

---

---

# **STORMSim: STORM Microscopy Simulator**

Developing a Tool for the Simulation of Realistic STORM  
Microscopy Image Data with a view to Develop Machine Learning  
Techniques to Aid in STORM Image Reconstruction

---

---

Project Report  
Connor Wilkes, ID:1854862

Birmingham University  
School of Computer Science

## Abstract

Microscopy is one of the most important research tools utilised in a vast majority of biological research. However the diffraction limit discovered by Abbe in 1873 prevents imaging of specimens below 250nm in the XY dimension and 550nm in the Z dimension (Abbe 1873). A large number of biological entities are smaller than this limit and therefore this poses a problem (Huang, Bates, and Zhuang 2009). Traditional solutions to this problem involved fixing the specimen however this means the specimen must be dead (Yuste 2005). Recent advances have created techniques known as super resolution microscopy that enable users to overcome this diffraction limit (Leung and Chou 2011). One such example of this technique is Stochastic Optical Reconstruction Microscopy (STORM). STORM uses the stochastic nature of photoswitchable fluorophores to achieve this (Rust, Bates, and Zhuang 2006). A simulator of this technique was built as a tool to visualise this technique. Additionally, the tool was used to benchmark STORM reconstruction techniques. Data created by the tool can be used in the future to train machine learning models to reconstruct the ground truth locations of molecules from a STORM imagery.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Motivation . . . . .	2
1.2	Problem Area . . . . .	2
1.3	Tools Used . . . . .	2
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Fluorescence Microscopy . . . . .	3
2.2	Point Spread Function (PSF) . . . . .	4
2.3	Super Resolution Microscopy Techniques . . . . .	7
2.4	STORM . . . . .	9
<b>3</b>	<b>Research</b>	<b>11</b>
3.1	Introduction . . . . .	11
3.2	PSF Modelling . . . . .	11
3.3	Photoswitching Modelling . . . . .	12
3.4	Reconstruction Techniques . . . . .	13
3.5	Evaluation Techniques . . . . .	13
3.6	Deep Learning techniques . . . . .	14
<b>4</b>	<b>Methods</b>	<b>18</b>
4.1	Overview . . . . .	18
4.2	STORMSim Requirements . . . . .	20
4.2.1	Functional Requirements . . . . .	20
4.2.2	Non functional Requirements . . . . .	20
4.3	Ground Truth generation . . . . .	20
4.4	Fluorophores . . . . .	20
4.5	Photoswitching Model . . . . .	21
4.6	PSF Model . . . . .	21
4.7	Frame and Frame Stack construction . . . . .	23
4.8	Results Table . . . . .	25
4.9	User Interface . . . . .	25
4.10	ThunderSTORM reconstruction . . . . .	25
4.11	Evaluation Techniques . . . . .	26
4.11.1	Jaccard Index . . . . .	26
4.11.2	Hausdorff Distance . . . . .	27
4.11.3	RSME . . . . .	28
4.12	Deep Learning . . . . .	28
4.12.1	Convolutional Neural Network . . . . .	28
4.12.2	Auto Encoder Neural Network . . . . .	29

<b>5 Results</b>	<b>30</b>
5.1 STORMSim . . . . .	30
5.2 Reconstruction Techniques . . . . .	30
5.2.1 MLE . . . . .	30
5.2.2 Least Squares . . . . .	31
5.2.3 Comparison . . . . .	31
5.3 Deep Learning . . . . .	32
<b>6 Discussion</b>	<b>39</b>
6.1 STORMSim . . . . .	39
6.2 Reconstruction . . . . .	39
6.3 Deep Learning . . . . .	40
6.4 Future Remarks . . . . .	40
<b>A Running the code</b>	<b>44</b>
<b>B Additional PSF models</b>	<b>45</b>
B.1 Gibson & Lanni PSF model . . . . .	45
B.2 Richards & Wolf PSF model . . . . .	46
B.3 Parameters . . . . .	47
<b>C Results Raw Data</b>	<b>48</b>
C.1 MLE . . . . .	48
C.2 Least Squares . . . . .	51
C.3 Averages . . . . .	54
<b>D Project Management</b>	<b>55</b>
D.1 Overview . . . . .	55
D.2 Week 1, meeting on 14/06/2018: . . . . .	55
D.3 Week 2, meeting on 22/06/2018: . . . . .	55
D.4 Week 3, meeting on 28/06/2018: . . . . .	55
D.5 Week 4, meeting on 03/07/2018: . . . . .	55
D.6 Week 5, self directed: . . . . .	55
D.7 Week 6, 18/07/2018: . . . . .	55
D.8 Week 7, 23/07/2018: . . . . .	56
D.9 Week 8, self directed: . . . . .	56
D.10 Week 9, 07/08/2018: . . . . .	56
D.11 Week 10, self directed: . . . . .	56
D.12 Week 11, self directed: . . . . .	56
D.13 Week 12 onwards: . . . . .	56

