular HDF5 file, the authors did not appear to intend for third parties to implement their own PhotonHDF5 readers and writers, instead providing a Python library for reading and writing PhotonHDF5 and a tool (phconvert) for converting other common file formats into PhotonHDF5 [159]. While this library probably could have been used directly from smOtter by embedding a python interpreter, this is a complex process and prone to error. While easier in the long run, writing PhotonHDF5 using the standard HDF5 C++ library revealed an error in the provided PhotonHDF5 Python libraries than needed to be worked around. The Python library used by the PhotonHDF5 library to read HDF5 files (pytables) will interpret some data differently depending on whether or not it was saved from Python. Specifically, datasets consisting of a single string value are only loaded into a python string if certain metadata is present, otherwise it is loaded as a single element numpy array containing the string value. The PhotonHDF5 reader library doesn't take this into account, causing a crash when attempting to read datasets containing a single string, despite the files conforming to the PhotonHDF5 specification as written. The solution, is to write extra attributes to these datasets to make it appear that they were written by pytables. Three extra string attributes are required, ‘CLASS’, ‘FLAVOUR’ and ‘VERSION’ with values ‘ARRAY’, ‘python’ and ‘2.4’ respectively. FRETBursts makes the assumption that photon timestamps are monotonically increasing, which presumably holds for files that have been converted using the phconvert utility, however, does not for the PhotonHDF5 files produced by smOtter. The reason for this is that photons are read from the NIDAQ in blocks, where each block contains only photons from one channel. The timestamps within a block and the timestamps between blocks from the same channel are in chronological order, however, timestamps between blocks from different channels are almost guaranteed not to be. When the timestamps are combined to produce the timestamps dataset, they are written in the order of arrival and are therefore not in chronological order. For smOtter to ensure that the timestamps dataset is in chronological order, it would need to reload the file at the end of acquisition, sort the data and write it back to disk (sorting can’t be performed on the fly for the reasons discussed in Section 5.4.2). Since the PhotonHDF5 standard doesn’t require the timestamps dataset be in chronological order, the data is instead sorted in Python prior to performing FRETBursts analysis using the code in Listing 5.1.

```python
# Import FRETBursts library
from fretbursts import *

# Load the PhotonHDF5 file
d = loader.photon_hdf5(filename)

# Sort the photon arrival times into chronological order
for i in range(0, len(d.ph_times_t)):
    indices = d.ph_times_t[i].argsort()
    d.ph_times_t[i], d.det_t[i] =
        d.ph_times_t[i][indices], d.det_t[i][indices]

# Perform remaining analysis as usual...
```

Listing 5.1: Sorts photon arrival times into chronological order before FRETBursts performs its analysis. Both the photon timestamps dataset and the detectors dataset must be rearranged in the same way since the relationship between the two datasets is implicit.

In principal, once the photon data has been processed in order to produce the various displays in the GUI and saved to disk, it can be released from memory. Due to an oversight during development, the photon data is not released and so the program's memory usage will steadily increase during the course of an acquisition. This doesn't cause issues when performing smFRET, due to the relatively low photon rates, however, can rapidly fill up several gigabytes of memory during a high concentration FCS acquisition. Since memory is correctly freed at the end of acquisition, this will only cause an issue if the number of photons detected during a single acquisition is sufficiently high as to induce the operating system to cache memory pages to disk in order to free up space in main memory. When this happens, the application is temporarily paused, during which time the timestamps buffer usually overflows. This oversight was corrected in the commercial version of the software during the work for Exciting Instruments, allowing the program to prevent memory filling up even at the photon rates typical of high concentration FCS experiments.

5.4.5. Camera Modules

The smfCameraModules framework, used to interface with cameras used for focusing, allows the program to support multiple cameras without the end user needing to have every SDKs installed. It has the added benefit that labs wanting to provide support for different cameras could do so without needing to recompile the entire smOtter software package. The smfCameraModules library is completely standalone, requiring no dependencies, making development environment setup significantly easier than what is required for recompiling smOtter from scratch. The design of the smfCameraModules was inspired by µManager, but corrects a flaw in µManager's device adapter design.

The C and C++ programming languages have many similarities, with C++ often mistaken for a superset of the C language. While C++'s C interoperability is close to seamless, there are some subtle differences between the two languages. One of the differences is function overloading, which is supported in C++ but not in C. This allows multiple functions to have the same name, as long as they have different arguments; the C++ compiler will select the appropriate overload automatically based on the types of the parameters provided. While useful, the feature comes with a subtle drawback, namely that when functions are compiled and passed to the linker, they cannot be referred to by name as is the case in C. A C++ compiler must perform an extra step, 'name mangling', to disambiguate the multiple overloads during compilation and allow the linker to link against the correct variant. The C++ standard provides no direction on exactly what the resulting mangled names should be and as a result, C++ libraries compiled with different compilers aren't guaranteed to be compatible. To allow compilers to optimise for their particular platforms, the C++ standard also doesn't enforce the memory layout of classes or how their virtual method tables (vtables) should be implemented. While this doesn't cause issues most of the time, these factors complicate the process of dynamically loading libraries, as required by the smfCameraModules framework. Name mangling means that knowledge of the name of a function in the library's source code doesn't allow you to dynamically load it. The lack of fixed memory layout and vtable structure means that the compiler version used by the application code must match the version used by the library. Additionally, extreme caution must be exercised when mixing different versions of a C++ library, seemingly innocuous actions such as adding an extra method overload or a new virtual function may cause the compiler to rearrange a class' vtable, causing a break in the library's ABI. If application code links against an old version of a dynamically loaded library but a newer DLL is provided at runtime, the application will attempt to access functions using the old vtable layout, causing unpredictable behaviour. The C standard on the other hand, is much more prescriptive, enforcing a consistent and stable ABI across compilers. Breaking changes to library ABI can still cause unpredictable behaviour, however, the relative simplicity of C makes it easier for library developers to determine when a code change will break ABI compatibility.

When accessing code in a dynamic library, the function location within the library must be determined at runtime. If the specific library is known at compile time, the compiler can automatically insert the code to load the library and access its functions during the linking stage. If the specific library isn't known, as in the case of µManager's device adapters and smfCameraModules, then this code must be written manually. The process consists of loading the DLL into memory using its file name, loading pointers to the functions contained in the DLL by name, using those pointers to call the contained functions and finally unloading the DLL when it is no longer needed. This process cannot be performed directly on C++ DLLs due to the name mangling issues mentioned previously preventing the second step, so the DLL must export at least some functions that use the C ABI (this can be done using the `extern "c"` feature built into C++).

All µManager device adapter DLLs export 9 C ABI functions: `CreateDevice`, `DeleteDevice`, `GetDeviceDescription`, `GetDeviceInterfaceVersion`, `GetDeviceName`, `GetDeviceType`, `GetModuleVersion`, `GetNumberOfDevices` and `InitializeModuleData`. These functions allow the the MMCore library to retrieve information about the device support provided by the DLL, the version of the library the DLL was compiled for, as well as creating/destroying device handles. The `CreateDevice` function is expected to return a pointer to a C++ object that implements one of the micromanager device abstract classes. Interfacing with devices supported by the device adapter is performed via virtual method calls on that device pointer. Since cross-DLL-boundary virtual method calls are made, compatibility of the compiler that built the DLL with the compiler that built the MMCore library must be ensured. Since the DLL is loaded dynamically, there must be some function to ensure the library ABI is compatible with the version of MMCore. µManager uses a value referred to as the 'device interface version', if the device interface version provided by the DLL's `GetDeviceInterfaceVersion` function is different from that contained in the MMCore binary, MMCore will reject the library, assuming it to be incompatible. Unfortunately, since µManager uses virtual method calls, the ABI is a C++ ABI, which as discussed previously, makes it easy for the µManager developers to accidentally break ABI compatibility. The µManager developers therefore choose to err on the side of caution, incrementing the device interface version even if ABI stability probably hasn't been broken. Even so, there was at least one occurrence during the course of this PhD where a breaking ABI change was made without the device interface version being incremented, causing unpredictable behaviour with a custom device adapter.

Though it is a C++ library, the smfCameraModules library makes no cross-DLL-boundary C++ ABI calls. Camera modules export a number of C ABI functions (see Appendix F) that perform both module interaction (e.g., querying the interface version, connecting to cameras, etc.) and camera interaction, i.e., C functions are used in place of the virtual method calls in the µManager interface. To improve ease of use when accessing smfCameraModules from application code, the C API is wrapped by a series of C++ classes to provide idiomatic object oriented access to the code contained in the DLL. This approach has two advantages, firstly, since the DLL exports a C API, there is no need to ensure any compiler compatibility. In fact, since the handles passed between the DLL and application code are opaque, the camera module need not be implemented in C++ at all. As long as the language is capable of producing a DLL that conforms to the interface, then application code will be able to access it as if it is written in C++. Other than C++, the primary languages expected to be used to implement camera modules would be C and Rust, however, it may be possible to use other languages such as Python (via technologies like Cython [169]). A request was raised to enable writing µManager device adapters in Python at the 2020 Software for Microscopy Workshop, held at the Janelia Research Campus, however, µManager's current structure makes this difficult.

The second advantage of the approach taken by the smfCameraModules library is that changes to the API have more predicable effects on the library's ABI. Because of this, the smfCameraModules library employs semantic versioning to determine compatibility rather than a single number as used in µManager. The version designators used in the smfCameraModules library consist of 3 numbers, representing the major, minor and patch versions of the library. When making changes such as bug fixes or changes to the C++ wrapper classes, the patch number is incremented. Non-breaking API changes, such as adding new functions, is represented by incrementing the minor version and breaking API changes, such as changing a function signature, is indicated by incrementing the major version. The smfCameraModules library can ignore differing patch numbers, accept modules with minor versions greater than or equal to its own and reject libraries with differing major versions.

A disadvantage of the smfCameraModules library's approach versus that used by µManager is that the interface must be implemented as C functions as opposed to using inheritance. This complicates implementation slightly, as C++ types such as objects and exceptions must be converted to C compatible equivalents. In principal, this complexity could be completely removed by implementing a 'device support library' that is capable of automatically converting an implementation of a C++ abstract class into a series of C functions that conform to the smfCameraModules interface.

5.4.6. Exciting Instruments

This Chapter has so far primarily discussed the development of smOtter and the smfBox, both of which have been made freely available and open source. The aim of the smfBox project, was the help widen access to smFRET techniques by providing the community with instruction on how to build a smFRET capable microscope for a price within the kinds of budgets available to most labs. Exciting Instruments was spun out in order to develop the smfBox into a commercially viable product, that would enable labs without the technical expertise to build a smfBox to benefit from the ability to perform smFRET techniques. The aim of the company's first product, the EI-FLEX, was to provide a microscope, capable of performing both 2-colour confocal smFRET and FCS experiments within a bench-top, class 1 laser product. The company decided that the best way to provide software support for the instrument was by extending smOtter. A 3 month break from the PhD was taken to help facilitate this development and address some of the issues with smOtter mentioned in this chapter as well as to help with some further hardware development.

The development of the software was primarily focused on the addition of FCS specific features, such as live photon correlation calculations. It also involved improving the performance of the software under high photon count rates, bringing control of laser powers into the software (previously, laser powers were controlled via the Omicron laser bank software) and helping with integration of the software with the Exciting Instruments cloud analysis platform. To facilitate the additions, the backend was augmented to provide extensible support for multiple acquisition types and the GUI was tweaked to streamline and simplify the user experience.

5.5. Conclusion

This chapter has discussed the development of the smOtter software. The software provides a user friendly, open source, C++ interface for interacting with the open source smfBox microscope. A method for utilising National Instruments DAQ boards to perform single photon timestamping with a resolution of 10 ns and effectively arbitrary experiment lengths was presented and contrasted to alternative approaches. The software's current support for continuous measurement of laser powers and the steps required to make this useful were both outlined. Output of data generated by the program to the PhotonHDF5 file format and the required workarounds were also discussed. Dynamic loading of camera support is provided via a µManager inspired interface, which addresses some of the drawbacks of the device adapter model used by µManager. Finally, a brief outline of the additions made to the software to support the spin out of Exciting Instruments from the University of Sheffield was given.

CHAPTER

SIX

CONCLUSION

This thesis has described work towards the development of two novel microscopy systems, the CairnFocal, a DMD based microscope capable of multi-modality imaging, and the smfBox, an open-source confocal smFRET microscope.

Chapter 2 detailed the hardware and software development that laid the groundwork for the microscopy experiments in the future chapters. It described the development of standalone control software for the DMD, which enables a user to configure the system to perform a variety of microscopy techniques. The control software contains a small TCP/IP server, which can be used to allow other programs to control the currently displayed patterns. An alternative method to control the DMD entirely from within µManager was provided by writing a device adapter to control the DMD and a plugin to allow an end user to illuminate specific regions by drawing ImageJ ROIs. FPGA based hardware was designed and built to perform synchronisation between the DMD and the rest of the microscope system. Support for the device was integrated into the CairnFocal Control software to allow end users to perform more complex imaging techniques without needing to think about synchronisation issues and to prepare for fully automatic modality switching. A device adapter was written for the LDI laser bank from 89-North. This device adapter was integrated into the µManager project and it is now used by groups all across the globe.

In Chapter 3, the implementation of several common microscopy techniques on the CairnFocal was detailed. The label free techniques brightfield, darkfield and phase contrast as well as the fluorescent techniques widefield, SDCM and ISM were implemented and their performances analysed. The implementation of confocal was shown to improve optical sectioning when compared to widefield. Reducing the size of the pinholes and increasing the space between them was demonstrated to further improve optical sectioning at the expense of reducing detected signal levels. A µManager plugin was developed to allow the collection of multi-dimensional ISM data and an ImageJ plugin was produced to perform reconstruction. Two methods for determining the location of pinholes in regions of low fluorescence were described and the importance of adequate background correction shown. The final performance of the ISM implementation was demonstrated to be roughly in line with the performance of other implementations in the literature. To support the analyses performed in this chapter, an ImageJ plugin was developed to allow the automatic analysis of multiple PSFs from images of beads.

Chapter 4 described some applications of the system to live cell imaging, where light intensities associated with super-resolution techniques cause problems for sample longevity. The chapter focused on imaging human neutrophils and amoeba where both long term imaging and high resolution were required. In these situations, modality switching provides the CairnFocal with the ability to meet both these requirements. The development of a µManager plugin for automatic modality hopping was outlined and the potential extension of this plugin to provide fully autonomous imaging was discussed. The neutrophil imaging described in this section culminated in a pre-print publication.

Finally, Chapter 5 concerns the development of the supporting software for the smfBox. The implementation of single photon counting using NIDAQ boards was explained as well as the advantages and disadvantages of this approach. The structure and features of the program were described with particular emphasis placed on the multi-threaded handling of photon data and the method for supporting multiple cameras. In discussion of the latter, a comparison to the design of device adapter interface of µManager is drawn, with the smfCameraModules interface providing significant benefits over the methods employed by µManager. The program integrates all of the features of the original LabVIEW control program into a single user friendly package, which is used by the Craggs Lab and their collaborators. The program was made open-source and formed a significant part of a publication in Nature Communications. When Exciting Instruments was spun out from the University of Sheffield, the software formed a large part of the intellectual property licensed to the company by the University. A short break in the PhD was taken in order to support Exciting Instruments in the further development of the software. This extended version of the program is now used as the acquisition software for the company's EI-FLEX microscope.

BIBLIOGRAPHY

[1] E. Hecht, *Optics* (Always learning). Pearson, 2016, ISBN: 9781292096933. [Online]. Available: https://books.google.co.uk/books?id=Bv1RrgEACAAJ.

[2] R. Hooke, *Micrographia*. BoD–Books on Demand, 2020.

[3] T. Tkaczyk, *Field Guide to Microscopy*. SPIE, 2010.

[4] F. Zernike, "Phase contrast, a new method for the microscopic observation of transparent objects," *Physica*, vol. 9, no. 7, pp. 686–698, 1942, ISSN: 0031-8914. DOI: https://doi.org/10.1016/S0031-8914(42)80035-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S003189144280035X.

[5] K. F. WEBB, "Condenser-free contrast methods for transmitted-light microscopy," *Journal of Microscopy*, vol. 257, no. 1, pp. 8–22, 2015. DOI: https://doi.org/10.1111/jmi.12181. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jmi.12181. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/jmi.12181.

[6] G. G. Stokes, "Xxx. on the change of refrangibility of light," *Philosophical Transactions of the Royal Society of London*, vol. 142, pp. 463–562, 1852. DOI: 10.1098/rstl.1852.0022. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rstl.1852.0022. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rstl.1852.0022.

[7] . T. Scientific, *The molecular probes handbook*, https://www.thermofisher.com/uk/en/home/references/molecular-probes-the-handbook.html.

[8] *Education in microscopy and digital imaging*. [Online]. Available: https://zeiss-campus.magnet.fsu.edu/articles/basics/fluorescence.html (visited on 11/30/2022).

[9] A. H. Coons, H. J. Creech, and R. N. Jones, "Immunological properties of an antibody containing a fluorescent group.," *Proceedings of the Society for Experimental Biology and Medicine*, vol. 47, no. 2, pp. 200–202, 1941. DOI: 10.3181/00379727-47-13084P. eprint: https://doi.org/10.3181/00379727-47-13084P. [Online]. Available: https://doi.org/10.3181/00379727-47-13084P.

[10] O. SHIMOMURA, "The discovery of aequorin and green fluorescent protein," *Journal of Microscopy*, vol. 217, no. 1, pp. 3–15, 2005. DOI: https://doi.org/10.1111/j.0022-2720.2005.01441.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.0022-2720.2005.01441.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.0022-2720.2005.01441.x.

[11] E. Ridgway and C. Ashley, "Calcium transients in single muscle fibers," *Biochemical and Biophysical Research Communications*, vol. 29, no. 2, pp. 229–234, 1967, ISSN: 0006-291X. DOI: https://doi.org/10.1016/0006-291X(67)90592-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0006291X6790592X.

[12] R. Y. Tsien, "New calcium indicators and buffers with high selectivity against magnesium and protons: Design, synthesis, and properties of prototype structures," *Biochemistry*, vol. 19, no. 11, pp. 2396–2404, 1980.

[13] M. Chalfie, Y. Tu, G. Euskirchen, W. W. Ward, and D. C. Prasher, "Green fluorescent protein as a marker for gene expression," *Science*, vol. 263, no. 5148, pp. 802–805, 1994. DOI: 10.1126/science.8303295. eprint: https://www.science.org/doi/pdf/10.1126/science.8303295. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.8303295.

[14] J. Huff, "The airyscan detector from zeiss: Confocal imaging with improved signal-to-noise ratio and super-resolution," *Nature methods*, vol. 12, no. 12, pp. i–ii, 2015.

[15] K. Morimoto, A. Ardelean, M.-L. Wu, *et al.*, "Megapixel time-gated spad image sensor for 2d and 3d imaging applications," *Optica*, vol. 7, no. 4, pp. 346–354, Apr. 2020. DOI: 10.1364/OPTICA.386574. [Online]. Available: https://opg.optica.org/optica/abstract.cfm?URI=optica-7-4-346.

[16] K. Morimoto, J. Iwata, M. Shinohara, *et al.*, "3.2 megapixel 3d-stacked charge focusing spad for low-light imaging and depth sensing," in *2021 IEEE International Electron Devices Meeting (IEDM)*, 2021, pp. 20.2.1–20.2.4. DOI: 10.1109/IEDM19574.2021.9720605.

[17] E. Abbe, "Beiträge zur theorie des mikroskops und der mikroskopischen wahrnehmung," *Archiv für mikroskopische Anatomie*, vol. 9, no. 1, pp. 413–468, 1873.

[18] J. Goodman, *Introduction to Fourier Optics* (McGraw-Hill Series in Electrical and Computer Engineering: Communications and Signal Processing). McGraw-Hill, 1996, ISBN: 9780070242548. [Online]. Available: https://books.google.co.uk/books?id=QllRAAAAMAAJ.

[19] L. Rayleigh, "Xv. on the theory of optical images, with special reference to the microscope," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 42, no. 255, pp. 167–195, 1896.

[20] H. Osterberg, "Microscope imagery and interpretations," *JOSA*, vol. 40, no. 5, pp. 295–303, 1950.

[21] C. M. Sparrow, "On spectroscopic resolving power," *The Astrophysical Journal*, vol. 44, p. 76, 1916.

[22] C. Sheppard, I. J. Cox, and D. Hamilton, "Edge detection in micrometrology with nearly confocal microscopy," *Applied Optics*, vol. 23, no. 5, pp. 657–658, 1984.

[23] H. H. Hopkins, "The concept of partial coherence in optics," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 208, no. 1093, pp. 263–277, 1951.

[24] P. H. van Cittert, "Die wahrscheinliche schwingungsverteilung in einer von einer lichtquelle direkt oder mittels einer linse beleuchteten ebene," *Physica*, vol. 1, no. 1-6, pp. 201–210, 1934.

[25] F. Zernike, "The concept of degree of coherence and its application to optical problems," *Physica*, vol. 5, no. 8, pp. 785–795, 1938.

[26] H. H. Hopkins, "On the diffraction theory of optical images," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 217, no. 1130, pp. 408–432, 1953.

[27] P. Sarder and A. Nehorai, "Deconvolution methods for 3-d fluorescence microscopy images," *IEEE Signal Processing Magazine*, vol. 23, no. 3, pp. 32–45, 2006.

[28] D. Sage, L. Donati, F. Soulez, *et al.*, "Deconvolutionlab2: An open-source software for deconvolution microscopy," *Methods*, vol. 115, pp. 28–41, 2017.

[29] E. Sekko, G. Thomas, and A. Boukrouche, "A deconvolution technique using optimal wiener filtering and regularization," *Signal processing*, vol. 72, no. 1, pp. 23–32, 1999.

[30] C. Preza, M. I. Miller, L. J. Thomas, and J. G. McNally, "Regularized linear method for reconstruction of three-dimensional microscopic objects from optical sections," *JOSA A*, vol. 9, no. 2, pp. 219–228, 1992.

[31] W. Carrington, K. Fogarty, and F. Fay, "3d fluorescence imaging of single cells using image restoration," *Modern cell biology*, vol. 9, pp. 53–72, 1990.