# Chapter 1

## Introduction

### 1.1 Motivation

I understood from the outset of the project that I wished to combine my previous study in Biomedical Science alongside my current study in Computer Science. This choice stemmed from my passion for both the subjects, gained as a result of studying them but also from a willingness to push myself and demonstrate the full range of my ability. It was also clear to me that I wished to combine these two areas within the scope of Machine Learning, in particular Deep Learning as I had been particularly engaged with the Deep Learning module throughout the course of the year.

Upon discussion with Dr. Styles it was decided that an area of particular interest (additionally pertaining to research within my undergraduate dissertation that focused on cellular imaging) would be Stochastic Optical Reconstruction Microscopy (STORM) Imaging technique described in the 2006, landmark paper published of the same name (Rust, Bates, and Zhuang 2006). In particular, the project encompassed developing a simulator of STORM imaging data that could be used as a tool to educate people on the technique whilst also being able to be used as a tool to train and develop machine learning models.

### 1.2 Problem Area

The problem area presented is twofold: firstly, it is to develop an accurate simulator of STORM microscopy data that models as closely as possible the physical and chemical processes that make up the technique itself; and secondly, it is to develop neural network architectures that are trained on this simulated data that can locate the positions of the ground truth molecules such that these neural networks may be used for real STORM image data in the future.

### 1.3 Tools Used

The system was built as a Java plug in on top of a widely used scientific image processing platform called ImageJ. This was achieved by utilising the SciJava API developed alongside ImageJ that enables users to write plugins and macros to their own ends on top of the ImageJ platform. Maven was used to set up a working environment and as a build automation tool. ThunderSTORM was used to perform traditional STORM image reconstruction techniques on the simulated images. The results of these were then evaluated using evaluation methods written in Python. Finally the neural network architectures were built with Keras using a Tensorflow backend. All code is available for viewing in the git repository: <https://git-teaching.cs.bham.ac.uk/mod-msc-proj-2017/clw762>.