[32] T. J. Holmes, "Maximum-likelihood image restoration adapted for noncoherent optical imaging," *JOSA A*, vol. 5, no. 5, pp. 666–673, 1988.

[33] W. H. Richardson, "Bayesian-based iterative method of image restoration," *JoSA*, vol. 62, no. 1, pp. 55–59, 1972.

[34] L. B. Lucy, "An iterative technique for the rectification of observed distributions," *The astronomical journal*, vol. 79, p. 745, 1974.

[35] F. Soulez and M. Unser, "Superresolution with optically-motivated blind deconvolution," in *Laser Applications to Chemical, Security and Environmental Analysis*, Optical Society of America, 2016, JT3A–38.

[36] J. Jonkman and C. Brown, "Any way you slice it-a comparison of confocal microscopy techniques," *Journal of Biomolecular Techniques*, vol. 26, no. 2, pp. 54–65, 2015.

[37] P. M. Carlton, J. Boulanger, C. Kervrann, *et al.*, “Fast live simultaneous multiwavelength four-dimensional optical microscopy,” *Proceedings of the National Academy of Sciences*, vol. 107, no. 37, pp. 16 016–16 022, 2010, ISSN: 0027-8424. DOI: `10.1073/pnas.1004037107`. eprint: `https://www.pnas.org/content/107/37/16016.full.pdf`. [Online]. Available: `https://www.pnas.org/content/107/37/16016`.

[38] N. Callamaras and I. Parker, “Construction of a confocal microscope for real-time xy and xz imaging,” *Cell calcium*, vol. 26, no. 6, pp. 271–279, 1999.

[39] C. Sheppard and A. Choudhury, “Image formation in the scanning microscope,” *Optica Acta: International Journal of Optics*, vol. 24, no. 10, pp. 1051–1073, 1977. DOI: `10.1080/713819421`. eprint: `https://doi.org/10.1080/713819421`. [Online]. Available: `https://doi.org/10.1080/713819421`.

[40] I. J. Cox, C. J. Sheppard, and T. Wilson, “Improvement in resolution by nearly confocal microscopy,” *Applied optics*, vol. 21, no. 5, pp. 778–781, 1982.

[41] C. Sheppard, “Super-resolution in confocal imaging,” *Optik - International Journal for Light and Electron Optics*, vol. 80, p. 53, Feb. 1988.

[42] G. Xiao and G. Kino, “A real-time confocal scanning optical microscope,” in *Scanning Imaging Technology*, SPIE, vol. 809, 1987, pp. 107–113.

[43] G. Xiao, G. Kino, and B. Masters, “Observation of the rabbit cornea and lens with a new real-time confocal scanning optical microscope,” *Scanning*, vol. 12, no. 3, pp. 161–166, 1990.

[44] B. R. Masters, *Handbook of Biological Confocal Microscopy, Third Edition.* Springer US, 2008, ISBN: 9780387259215. DOI: `10.1117/1.2911629`. [Online]. Available: `https://doi.org/10.1117/1.2911629`.

[45] T. Tanaami, S. Otsuki, N. Tomosada, Y. Kosugi, M. Shimizu, and H. Ishida, “High-speed 1-frame/ms scanning confocal microscope with a microlens and nipkow disks,” *Applied optics*, vol. 41, no. 22, pp. 4704–4708, 2002.

[46] L. Valiya Peedikakkal, A. Furley, and A. J. Cadby, “A multimodal adaptive super-resolution and confocal microscope,” *bioRxiv*, 2018. DOI: `10.1101/397273`. eprint: `https://www.biorxiv.org/content/early/2018/08/24/397273.full.pdf`. [Online]. Available: `https://www.biorxiv.org/content/early/2018/08/24/397273`.

[47] . C. Optics, *X-light v3*, `https://crestopt.com/xlightv3/`.

[48] C. B. Müller and J. Enderlein, “Image scanning microscopy,” *Phys. Rev. Lett.*, vol. 104, p. 198 101, 19 May 2010. DOI: `10.1103/PhysRevLett.104.198101`. [Online]. Available: `https://link.aps.org/doi/10.1103/PhysRevLett.104.198101`.

[49] M. G. L. Gustafsson, "Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy," *Journal of Microscopy*, vol. 198, no. 2, pp. 82–87, May 2000, ISSN: 1365-2818. DOI: 10.1046/j.1365-2818.2000.00710.x. [Online]. Available: https://doi.org/10.1046/j.1365-2818.2000.00710.x.

[50] C. Z. AG, *Introducing lattice sim for zeiss elyra 7 structured illumination microscopy with a 3d lattice for live cell imagingintroducing lattic*, https://applications.zeiss.com/C125792900358A3F/0/31094A41795C8E17C125835A005391AF/$FILE/EN_41_013_188-WP_Introducing_Lattice_SIM_for_Elyra-7.pdf.

[51] A. G. York, P. Chandris, D. D. Nogare, *et al.*, "Instant super-resolution imaging in live cells and embryos via analog image processing," *Nature methods*, vol. 10, no. 11, pp. 1122–1126, 2013.

[52] A. G. York, S. H. Parekh, D. D. Nogare, *et al.*, "Resolution doubling in live, multicellular organisms via multifocal structured illumination microscopy," *Nature methods*, vol. 9, no. 7, pp. 749–754, 2012.

[53] T. Azuma and T. Kei, "Super-resolution spinning-disk confocal microscopy using optical photon reassignment," *Opt. Express*, vol. 23, no. 11, pp. 15 003–15 011, Jun. 2015. DOI: 10.1364/OE.23.015003. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-23-11-15003.

[54] C. Roider, M. Ritsch-Marte, and A. Jesacher, "High-resolution confocal raman microscopy using pixel reassignment," *Optics Letters*, vol. 41, no. 16, pp. 3825–3828, 2016.

[55] R. Heintzmann, T. M Jovin, and C. Cremer, "Saturated patterned excitation microscopy-a concept for optical resolution improvement. j. opt. soc. am. a. opt. image sci. vis. 19, 1599-1609," *Journal of the Optical Society of America. A, Optics, image science, and vision*, vol. 19, pp. 1599–609, Sep. 2002. DOI: 10.1364/JOSAA.19.001599.

[56] M. G. L. Gustafsson, "Nonlinear structured-illumination microscopy: Wide-field fluorescence imaging with theoretically unlimited resolution," *Proceedings of the National Academy of Sciences*, vol. 102, no. 37, pp. 13 081–13 086, 2005, ISSN: 0027-8424. DOI: 10.1073/pnas.0406877102. eprint: https://www.pnas.org/content/102/37/13081.full.pdf. [Online]. Available: https://www.pnas.org/content/102/37/13081.

[57] G. Tortarolo, M. Castello, S. Koho, and G. Vicidomini, "Synergic combination of stimulated emission depletion microscopy with image scanning microscopy to reduce light dosage," *bioRxiv*, p. 741 389, 2019.

[58] M. A. Lauterbach, "Finding, defining and breaking the diffraction barrier in microscopy – a historical perspective," *Optical Nanoscopy*, vol. 1, no. 1, p. 8, Nov. 2012, ISSN: 2192-2853. DOI: 10.1186/2192-2853-1-8. [Online]. Available: https://doi.org/10.1186/2192-2853-1-8.

[59] K. C. Gwosch, J. K. Pape, F. Balzarotti, *et al.*, "Minflux nanoscopy delivers 3d multicolor nanometer resolution in cells," *Nature methods*, vol. 17, no. 2, pp. 217–224, 2020.

[60] U. Endesfelder and M. Heilemann, "Direct stochastic optical reconstruction microscopy (dstorm)," in *Advanced Fluorescence Microscopy: Methods and Protocols*, P. J. Verveer, Ed. New York, NY: Springer New York, 2015, pp. 263–276, ISBN: 978-1-4939-2080-8. DOI: 10.1007/978-1-4939-2080-8_14. [Online]. Available: https://doi.org/10.1007/978-1-4939-2080-8_14.

[61] D. J. Nieves, K. Gaus, and M. A. B. Baker, "Dna-based super-resolution microscopy: Dna-paint," *Genes*, vol. 9, no. 12, 2018, ISSN: 2073-4425. DOI: 10.3390/genes9120621. [Online]. Available: http://www.mdpi.com/2073-4425/9/12/621.

[62] F. Balzarotti, Y. Eilers, K. C. Gwosch, *et al.*, "Nanometer resolution imaging and tracking of fluorescent molecules with minimal photon fluxes," *Science*, vol. 355, no. 6325, pp. 606–612, 2017.

[63] S. W. Hell and J. Wichmann, "Breaking the diffraction resolution limit by stimulated emission: Stimulated-emission-depletion fluorescence microscopy," *Opt. Lett.*, vol. 19, no. 11, pp. 780–782, Jun. 1994. DOI: 10.1364/OL.19.000780. [Online]. Available: http://ol.osa.org/abstract.cfm?URI=ol-19-11-780.

[64] S. W. Hell and M. Kroug, "Ground-state-depletion fluorscence microscopy: A concept for breaking the diffraction resolution limit," *Applied Physics B*, vol. 60, no. 5, pp. 495–497, May 1995, ISSN: 1432-0649. DOI: 10.1007/BF01081333. [Online]. Available: https://doi.org/10.1007/BF01081333.

[65] M. Hofmann, C. Eggeling, S. Jakobs, and S. W Hell, "Breaking the diffraction barrier in fluorescence microscopy at low light intensities by using reversibly photoswitchable proteins," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, pp. 17 565–9, Jan. 2006. DOI: 10.1073/pnas.0506010102.

[66] G. Carlo, "Über entstehung wahrer lichtabsorption und scheinbare koppelung von quantensprüngen," *Zeitschrift für Physik*, vol. 10, no. 1, pp. 185–199, 1922.

[67] T. Förster, "Zwischenmolekulare energiewanderung und fluoreszenz," *Annalen der physik*, vol. 437, no. 1-2, pp. 55–75, 1948.

[68] A. Munoz-Losa, C. Curutchet, B. P. Krueger, L. R. Hartsell, and B. Mennucci, "Fretting about fret: Failure of the ideal dipole approximation," *Biophysical journal*, vol. 96, no. 12, pp. 4779–4788, 2009.

[69] B. W. VanDerMeer, "Kappaphobia is the elephant in the fret room," *Methods and Applications in Fluorescence*, vol. 8, no. 3, p. 030 401, 2020.

[70] D. B. VanBeek, M. C. Zwier, J. M. Shorb, and B. P. Krueger, "Fretting about fret: Correlation between kappa and r," *Biophysical journal*, vol. 92, no. 12, pp. 4168–4178, 2007.

[71] B. Van der Meer, “Kappa-squared: From nuisance to new sense,” *Reviews in Molecular Biotechnology*, vol. 82, no. 3, pp. 181–196, 2002.

[72] T. Eilert, E. Kallis, J. Nagy, C. Röcker, and J. Michaelis, “Complete kinetic theory of fret,” *The Journal of Physical Chemistry B*, vol. 122, no. 49, pp. 11 677–11 694, 2018.

[73] N. K. Lee, A. N. Kapanidis, Y. Wang, *et al.*, “Accurate fret measurements within single diffusing biomolecules using alternating-laser excitation,” *Biophysical journal*, vol. 88, no. 4, pp. 2939–2953, 2005.