# Chapter 2

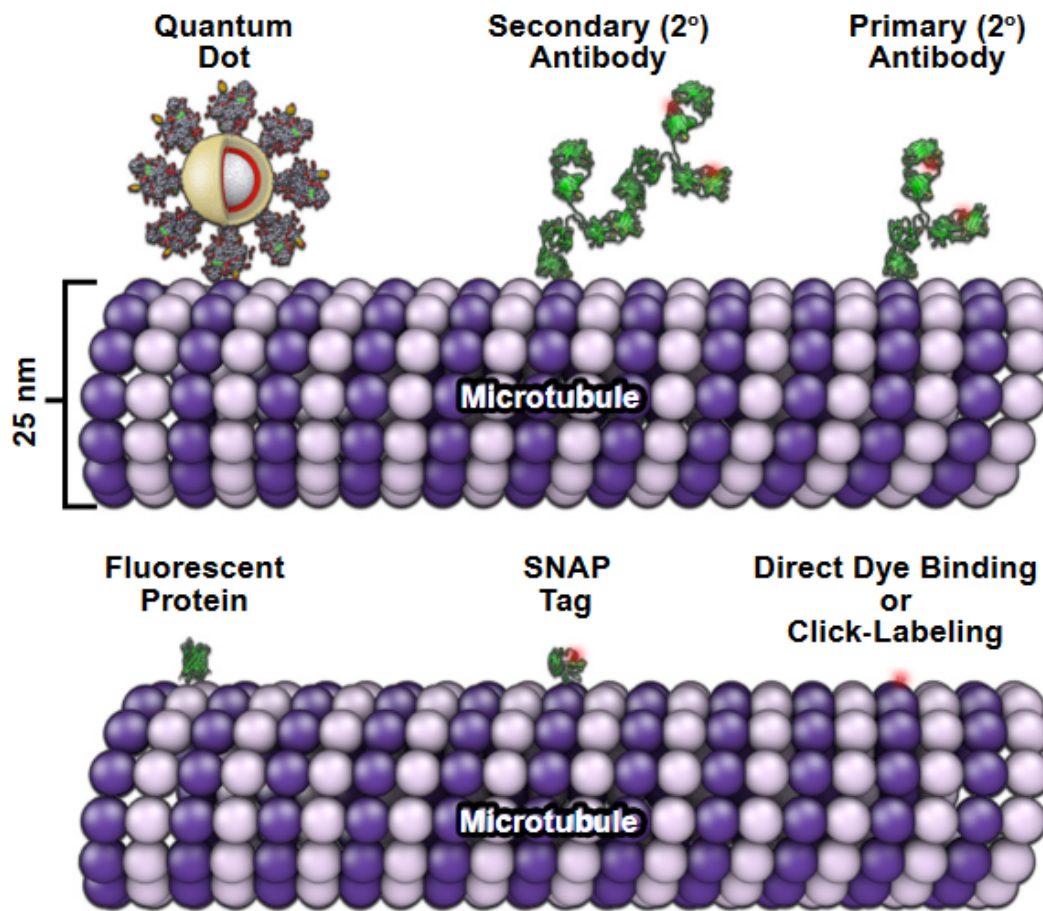
## Background

### 2.1 Fluorescence Microscopy

Microscopy has been for a long time a fundamental pillar of the way that we as humans conduct research and diagnose pathology; this is likely primarily driven by the very nature of our brains being wired towards visual sensory input. However the problem that we faced for a long time in regards to microscopy of tissue was that it does not image well down a microscope due to the inherent, photoresistant properties in many living animals and plants; likely due to the damage that light can cause to cellular processes (Yuste 2005). Histology for a long time was deemed the common fix for this problem but it meant that the imaging subject had to be fixed and stained and as a result, had to be dead (Lichtman and Fraser 2001). Not only did this prevent live microscopy, it was also a slow process to reconstruct full picture of a specimen from a number of different slices. The development of fluorescence imaging overcame this particular problem. It has enabled specimens to be viewed down the microscope live rather than having to be stained and fixed beforehand. Thereby allowing us to observe the specimen in action, which in addition being much more interesting has also allowed for previously unnoticed, dynamic processes to be ascertained giving us a much more clear understanding of the way that tissue functions (Yuste 2005).

The 'staining' process of fluorescence microscopy involves the tagging of particular molecules in the specimen with some kind of fluorophore. A fluorophore is a small chemical protein that when excited due to a change in the chemical makeup of the medium it is in or by wavelength excitation emits a burst of light (Chapman, Oparka, and Roberts 2005). There are a large number of these fluorophore molecules that serve different purposes and the choice of each different fluorophore is often specific to a particular experiment, indeed several different types may be used within one particular imaging process in order to distinguish between different molecules in the specimen; they may differ in their size, the colour that they give off upon excitation, their switching cycle (the number of times they may be switched on and off before photobleaching), excitation/emission properties and their method for excitation. The fluorophore is attached to some kind of labelling protein that gives the fluorophore specificity for a particular molecule (Leung and Chou 2011). These labelling proteins come in a number of sizes and types and a selection are demonstrated in figure 2.1.