[74] T. Ha, “Single-molecule fret,” in *Single Molecules*, Wiley Online Library, vol. 2, 2001, pp. 283–284.

[75] A. Ingargiola, E. Lerner, S. Chung, S. Weiss, and X. Michalet, “Fretbursts: An open source toolkit for analysis of freely-diffusing single-molecule fret,” *PloS one*, vol. 11, no. 8, e0160716, 2016.

[76] J. Hohlbein, T. D. Craggs, and T. Cordes, “Alternating-laser excitation: Single-molecule fret and beyond,” *Chemical Society Reviews*, vol. 43, no. 4, pp. 1156–1171, 2014.

[77] Z. Liu, R. Wang, X. Tian, *et al.*, “Dynamic, inter-subunit interactions between the n-terminal and central mutation regions of cardiac ryanodine receptor,” *Journal of Cell Science*, vol. 123, no. 10, pp. 1775–1784, 2010.

[78] D. C. Lamb, “Single-pair fret: An overview with recent applications and future perspectives,” *Single Particle Tracking and Single Molecule Energy Transfer*, pp. 97–129, 2009.

[79] R. Roy, S. Hohng, and T. Ha, “A practical guide to single-molecule fret,” *Nature methods*, vol. 5, no. 6, pp. 507–516, 2008.

[80] C. Bruschini, H. Homulle, I. M. Antolovic, S. Burri, and E. Charbon, “Single-photon avalanche diode imagers in biophotonics: Review and outlook,” *Light: Science & Applications*, vol. 8, no. 1, pp. 1–28, 2019.

[81] M. Oheim, A. Salomon, A. Weissman, M. Brunstein, and U. Becherer, “Calibrating evanescent-wave penetration depths for biological tirf microscopy,” *Biophysical journal*, vol. 117, no. 5, pp. 795–809, 2019.

[82] R. Yasuda, T. Masaike, K. Adachi, H. Noji, H. Itoh, and K. Kinosita Jr, “The atp-waiting conformation of rotating f1-atpase revealed by single-pair fluorescence resonance energy transfer,” *Proceedings of the National Academy of Sciences*, vol. 100, no. 16, pp. 9314–9318, 2003.

[83] R. Lamichhane, A. Solem, W. Black, and D. Rueda, “Single-molecule fret of protein–nucleic acid and protein–protein complexes: Surface passivation and immobilization,” *Methods*, vol. 52, no. 2, pp. 192–200, 2010.

[84] B. Ambrose, J. M. Baxter, J. Cully, *et al.*, “The smfbox is an open-source platform for single-molecule fret,” *Nature communications*, vol. 11, no. 1, pp. 1–6, 2020.

[85] H. Wilson and Q. Wang, "Abel-fret: Tether-free single-molecule fret with hydrodynamic profiling," *Nature methods*, vol. 18, no. 7, pp. 816–820, 2021.

[86] B. Hellenkamp, S. Schmid, O. Doroshenko, *et al.*, "Precision and accuracy of single-molecule fret measurements—a multi-laboratory benchmark study," *Nature methods*, vol. 15, no. 9, pp. 669–676, 2018.

[87] D. Magde, E. Elson, and W. W. Webb, "Thermodynamic fluctuations in a reacting system—measurement by fluorescence correlation spectroscopy," *Physical review letters*, vol. 29, no. 11, p. 705, 1972.

[88] E. L. Elson, "Fluorescence correlation spectroscopy: Past, present, future," *Biophysical journal*, vol. 101, no. 12, pp. 2855–2870, 2011.

[89] K. Bacia, S. A. Kim, and P. Schwille, "Fluorescence cross-correlation spectroscopy in living cells," *Nature methods*, vol. 3, no. 2, pp. 83–89, 2006.

[90] A. Rossetta, E. Slenders, M. Donato, *et al.*, "The brighteyes-ttm: An open-source time-tagging module for single-photon microscopy," *bioRxiv*, 2021. DOI: 10.1101/2021.10.11.463950. eprint: https://www.biorxiv.org/content/early/2021/10/12/2021.10.11.463950.full.pdf. [Online]. Available: https://www.biorxiv.org/content/early/2021/10/12/2021.10.11.463950.

[91] K. Suhling, L. M. Hirvonen, J. A. Levitt, *et al.*, "Fluorescence lifetime imaging (flim): Basic concepts and some recent developments," *Medical Photonics*, vol. 27, pp. 3–40, 2015.

[92] A. Battisti, S. Panettieri, G. Abbandonato, *et al.*, "Imaging intracellular viscosity by a new molecular rotor suitable for phasor analysis of fluorescence lifetime," *Analytical and bioanalytical chemistry*, vol. 405, no. 19, pp. 6223–6233, 2013.

[93] M. A. Bennet, P. R. Richardson, J. Arlt, A. McCarthy, G. S. Buller, and A. C. Jones, "Optically trapped microsensors for microfluidic temperature measurement by fluorescence lifetime imaging microscopy," *Lab on a Chip*, vol. 11, no. 22, pp. 3821–3828, 2011.

[94] M. J. Behne, J. W. Meyer, K. M. Hanson, *et al.*, "Nhe1 regulates the stratum corneum permeability barrier homeostasis: Microenvironment acidification assessed with fluorescence lifetime imaging," *Journal of biological chemistry*, vol. 277, no. 49, pp. 47 399–47 406, 2002.

[95] B. Herman, P. Wodnicki, S. Kwon, *et al.*, "Recent developments in monitoring calcium and protein interactions in cells using fluorescence lifetime microscopy," *Journal of Fluorescence*, vol. 7, no. 1, pp. 85–91, 1997.

[96] H. Szmacinski and J. R. Lakowicz, "Potassium and sodium measurements at clinical concentrations using phase-modulation fluorometry," *Sensors and Actuators B: Chemical*, vol. 60, no. 1, pp. 8–18, 1999.

[97] H. Szmacinski and J. R. Lakowicz, "Fluorescence lifetime characterization of magnesium probes: Improvement of mg2+ dynamic range and sensitivity using phase-modulation fluorometry," *Journal of fluorescence*, vol. 6, no. 2, pp. 83–95, 1996.

[98] J. Aluko, C. Perrin, V. Devauges, *et al.*, "Semi-autonomous real-time programmable fluorescence lifetime segmentation with a digital micromirror device," *Optics Express*, vol. 26, no. 24, pp. 31 055–31 074, 2018.

[99] P. French, N. Andrews, M. Dallman, *et al.*, "Visualising apoptosis in live zebrafish using fluorescence lifetime imaging with optical projection tomography to map fret biosensor activity in space and time," *Journal of Biophotonics*, vol. 9, no. 4, pp. 414–424, Apr. 2016.

[100] C. Gong and T. Hogan, "Cmos compatible fabrication processes for the digital micromirror device," *IEEE Journal of the Electron Devices Society*, vol. 2, no. 3, pp. 27–32, May 2014, ISSN: 2168-6734. DOI: 10.1109/JEDS.2014.2309129.

[101] J. P. Rice, J. Neira, M. Kehoe, and R. Swanson, "Dmd diffraction measurements to support design of projectors for test and evaluation of multispectral and hyperspectral imaging sensors," *Proc SPIE*, vol. 7210, Feb. 2009. DOI: 10.1117/12.808990.

[102] M. Thomas, "Improved optical arrangement for digital micromirror device," GB2520554B, 2013.

[103] B.-J. Chang, L.-J. Chou, Y.-C. Chang, and S.-Y. Chiang, "Isotropic image in structured illumination microscopy patterned with a spatial light modulator," *Opt. Express*, vol. 17, no. 17, pp. 14 710–14 721, Aug. 2009. DOI: 10.1364/OE.17.014710. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-17-17-14710.

[104] D. Dan, M. Lei, B. Yao, *et al.*, "Dmd-based led-illumination super-resolution and optical sectioning microscopy," *Scientific reports*, vol. 3, no. 1, pp. 1–7, 2013.

[105] P. Verveer, Q. Hanley, P. Verbeek, L. van Vliet, and T. Jovin, "Theory of confocal fluorescence imaging in the programmable array microscope (pam)," *Journal of Microscopy*, vol. 189, no. 3, pp. 192–198, 1998, ISSN: 0022-2720.

[106] Verveer, Hanley, Verbeek, V. Vliet, and Jovin, "Theory of confocal fluorescence imaging in the programmable array microscope (pam)," *Journal of Microscopy*, vol. 189, no. 3, pp. 192–198, 1998. DOI: 10.1046/j.1365-2818.1998.00336.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1046/j.1365-2818.1998.00336.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1046/j.1365-2818.1998.00336.x.

[107] R. Heintzmann, Q. Hanley, D. Arndt-Jovin, and T. Jovin, "A dual path programmable array microscope (pam): Simultaneous acquisition of conjugate and non-conjugate images," *Journal of microscopy*, vol. 204, no. 2, pp. 119–135, 2001.

[108] P. A. De Beule, A. H. de Vries, D. J. Arndt-Jovin, and T. M. Jovin, "Generation-3 programmable array microscope (pam) with digital micro-mirror device (dmd)," in *Emerging Digital Micromirror Device Based Systems and Applications III*, SPIE, vol. 7932, 2011, pp. 155–164.

[109] Q. S. Hanley, P. J. Verveer, and T. M. Jovin, "Spectral imaging in a programmable array microscope by hadamard transform fluorescence spectroscopy," *Applied spectroscopy*, vol. 53, no. 1, pp. 1–10, 1999.

[110] P. G. Pitrone, J. Schindelin, L. Stuyvenberg, *et al.*, "Openspim: An open-access light-sheet microscopy platform," *Nature methods*, vol. 10, no. 7, pp. 598–599, 2013.

[111] V. Maioli, F. Görlitz, S. Warren, *et al.*, "Three-dimensional fluorescence imaging by stage-scanning oblique plane microscopy (conference presentation)," in *Three-Dimensional and Multidimensional Microscopy: Image Acquisition and Processing XXIII*, SPIE, vol. 9713, 2016, pp. 215–215.

[112] M. H. Jenkins and T. K. Gaylord, "Three-dimensional quantitative phase imaging via tomographic deconvolution phase microscopy," *Applied optics*, vol. 54, no. 31, pp. 9213–9227, 2015.

[113] D. M. Ritchie, S. C. Johnson, M. Lesk, B. Kernighan, *et al.*, "The c programming language," *Bell Sys. Tech. J*, vol. 57, no. 6, pp. 1991–2019, 1978.

[114] B. Stroustrup, *The C++ programming language*. Pearson Education, 2013.

[115] B. Stroustrup, H. Sutter, and G. Dos Reis, *A brief introduction to c++s model for type-and resource-safety*, 2015.

[116] *Rust*. [Online]. Available: `https://www.rust-lang.org/` (visited on 11/30/2022).

[117] B. Qin, Y. Chen, Z. Yu, L. Song, and Y. Zhang, "Understanding memory and thread safety practices and issues in real-world rust programs," in *Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2020, pp. 763–779.

[118] J. Bloch, *Effective java*. Addison-Wesley Professional, 2008.

[119] B. Porter, J. v. Zyl, and O. Lamy, *Welcome to apache maven*. [Online]. Available: `https://maven.apache.org/` (visited on 11/30/2022).

[120] *Gradle build tool*. [Online]. Available: `https://gradle.org/` (visited on 11/30/2022).

[121] C. MacNeill and S. Bodewig, *Apache ant*. [Online]. Available: `https://ant.apache.org/` (visited on 11/30/2022).

[122] *Project panama: Interconnecting jvm and native code*. [Online]. Available: `https://openjdk.org/projects/panama/` (visited on 11/30/2022).

[123] *Kotlin programming language*. [Online]. Available: `https://kotlinlang.org/` (visited on 11/30/2022).

[124] A. Krall, "Efficient javavm just-in-time compilation," in *Proceedings. 1998 International Conference on Parallel Architectures and Compilation Techniques (Cat. No. 98EX192)*, IEEE, 1998, pp. 205–212.

[125] L. Gherardi, D. Brugali, and D. Comotti, "A java vs. c++ performance evaluation: A 3d modeling benchmark," in *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, Springer, 2012, pp. 161–172.

[126] *Matlab.* [Online]. Available: `https://uk.mathworks.com/products/matlab.html` (visited on 11/30/2022).

[127] *Gnu octave.* [Online]. Available: `https://octave.org/` (visited on 11/30/2022).

[128] M. Abramoff, P. Magalhaes, and S. Ram, "Image processing with imagej," *Biophotonics International*, vol. 11, no. 7, pp. 36–42, 2004.

[129] C. Matthews and F. P. Cordelieres, "Metroloj: An imagej plugin to help monitor microscopes health," in *ImageJ User & Developer Conference proceedings*, 2010, pp. 1–6.

[130] H. Kirshner, D. Sage, and M. Unser, "3d psf models for fluorescence microscopy in imagej," in *Proceedings of the Twelfth International Conference on Methods and Applications of Fluorescence Spectroscopy, Imaging and Probes (MAF'11)*, Strasbourg French Republic, vol. 154, 2011.

[131] M. Ovesny, P. Křižek, J. Borkovec, Z. Švindrych, and G. M. Hagen, "Thunderstorm: A comprehensive imagej plug-in for palm and storm data analysis and super-resolution imaging," *Bioinformatics*, vol. 30, no. 16, pp. 2389–2390, 2014.

[132] R. F. Laine, K. L. Tosheva, N. Gustafsson, *et al.*, "Nanoj: A high-performance open-source super-resolution microscopy toolbox," *Journal of physics D: Applied physics*, vol. 52, no. 16, p. 163 001, 2019.

[133] E. M. Steele and D. S. Steele, "Automated detection and analysis of ca2+ sparks in xy image stacks using a thresholding algorithm implemented within the open-source image analysis platform imagej," *Biophysical journal*, vol. 106, no. 3, pp. 566–576, 2014.

[134] M. Müller, V. Mönkemöller, S. Hennig, W. Hübner, and T. Huser, "Open-source image reconstruction of super-resolution structured illumination microscopy data in imagej," *Nature communications*, vol. 7, no. 1, pp. 1–6, 2016.

[135] A. Edelstein, M. Tsuchida, N. Amodaj, H. Pinkard, R. Vale, and N. Stuurman, "Advanced methods of microscope control using µmanager software," *Journal of Biological Methods*, vol. 1, no. 2, 2014.

[136] *Python.* [Online]. Available: `https://www.python.org/` (visited on 12/07/2022).

[137] C. R. Harris, K. J. Millman, S. J. Van Der Walt, *et al.*, "Array programming with numpy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[138] E. Jones, T. Oliphant, P. Peterson, *et al.*, *SciPy: Open source scientific tools for Python*, 2001–. [Online]. Available: `http://www.scipy.org/`.

[139] A. Kutter, *The schiefspiegler (oblique telescope).* Sky Pub., 1958.

[140] R. Ranjan, M. Ferrara, A. Filograna, C. Valente, and L. Sirleto, "Femtosecond stimulated raman microscopy: Home-built realization and a case study of biological imaging," *Journal of Instrumentation*, vol. 14, no. 09, P09008, 2019.

[141] N. A. Hartell, "Simple windows-based software for the control of laser scanning confocal microscopes," *Journal of neuroscience methods*, vol. 162, no. 1-2, pp. 26–31, 2007.

[142] E. J. Gualda, T. Vale, P. Almada, J. A. Feijó, G. G. Martins, and N. Moreno, "Openspinmicroscopy: An open-source integrated microscopy platform," *Nature methods*, vol. 10, no. 7, pp. 599–600, 2013.

[143] A. Guver, N. Fifita, P. Milas, *et al.*, "A low-cost and high-precision scanning electrochemical microscope built with open source tools," *HardwareX*, vol. 6, e00082, 2019.

[144] M. Rubini and C. Rajasekaran, "Design of high performance system-on-chips using field programmable gate arrays (fpga)," in *2014 International Conference on Communication and Signal Processing*, IEEE, 2014, pp. 358–362.

[145] S. Cass, "Painless fpga programming: The alchitry au kit can simplify projects that need a lot of input/output," *IEEE Spectrum*, vol. 57, no. 12, pp. 18–20, 2020.

[146] *Zeromq guide.* [Online]. Available: `https://zguide.zeromq.org/` (visited on 12/05/2022).

[147] H. Pinkard, N. Stuurman, I. E. Ivanov, *et al.*, "Pycro-manager: Open-source software for customized and reproducible microscope control," *Nature methods*, vol. 18, no. 3, pp. 226–228, 2021.

[148] S. Cass, "Nvidia makes it easy to embed ai: The jetson nano packs a lot of machine-learning power into diy projects," *IEEE Spectrum*, vol. 57, no. 7, pp. 14–16, 2020.

[149] C. J. Sheppard, S. B. Mehta, and R. Heintzmann, "Superresolution by image scanning microscopy using pixel reassignment," *Optics letters*, vol. 38, no. 15, pp. 2889–2892, 2013.

[150] Y.-C. Huang and J.-W. Pan, "High contrast ratio and compact-sized prism for dlp projection system," *Opt. Express*, vol. 22, no. 14, pp. 17016–17029, Jul. 2014. DOI: `10.1364/OE.22.017016`. [Online]. Available: `https://opg.optica.org/oe/abstract.cfm?URI=oe-22-14-17016`.

[151] M. J Rust, M. Bates, and X. Zhuang, "Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (storm)," *Nature methods*, vol. 3, pp. 793–5, Nov. 2006. DOI: `10.1038/nmeth929`.

[152] B. Scholkopf, K.-K. Sung, C. J. Burges, *et al.*, "Comparing support vector machines with gaussian kernels to radial basis function classifiers," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2758–2765, 1997.

[153] A. W. Fitzgibbon, "Simultaneous linear estimation of multiple view geometry and lens distortion," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, IEEE, vol. 1, 2001, pp. I–I.

[154] *Epsrc policy framework on research data.* [Online]. Available: `https://www.ukri.org/about-us/epsrc/our-policies-and-standards/policy-framework-on-research-data/expectations/` (visited on 12/06/2022).

[155] M. Folk, G. Heber, Q. Koziol, E. Pourmal, and D. Robinson, "An overview of the hdf5 technology suite and its applications," in *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*, 2011, pp. 36–47.

[156] *Pcie-6353 and usb-6353 specifications.* [Online]. Available: https://www.ni.com/docs/en-US/bundle/pcie-usb-6353-specs/page/specs.html (visited on 12/07/2022).

[157] *Lighthub compact laser combiners.* [Online]. Available: https://www.omicron-laser.de/english/light-engines/lighthub-laser-combiner/lighthub-laser-combiner.html (visited on 12/07/2022).

[158] *Excelitas photon detection catalog.* [Online]. Available: https://www.excelitas.com/file-download/download/public/59646?filename=PD_Excelitas%20Photon%20Detection%20Catalog.pdf (visited on 12/07/2022).

[159] A. Ingargiola, T. Laurence, R. Boutelle, S. Weiss, and X. Michalet, "Photon-hdf5: An open file format for timestamp-based single-molecule fluorescence experiments," *Biophysical journal*, vol. 110, no. 1, pp. 26–33, 2016.

[160] E. Bisong, "Matplotlib and seaborn," in *Building machine learning and deep learning models on google cloud platform*, Springer, 2019, pp. 151–165.

[161] TIOBE, *Tiobe index 2022.* [Online]. Available: https://www.tiobe.com/tiobe-index/ (visited on 12/07/2022).

[162] *Qt framework.* [Online]. Available: https://www.qt.io/ (visited on 12/07/2022).

[163] S. C. Bera, T. Paul, A. S. Iyengar, and P. P. Mishra, "Direct observation of the external force mediated conformational dynamics of an ihf bound holliday junction," *Faraday Discussions*, vol. 207, pp. 251–265, 2018.

[164] J. Ross, P. Buschkamp, D. Fetting, A. Donnermeyer, C. M. Roth, and P. Tinnefeld, "Multicolor single-molecule spectroscopy with alternating laser excitation for the investigation of interactions and dynamics," *The Journal of Physical Chemistry B*, vol. 111, no. 2, pp. 321–326, 2007.

[165] M. Y. Berezin and S. Achilefu, "Fluorescence lifetime measurements and biological imaging," *Chemical reviews*, vol. 110, no. 5, pp. 2641–2684, 2010.

[166] R. Machado, J. Cabral, and F. S. Alves, "Recent developments and challenges in fpga-based time-to-digital converters," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 11, pp. 4205–4221, 2019. DOI: 10.1109/TIM.2019.2938436.

[167] A. N. Kapanidis, N. K. Lee, T. A. Laurence, S. Doose, E. Margeat, and S. Weiss, "Fluorescence-aided molecule sorting: Analysis of structure and interactions by alternating-laser excitation of single molecules," *Proceedings of the National Academy of Sciences*, vol. 101, no. 24, pp. 8936–8941, 2004.

[168] H. Johnson and M. Graham, *High-speed Digital Design: A Handbook of Black Magic* (Prentice Hall Modern Semiconductor Design). Prentice Hall, 1993, ISBN: 9780133957242. [Online]. Available: `https://books.google.co.uk/books?id=H5SsQgAACAAJ`.

[169] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith, "Cython: The best of both worlds," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 31–39, 2010.