The staining process is then followed by the 'imaging' process which involves excitation of the specimen such that the fluorophores present become excited. The emitted bursts of light from the fluorophores in the specimen can then be visualised down a microscope and marked out distinctly from any background noise and as a result the location of the fluorophores can be determined. The fluorophores within the specimen will continue to emit bursts of light up until they become photobleached whereupon they are photodamaged and may no longer emit light upon excitation (Yuste 2005). The primary problem however that restricts the accuracy and detail with regards to the imaging of fluorophores in fluorescence microscopy is the diffraction

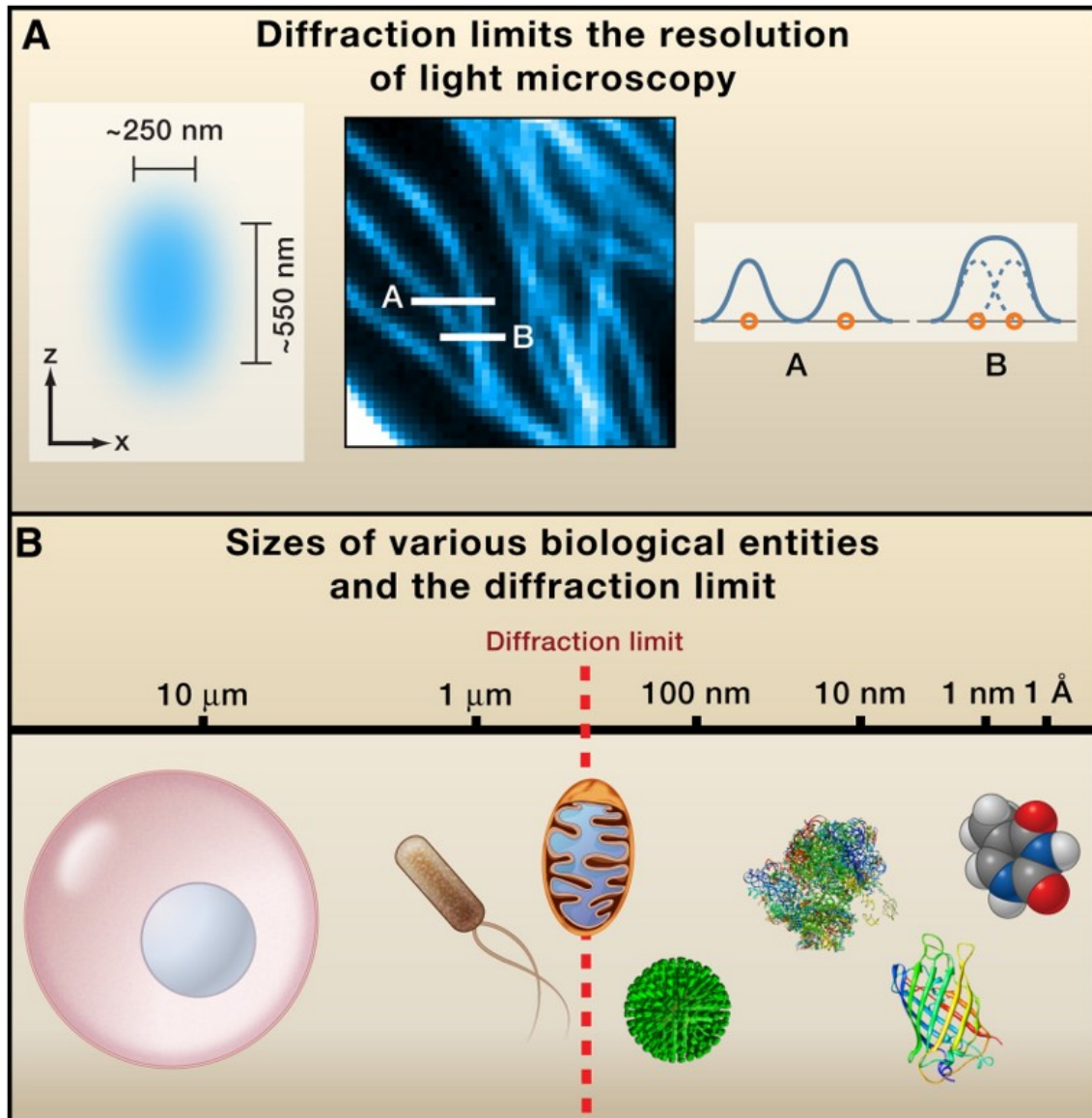


**Figure 2.1:** A figure showing a number of different methods that may be used for the tagging process. Size is shown in scale with 25nm scale bar. The labelling proteins are shown attached to a microtubule structure. As demonstrated there are a large variety of different labelling proteins, some are quite large (Quantum dot) whilst some are much smaller (SNAP tag), indeed it is possible in certain circumstances to tag directly on the area of interest with Click-Labeling. The different kinds of labelling, much like the choice of fluorophore, are chosen on an experimental basis. Note that these techniques also apply to super resolution microscopy methods. Figure taken from MicroscopyU 2017

limit. This diffraction limit with size 200-300nm in the lateral direction and 500-700nm in the axial dimension means that structures of size smaller than this cannot be effectively imaged using conventional fluorescence microscopy (see figure 2.2 for a pictorial demonstration of these limits) (Huang, Bates, and Zhuang 2009).

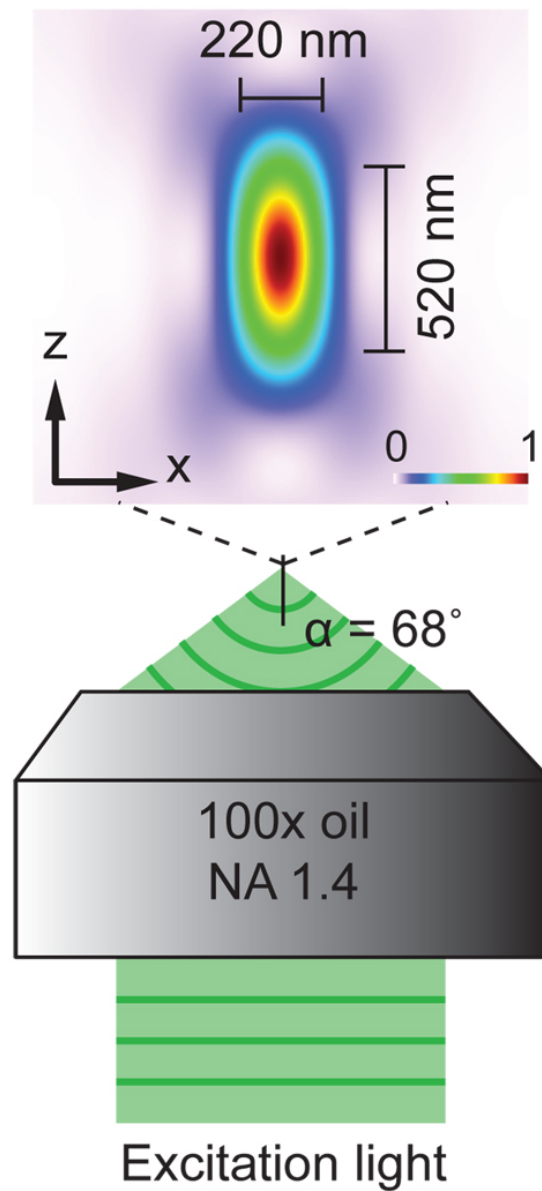
## 2.2 Point Spread Function (PSF)

The PSF of a point object describes the response that an imaging system has to the point object with respect to the diffraction of light, i.e what is actually visualised in the imaging system, typically this is a fixed-size, blur like object, and is represented as a three-dimensional (3D) intensity distribution (Huang, Bates, and Zhuang 2009). Ernest Abbe in 1873 stated that "diffraction prevents resolution of two objects if their distance apart is less than half of the full width half maximum (FWHM) generated by their point spread function" and defined the diffraction limit  $D = \lambda / 2NA$  where  $\lambda$  is the wavelength of light and NA is the numerical aperture (Abbe 1873). This suggests that any two point objects that are closer than the FWHM of the PSF cannot be easily resolved from one another (Huang, Bates, and Zhuang 2009; MacDonald, Baldini, and Storrie 2015).