APPENDIX

A

FPGA AND NETWORK INTERFACE COMMANDS

Command	Description
W <counter><register><value>	Writes a value to a given configuration register **Parameters:** counter - The counter to configure (integer between 0 and 11) register - The register to write to value - The value to write (ASCII encoded base 10 integer) **Available values for register parameter:** P - The period register S - The start/on time register E - The end/off time register C - The configuration register
R <counter><register>	Read the contents of a configuration register (see W for description of parameters)
BGN	Begin acquisition, i.e., enable outputs
END	End acquisition, i.e., disable outputs
PAU	Pause counters
VER	Return the version of the trigger interface in <major>.<minor>.<patch>format

Table A.1.: Serial commands for the FPGA trigger interface

Command	Description
CONNECT	Requests connection information from the server. The server will respond with the IP address of its PUB socket
UPDATE	Requests an update on the current state of the DMD (i.e., the uploaded patterns and the pattern that is currently displayed, if any)
RUN <index:Integer> <exposure:Integer>	Requests the server run the pattern at the given index. The exposure time is provided to allow the CairnFocal Control pattern to update any exposure related settings when using the FPGA Trigger Interface An index of -1 with no exposure time is interpreted as a stop request
PATTERNCONFIG <index:Integer>	Requests the configuration (i.e., the number of camera frames required per iteration of the pattern and the whether the camera is being triggered) of the pattern with the given index
OTTERCONFIG	Requests the names of the enabled outputs on the FPGA trigger interface

Table A.2.: Valid client commands for the CairnFocal Control program

Update Descriptor	Update Information	Description
U	See Appendix B.1	Triggered in response to an UPDATE command and when new patterns are uploaded to the DMD via the CairnFocal control software Contains information about the currently running pattern and the current patterns uploaded to the DMD
R	See Appendix B.2	Triggered when a pattern is started or stopped Contains the currently running pattern index (if any), as well as the number of camera frames required per image and the illumination order set up in the FPGA trigger interface
PC	See Appendix B.3	Triggered in response to a PATTERNCONFIG command from a client Contains extended information about the currently running pattern
OC	See Appendix B.4	Triggered by an OTTERCONFIG command or when the user changes the cofiguration of the FPGA trigger interface in the CairnFocal Control software
H	N/A	Periodic heartbeat message sent if another update message hasn't been sent within the last second

Table A.3.: The update messages produced by the CairnFocal Control server

APPENDIX

B

JSON SCHEMA FOR THE CAIRNFOCAL CONTROL NETWORK INTERFACE

B.1. State Schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema",
  "type": "object",
  "properties": {
    "currentlyRunning": {
      "type": "number"
    },
    "patterns": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "name": {
            "type": "string"
          },
          "nFrames": {
            "type": "number"
          }
        },
        "required": [
          "name",
          "nFrames"
        ]
      }
    }
  },
  "required": [
    "currentlyRunning",
    "patterns"
  ],
  "definitions": {
  }
}
```

B.2. Run Info

```
{
  "$schema": "http://json-schema.org/draft-07/schema",
  "type": "object",
  "properties": {
    "index": {
      "type": "number"
    },
    "lightOrder": {
      "properties": {
        "type": {
          "type": "string",
          "enum": [
            "unknown",
            "order"
          ]
        }
      },
      "anyOf": [
        {
          "type": "object"
        },
        {
          "type": "object",
          "properties": {
            "type": {
              "const": "order"
            },
            "lightOrder": {
              "type": "array",
              "items": {
                "type": "object",
                "properties": {
                  "index": {
                    "type": "number"
                  },
                  "n": {
                    "type": "number"
                  }
                },
                "required": [
                  "index",
                  "n"
                ]
              }
            },
            "scanned": {
              "type": "array",
              "items": {
                "type": "number"
              }
            },
            "nChannels": {
              "type": "number"
            }
          },
          "required": [
            "lightOrder"
          ]
        }
      ],
      "required": [
        "type"
      ]
    },
    "framesPerImage": {
      "type": "number"
    }
  },
  "required": [
    "index",
    "lightOrder",
    "framesPerImage"
  ],
  "definitions": {
  }
}
```

B.3. Pattern Config

```
{
  "$schema": "http://json-schema.org/draft-07/schema",
  "type": "object",
  "properties": {
    "index": {
      "type": "number"
    },
    "framesPerImage": {
      "type": "number"
    },
    "isOtterControlled": {
      "type": "boolean"
    }
  },
  "required": [
    "index",
    "framesPerImage",
    "isOtterControlled"
  ],
  "definitions": {
  }
}
```

B.4. Otter Setup

```
{
  "$schema": "http://json-schema.org/draft-07/schema",
  "properties": {
    "type": {
      "type": "string",
      "enum": [
        "none",
        "setup"
      ]
    }
  },
  "anyOf": [
    {
      "type": "object"
    },
    {
      "type": "object",
      "properties": {
        "type": {
          "const": "setup"
        },
        "outputs": {
          "type": "array",
          "items": {
            "type": "object",
            "properties": {
              "index": {
                "type": "number"
              },
              "label": {
                "type": "string"
              }
            },
            "required": [
              "index",
              "label"
            ]
          }
        }
      },
      "required": [
        "outputs"
      ]
    }
  ],
  "required": [
    "type"
  ],
  "definitions": {
  }
}
```

APPENDIX
C

EXAMPLE TRIGGER INTERFACE PROGRAM

```
// Import the Otter SDK
import uk.co.otter.*

fun main() {
    try {
        // List available serial ports (note: these might
        // not be valid FPGA Trigger Interfaces)
        val availablePorts = Otter.getPorts()

        // Exit program if there are no serial ports
        if (availablePorts.isEmpty()) {
            error("No FPGA Trigger Interfaces Connected")
        }

        // Attempt to connect to the first serial port
        val triggerInterface = Otter(availablePorts.first())

        // All 3 counters will use the same configuration
        val counterConfiguration = Value.Config(
            inp_src = CountSrc.Exposure1,// Count Exposure1
            clk_div = 1,                 // No clock
                                         //    division
            and_src = AndSrc.Exposure1,  // Effective Global
```

```
                                         //    Shutter using
                                         //    Exposure1
            continuous = false           // Increment counter
                                         //    on rising edge
        )

        // Create a list of command objects to be sent to the device
        val commands = listOf(
            Command.Write(0, Parameter.Period, Value.Int(6)),
            Command.Write(0, Parameter.Start, Value.Int(0)),
            Command.Write(0, Parameter.End, Value.Int(2)),
            Command.Write(0, Parameter.Config, counterConfiguration),

            Command.Write(1, Parameter.Period, Value.Int(6)),
            Command.Write(1, Parameter.Start, Value.Int(3)),
            Command.Write(1, Parameter.End, Value.Int(4)),
            Command.Write(1, Parameter.Config, counterConfiguration),

            Command.Write(2, Parameter.Period, Value.Int(6)),
            Command.Write(2, Parameter.Start, Value.Int(5)),
            Command.Write(2, Parameter.End, Value.Int(5)),
            Command.Write(2, Parameter.Config, counterConfiguration),
        )

        // Send the commands to the device
        for (command in commands) {
            triggerInterface.send(command)
        }

        // Start the acquisition, wait for 5 seconds
        // and then stop
        triggerInterface.send(Command.Start)
        Thread.sleep(5000)
        triggerInterface.send(Command.Stop)
    } catch (ex : Exception) {
        println(ex.message)
    }
}
```

Listing C.1: Example Kotlin program that demonstrates uploading a basic configuration to the FPGA Trigger Interface. The first 3 counters are configured to count the rising edges of the first input. For the first 3 camera exposures, the interface will output on the first channel, the next 2 on the second channel and the final image will correspond to an output on the 3rd channel. All triggers are gated using the first exposure.

APPENDIX D

MATLAB/OCTAVE SCRIPT TO GENERATE FIGURE 3.5

```
data = [0.722689617086106 0.525187147920930 0.663449506295816 0.557589098451781 0.508602124124338 0.687742594284194 0.588309114926233 0.796364027939603 0.556490556621363 0.551917188427631 0.665273587371560;...
0.565928305354024 0.669765166309847 0.757078386345100 0.697999915160579 0.670742685006313 0.705545655003519 0.646566790582412 0.561015018991770 0.617034116568503 0.694699328553482 0.671458197297660;...
0.681674459787080 0.629167355544549 0.659092646413700 0.661280303328813 0.639508074150300 0.580001251592032 0.591609035002517 0.613675069252871 0.705059202574050 0.675115739943391 0.618631139813655;...
0.684642824247538 0.631024377064321 0.674060800851197 0.692923810987957 0.692442552532806 0.663808875377328 0.629206050891740 0.653027917014504 0.666568296459503 0.667300849930014 0.685980636971413;...
0.695827648009808 0.695052727670529 0.679656714702113 0.678759053050423 0.654230336503714 0.672814894418316 0.626199878310331 0.584909213324261 0.614463162510847 0.674758569965546 0.677200826002660;...
0.665757834680879 0.679087908202466 0.685895100498940 0.690252050821221 0.702809054134719 0.625417309423711 0.627917618
700047 0.658548674580577 0.700318806597679 0.716380324842310 0.620366238234908;...
0.710333663169546 0.722747431518023 0.681244954916164 0.647310557668285 0.687099932165395 0.684747717699201 0.678417456259162 0.690349710879337 0.647022245231759 0.643341564314695 0.630003607222656;...
0.673647114032665 0.696052963907687 0.709398563575506 0.658618030464078 0.691835091354024 0.666641000792488 0.673852275439617 0.685428620851190 0.691154031290165 0.666787747433177 0.632666924854347]
[sp, sz] = meshgrid(5:15, 3:10);

data_reshaped = reshape(data, [88 1]);
sp_reshaped = reshape(sp, [88, 1]);
sz_reshaped = reshape(sz, [88, 1]);
o = ones(88, 1);

X = [o sz_reshaped sp_reshaped];
Y = data_reshaped;

b = ((X' * X)^-1) * X' * Y;

fit = reshape(X * b, [8, 11]);

figure;
surf(sp, sz, data, "FaceAlpha",0.3);
hold on;
surf(sp, sz, fit);
xlabel("Pinhole Spacing (pixels)");
ylabel("Pinhole Size (pixels)");
zlabel("Average Axial FWHM (um)");
title("Effect of Varying Pinhole Size and Spacing on Axial FWHM");

[~,idx] = min(data(:));
[r,c] = ind2sub(size(data), idx);
```

```
spacing = sp(r, c);
pinhole_size = sz(r, c);
```

Listing D.1: Script used to generate Figure 3.5

APPENDIX

E

CAIRNFOCAL RESOLUTION STATISTICAL SIGNIFICANCE TESTING

This notebook performs the statistical tests to determine if the resolutions achieved on the CairnFocal in its different modalities were significantly different from one another.

Load Libraries

[28]:

```
options(warn=0)
library(tidyverse)
library(colorspace)
library(rcartocolor)
library(ggforce)
library(ggdist)
library(ggridges)
library(ggbeeswarm)
library(gghalves)
library(systemfonts)
library(car)
library(report)
library(FSA)
```

Loading Data

The data are loaded from the CSV. The data consists of 3 columns, Technique, BeadSize and FWHM. Technique is a string representing the technique used (either Widefield, Confocal, ISM or ISM Deconvolved). BeadSize is either 200 or 100 indicating the size of the bead the measurement was taken with. FWHM is the measured FWHM of the bead (measured in microns). The BeadSize column is converted to a factor to allow it to be plotted as a category rather than numeric data. A 4th column, TechniqueBeadSize, is added to the dataframe. This column is simply the concatination of the Technique and BeadSize columns and is used to produce all the categories used in the statistical testing.

[29]:

```
data <- read_csv("fwhm.csv", show_col_types = FALSE)
data$BeadSize <- factor(as.factor(data$BeadSize), level = c(200, 100))
data$TechniqueBeadSize <- factor(paste(data$Technique, data$BeadSize))
print(data)
```