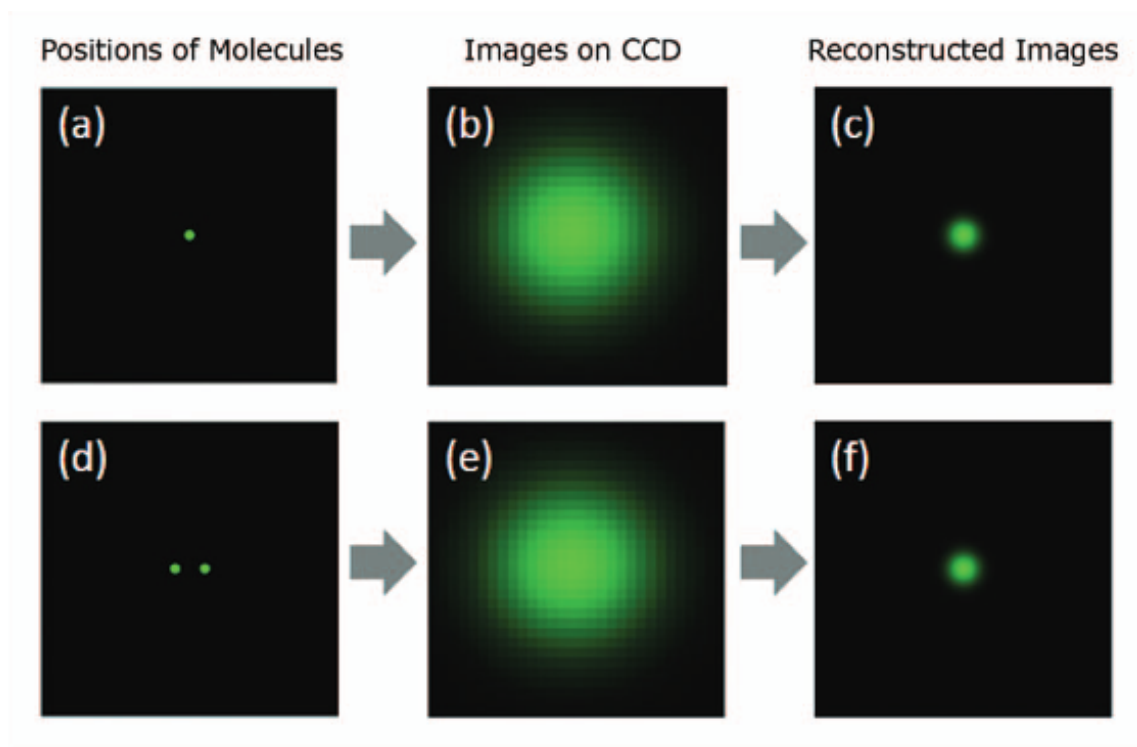
**Figure 2.2: A figure demonstrating the effect the diffraction limit has on imaging of molecular structures.** (A) A typical PSF is demonstrated with XY width of 250nm and Z width of 550nm. When objects in the imaging field are closer together than the width of the PSF they are indistinguishable from one another. This is shown by the central image present here of microtubules stained with a blue fluorophore and the cross sections next to it. Cross section A corresponds to the cross section underneath the line A and similarly for B. In both A and B's cross sections we can see that there are two molecules present in each. However the image does not show this due to the PSF interfering with our perception of the ground truth. This is further demonstrated by the cross sections A and B showing the locations of the molecules in orange and the resultant PSF in blue. (B) This figure demonstrates how the diffraction limit hampers our ability to view molecular structures. From left to right: a mammalian cell, a bacterial cell, a mitochondrion, an influenza virus, a ribosome, the green fluorescent protein and thymine (a small molecule component of DNA) Figure taken from Huang, Babcock, and Zhuang 2010.