```
# A tibble: 176 × 4
   Technique BeadSize  FWHM TechniqueBeadSize
   <chr>     <fct>
<dbl> <fct>
 1 Widefield 200      0.315 Widefield 200
 2 Widefield 200      0.311 Widefield 200
 3 Widefield 200      0.307 Widefield 200
 4 Widefield 200      0.318 Widefield 200
 5 Widefield 200      0.320 Widefield 200
 6 Widefield 200      0.300 Widefield 200
 7 Widefield 200      0.307 Widefield 200
 8 Widefield 200      0.304 Widefield 200
 9 Widefield 200      0.306 Widefield 200
10 Widefield 200      0.308 Widefield 200
# i 166 more rows
```

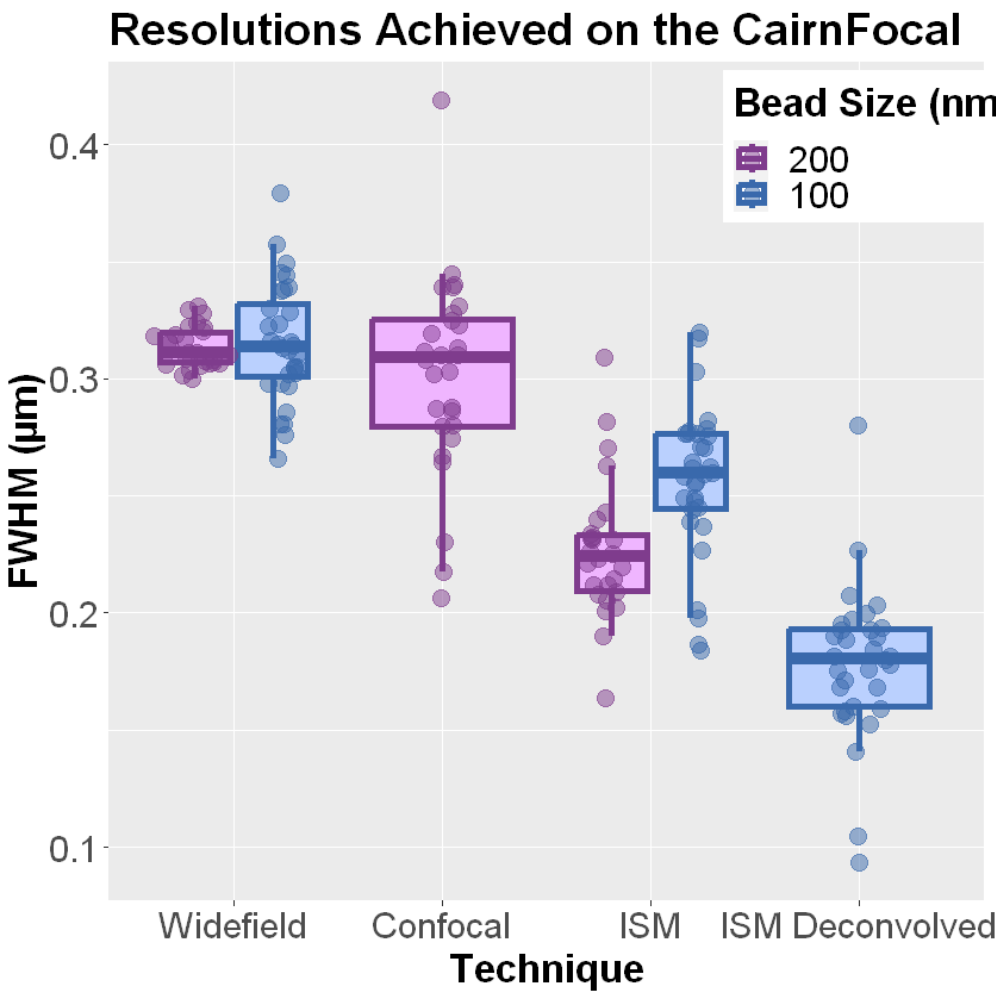
Plotting Data

The data are plotted as both scatter plots and box plots to generate Figure 3.11

[31]:

```
my_pal <- rcartocolor::carto_pal(n = 8, name = "Bold")[c(1, 3, 7, 2)]

g <- ggplot(
    data,
    aes(
      x = factor(Technique, level = c('Widefield', 'Confocal', 'ISM',␣
 ↪'ISM Deconvolved')), BeadSize,
      y = FWHM,
      color = BeadSize,
      fill = BeadSize,
    )
  ) +
  scale_color_manual(values = my_pal, guide = "none") +
  scale_fill_manual(values = my_pal, guide = "none") +
  ylab("FWHM (\u00b5m)") +
  xlab("Technique") +
  ggtitle("Resolutions Achieved on the CairnFocal") +
  guides(color = guide_legend(title = "Bead Size (nm)")) +
  theme(
    axis.text = element_text(size = 18),
    axis.title = element_text(size = 20, face = "bold"),
    legend.title = element_text(size = 20, face = "bold"),
    legend.text = element_text(size = 18),
    legend.position = c(0.875, 0.9),
    plot.title = element_text(size = 22, face = "bold")
  ) +
  geom_boxplot(
    aes(fill = BeadSize, fill = after_scale(colorspace::lighten(fill, .
 ↪7))),
    size = 1.5, outlier.shape = NA
  ) +
  ggforce::geom_sina(size = 4, alpha = .5)

print(g)
```

```
Warning message:
"Duplicated aesthetics after name standardisation: fill"
```



Significance Testing

We are interested in knowing if the means of the various datasets differ significantly and since there are multiple categories to compare an ANOVA should be the first test attempted. In an ANOVA test, H0 is that the means of all the various distributions are equal. H1 is therefore that at least one of the means is different. Throughout analysis, a p-value of 0.05 is used as the significance threshold.

Before applying the ANOVA test, the underlying assumptions of the test must be checked. The ANOVA has 4 main assumptions: Independance of Measurements, Equality of Variances, Normality, and that there are no significant outliers. Independance of

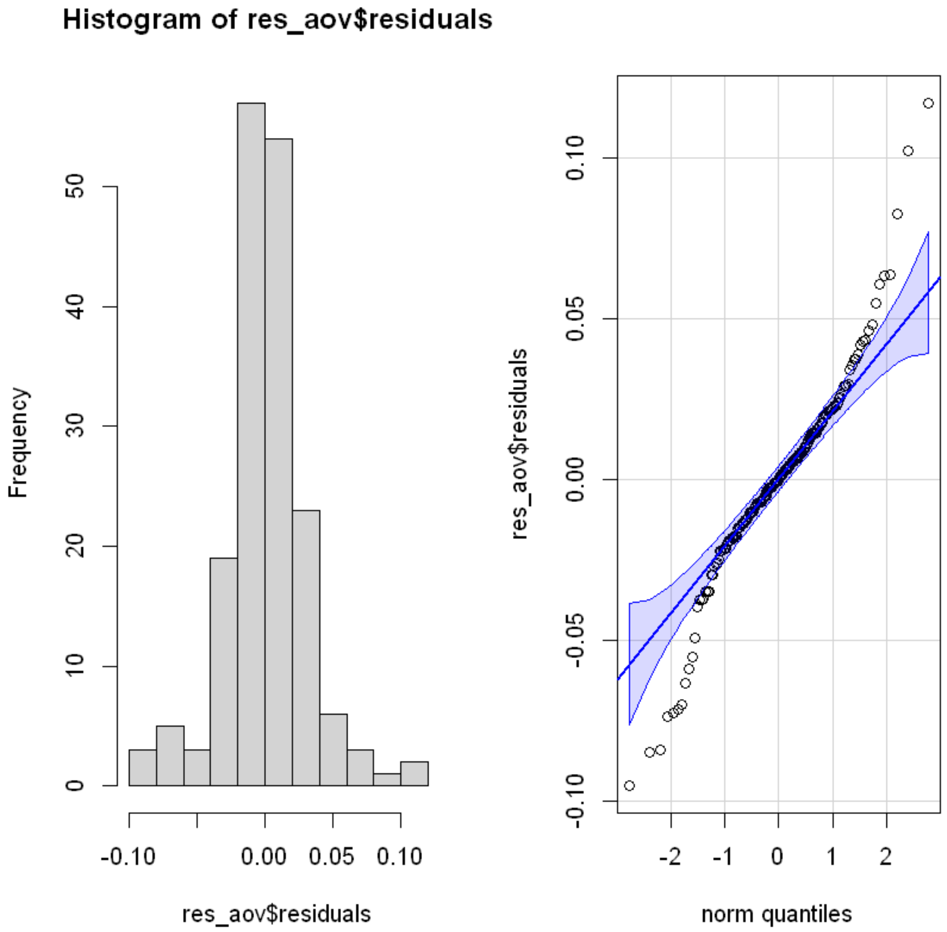
Measuremnets is implied by the data collection method, each FWHM is derrived from an indipendant measurement of a bead's size. From the above graph it's clear that the equality of variances criterion does not hold, 200nm Widefield clearly has a lower variance than 200nm Confocal data. If an ANOVA is to be performed it will therefore have to be the Welch variant, which can handle differing variances. The outliers criterion also appears not to be met, with most of the categories containing several datapoints that fall outside the IQR. Whether the normality criterion is met is less obvious so this shall be investigated both visually and by performing a Shapiro-Wilk test on the ANOVA residuals. If the normality or outlier criterion are not met then a Kruskal-Wallis test must be applied instead.

In the Shapiro-Wilk test, H0 is that all the data come from a normal distribution and H1 is that the data do not come from a normal distribution. Once again, a p-value of greater than 0.05 shall be used as the threshold to indicate normality.

[33]:

```
res_aov <- aov(FWHM ~ TechniqueBeadSize, data = data)
par(mfrow = c(1, 2))
hist(res_aov$residuals)

library(car)
qqPlot(res_aov$residuals, id = FALSE)

shapiro.test(res_aov$residuals)
```

```

        Shapiro-Wilk normality test

data:  res_aov$residuals
W = 0.94797, p-value = 4.694e-06
```



From the left plot, the data do not appear to be normally distributed, with small secondary peaks towards the left and right of the histogram. If the data were normally distributed, the datapoints would all appear close to the straight blue line in the right plot. Again towards the left and right of the plot the data differs from this expectation, indicating a lack of normality. Finally, the Shapiro-Wilk test resulted in p = 4.694e-6 which is much less than 0.05. H0 must therefore be rejected and we conclude that the data are not normally distributed. Since the normality criterion is not met, we must instead apply the Kruskal-Wallis test.

Kruskal-Wallis Test

The Kruskal-Wallis test is similar to an ANOVA, however, relaxes the normality, equality of variances and outlier requirements, assuming only independance of measurements. When performing this test, the hypotheses will be as follows:

H0 : The different techniques achieved the same resolutions as measured by the FWHM

H1 : At least one technique achieved a different resolution

As before, a p-value of less than 0.05 will indicate significance.

[34]:

```r
kruskal.test(FWHM ~ TechniqueBeadSize, data = data)
```

```
        Kruskal-Wallis rank sum test

data:  FWHM by TechniqueBeadSize
Kruskal-Wallis chi-squared = 126.95, df = 5, p-value < 2.2e-16
```

The test resulted in $p = 2.2e\text{-}16 < 0.05$. We therefore reject the null hypothesis and conclude that at least one technique achieved a different resolution to the others. To determine which are significant a pairwise Dunn Test shall be performed.

Post-Hoc Testing

The Dunn test will compare each of the categories pairwise to determine which are significantly different without suffering from the issues normally associated with multiple testing. When comparing two techniques, T1 and T2, the hypotheses are as follows:

H0 : The resolution achieved with T1 and T2 are the same

H1 : The resolution achieved with T1 and T2 are not the same

As before, $p < 0.05$ shall be considered significant

[120]:

```r
library(FSA)
res_dunn <- dunnTest(
  FWHM ~ TechniqueBeadSize,
  data = data,
  method = "holm"
)


cmp <- res_dunn$res$Comparison
```

```r
splt <- strsplit(cmp, split = " - ")

sig_to_star <- function(p) {
    if (p > 0.05) {
        return(" ")
    } else if (p > 0.01) {
        return("*")
    } else if (p > 0.001) {
        return("**")
    } else {
        return("***")
    }
}

final_results <- data.frame(
    FirstTechnique = sapply(splt, function(v) { v[1] }),
    SecondTechnique = sapply(splt, function(v) { v[2] }),
    P = res_dunn$res$P.adj,
    Significance = sapply(res_dunn$res$P.adj, sig_to_star)
)

print(final_results)
```

```
        FirstTechnique     SecondTechnique            P Significance
1         Confocal 200             ISM 100 3.428315e-03           **
2         Confocal 200             ISM 200 1.654226e-05          ***
3              ISM 100             ISM 200 4.690174e-01             
4         Confocal 200 ISM Deconvolved 100 7.706848e-13          ***
5              ISM 100 ISM Deconvolved 100 1.807660e-04          ***
6              ISM 200 ISM Deconvolved 100 7.813623e-02             
7         Confocal 200       Widefield 100 9.325946e-01             
8              ISM 100       Widefield 100 3.542430e-05          ***
9              ISM 200       Widefield 100 3.506279e-08          ***
10 ISM Deconvolved 100       Widefield 100 1.683350e-17          ***
11        Confocal 200       Widefield 200 6.361720e-01             
12             ISM 100       Widefield 200 8.417707e-05          ***
```

```
13             ISM 200       Widefield 200 1.433004e-07          ***
14 ISM Deconvolved 100       Widefield 200 6.802215e-16          ***
15       Widefield 100       Widefield 200 9.707787e-01
```

Conclusions

The post-hoc testing indicates that the majority of the techniques achieved different resolutions. Neither of the two techniques that were tested on both the 100 nm and 200 nm beads (widefield and ISM) achieved different resolutions on the different bead sizes, as is expected since both bead sizes were below the expected resolution limit of the technqiues. The slight improvement in resolution over widefield associated with confocal operation was determined not to be statistically signficant. Since theory predicts a small improvement in resolution, it is likely that the insignificant result is due to the difficulties measuring the FWHM in the confocal data that are discussed in detail in Chapter 3. One surprising result is that the difference in resolution achieved by ISM without deconvolution on the 200 nm beads when compared to ISM with deconvolution on the 100 nm beads was insignificant. A direct comparison between these two situations may not be appropriate, however (see Chapter 3's Discussion), and deconvolution did result in a significant improvement in resolution when looking at 100 nm beads.

APPENDIX

F

SMFBOX CAMERA MODULES INTERFACE DEFINITIONS

F.1. Data Types

This section describes the common types used by the smfCameraModules framework

F.1.1. ErrorCodes::ErrorCode

```
namespace ErrorCodes {
    using ErrorCode = int;
    constexpr ErrorCode no_error = 0;
    constexpr ErrorCode buffer_too_small = -1;
    constexpr ErrorCode index_out_of_range = -2;
    constexpr ErrorCode internal_error = -3;
    constexpr ErrorCode module_not_initialized = -4;
}
```

`ErrorCodes::ErrorCode` is the type returned by all functions that may fail. 5 common error types are provided as constexpr values, however, these are not exhaustive and camera modules may choose to return custom values. By convention, negative return codes indicate failure, positive indicate a warning and 0 indicates success.

F.1.2. Version

```
struct Version {
    int major;
    int minor;
    int patch;
};
```

The `Version` structure is used to describe the version of the library in order to determine compatibility between applications and dynamically loaded camera modules. A camera module is compatible with an application if the `major` version number matches and the `minor` version of the camera module is greater than or equal to the `minor` version number of the application's library. The `path` number is used to track changes that don't have an effect on the API and therefore isn't used when determining compatibility.

F.2. Module Functions

This section describes the set of functions used in interacting with the camera module dynamic library, rather than individual cameras.

F.2.1. get_interface_version

```
Version get_interface_version();
```

Returns the interface version implemented by the camera module. See F.1.2 for a description of the semantic versioning employed.

F.2.2. get_last_error

```
ErrorCodes::ErrorCode get_last_error(
    char*           buffer,
    size_t*         buffer_size);
```

Retrieves a string describing the last error encountered by the camera module. If `buffer` is a null pointer, then the required buffer size will be placed into the value pointed to by the `buffer_size` pointer. Otherwise, the error message will be written to the `buffer` string. A `buffer_too_small` error code is returned if `buffer_size` indicates that `buffer` is not large enough to store the whole string. `buffer_size` must not be a null pointer.

- buffer is a pointer to the destination string.
- buffer_size is a pointer to the length of the buffer array.

F.2.3. load_module

```
ErrorCodes::ErrorCode load_module();
```

Performs any required initialization. Needs to be called before all other functions other than get_interface_version and get_module_code. A call to unload_module is required before this function can be called again.

F.2.4. unload_module

```
void unload_module();
```

Performs any clean up required before the module can be closed. Shall only be called once per call to load_module.

F.2.5. get_module_code

```
const char* get_module_code();
```

Returns a 4 character string that identifies the camera module. The return value shall not be a null pointer.

F.2.6. get_n_available_cameras

```
ErrorCodes::ErrorCode get_n_available_cameras(size_t* size);
```

Returns the number of cameras currently available through this camera module. Subsequent calls to get_camera_id are guaranteed to return valid data, however, calls to connect_to_camera may fail if the system state has changed (e.g., a camera has been unplugged).

- size is a pointer to the number of available cameras.

F.2.7. get_camera_id

```
ErrorCodes::ErrorCode get_camera_id(
    size_t              index,
    char*               buffer,
    size_t*             buffer_size);
```

Used to get a unique identifier for the camera at the given index. This function is not guaranteed to succeed unless `get_n_available_cameras` has been called. If `buffer` is a null pointer, then the required buffer size will be placed into the value pointed to by the `buffer_size` pointer. Otherwise, the camera ID will be written to the `buffer` string. A `buffer_too_small` error code is returned if `buffer_size` indicates that `buffer` is not large enough to store the whole string. `buffer_size` must not be a null pointer.

- `index` is the index of the camera to to query for is unique identifier.
- `buffer` is a pointer to the destination string.
- `buffer_size` is a pointer to the length of the `buffer` array.

F.2.8. connect_to_camera

```
ErrorCodes::ErrorCode connect_to_camera(
    size_t              index,
    void**              camera);
```

Connects to the camera with a given index and returns a handle to an object representing the connection. Should be matched with a call to `disconnect_camera`.

- `index` is the index of the camera to connect to.
- `camera` is a pointer to the handle to the object representing the connection to the camera.

F.3. Camera Interaction Functions

The camera interaction functions are used to interface with a specific camera once a connection has been established. All camera interaction functions are marked with the ‘camera’ prefix and take a pointer to a camera connection as their first argument, which shall be a non-null handle returned by the `connect_to_camera` function.

F.3.1. disconnect_camera

```
void disconnect_camera(void* camera);
```

Terminates the connection to the camera and cleans up any resources associated with the connection. Shall not be called more than once per connection. The `camera` pointer is invalid after calling this function.

- `camera` is a pointer to the camera connection object to be destroyed.

F.3.2. camera_start_acquisition

```
ErrorCodes::ErrorCode camera_start_acquisition(void* camera);
```

Begins an acquisition. An acquisition must be running before `camera_snap` will return images. No guarantee is made as to whether this function will succeed if an acquisition is already running.

- `camera` is a pointer to the camera connection object.

F.3.3. camera_stop_acquisition

```
ErrorCodes::ErrorCode camera_stop_acquisition(void* camera);
```

Stops a currently running acquisition. No guarantee is made as to whether this function will succeed if an acquisition is not running.

- `camera` is a pointer to the camera connection object.

F.3.4. camera_snap

```
ErrorCodes::ErrorCode camera_snap(void* camera);
```

Instructs the camera to take a single image. This function may return immediately or may block until an image is available. Ideally, implementations should block in the `camera_get_image` function if possible, allowing other work to be performed as the image data is acquired.

- `camera` is a pointer to the camera connection object.

F.3.5. camera_get_image

```
ErrorCodes::ErrorCode camera_get_image(
    void*            camera,
    char*            buffer,
    size_t*          buffer_size);
```

Returns the last image captured by the camera, blocking if an image has not yet been received since the last call to `camera_snap`. If called before `camera_snap`, the function will not block, though no guarantees are made about the contents of `buffer` once the function has returned. If `buffer` is a null pointer, then the required buffer size will be placed into the value pointed to by the `buffer_size` pointer. Otherwise, the image data will be written to `buffer`. A `buffer_too_small` error code is returned if `buffer_size` indicates that `buffer` is not large enough to store the whole image. `buffer_size` must not be a null pointer.

- `camera` is a pointer to the camera connection object.
- `buffer` is a pointer to the destination image buffer.
- `buffer_size` is a pointer to the length of the `buffer`.

F.3.6. camera_get_width

```
ErrorCodes::ErrorCode camera_get_width(
    const void*      camera,
    size_t*          width);
```

Returns the width of the images expected from the camera.

- `camera` is a pointer to the camera connection object.
- `width` is a pointer to the expected width of a camera image.

F.3.7. camera_get_height

```
ErrorCodes::ErrorCode camera_get_height(
    const void*     camera,
    size_t*         height);
```

Returns the width of the images expected from the camera.

- `camera` is a pointer to the camera connection object.
- `height` is a pointer to the expected height of a camera image.

F.3.8. camera_get_min_exposure

```
ErrorCodes::ErrorCode camera_get_min_exposure(
    const void*     camera,
    double*         exposure);
```

Returns the minimum exposure time supported by the camera in milliseconds.

- `camera` is a pointer to the camera connection object.
- `exposure` is a pointer to the minimum exposure time supported by the camera in milliseconds.

F.3.9. camera_get_max_exposure

```
ErrorCodes::ErrorCode camera_get_max_exposure(
    const void*     camera,
    double*         exposure);
```

Returns the maximum exposure time supported by the camera in milliseconds.

- `camera` is a pointer to the camera connection object.
- `exposure` is a pointer to the maximum exposure time supported by the camera in milliseconds.

F.3.10. camera_get_exposure

```
ErrorCodes::ErrorCode camera_get_exposure(
    const void*     camera,
    double*         exposure);
```

Returns the current exposure setting of the camera in milliseconds.

- camera is a pointer to the camera connection object.
- exposure is a pointer to the current exposure setting of the camera in milliseconds.

F.3.11. camera_set_exposure

```
ErrorCodes::ErrorCode camera_set_exposure(
    void*           camera,
    double          exposure);
```

Sets the exposure setting of the camera in milliseconds.

- camera is a pointer to the camera connection object.
- exposure is the desired exposure setting of the camera in milliseconds.