**Figure 2.3: A figure demonstrating the PSF of a point object.** The objective lens has a magnification of 100x with an NA of 1.4x. The PSF is demonstrated in the above area and has been calculated numerically. The FWHM is shown as 220nm for the lateral dimension and 520nm in the axial dimension. This has been calculated using  $\Delta xy \approx 0.61\lambda/NA$  where  $\lambda$  is the wavelength of light, in this case 550nm. Figure taken from Huang, Bates, and Zhuang 2009.

Traditional fluorescence microscopy techniques such as wide field microscopy make use of objective lenses alongside refractive index-matched mounting media, much like the microscope demonstrated in figure 2.3, and these can achieve resolution close to the diffraction limit. As previously discussed this does not allow imaging below that; the average molecule is 2nm in size in the lateral dimension and as a result, a vast number of structures exist that are far smaller than the diffraction limit (MacDonald, Baldini, and Storrie 2015). Figure 2.4 clearly demonstrates this problem. It is clear therefore that techniques needed to be developed to tackle this problem.



**Figure 2.4:** A figure demonstrating how the diffraction limit affects the localisation of molecules from their ground truth positions. CCD here stands for charge-coupled device and is a way to image a single molecule. In (a) there is one molecule present in (b) there are two however when they are seen with their PSF in (b) and (e) they look identical. (b) and (e) were fit with a Gaussian distribution to locate their centres and the results are shown in (c) and (f). As can be seen the fitting returned the presence of one molecule in both (b) and (e). This error is caused by the PSF of the two molecules in (d) overlapping in (e). Figure taken from Leung and Chou 2011.

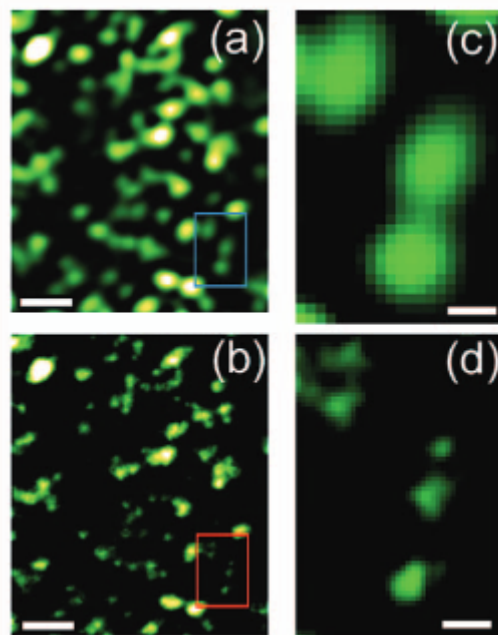
## 2.3 Super Resolution Microscopy Techniques

The techniques that seek to solve this problem and achieve sub diffraction limit microscopy are called Super Resolution Microscopy (SRM) techniques. They enable the user to distinguish much more clearly between point objects present close together in an image, effectively overcoming the constraints placed by the diffraction limit (Leung and Chou 2011). The importance of these techniques and their groundbreaking effect that these techniques have had and will continue to have on research is highlighted by the fact that the 2014 Nobel Prize in Chemistry was awarded for the discovery of these techniques; specifically "for the development of super-resolved fluorescence microscopy" (Sciences 2014). There are a large number of different techniques that fall under the scope of SRM, however they generally fall into three categories: structured illumination microscopy (SIM), spatially patterned excitation techniques and single-molecule localisation based imaging techniques (Huang, Bates, and Zhuang 2009; MacDonald, Baldini, and Storrie 2015).

SIM microscopy utilises Moire fringes to calculate light distribution. Slices of the specimens in the lateral, axial or both directions are imaged by a user under a patterned illumination field

at different wavelengths and three grids are collected: the first describes the light structure, the second describes the grid pattern by which the light is structured and the third is the Moire fringe pattern created by overlapping the first two grids (Saxena, Eluru, and Gorthi 2015). The moire fringe grid patterns contain increased information about the structure being imaged (Leung and Chou 2011). These moire fringes that are created are then used by a software program that generates the overall structure from these slices (MacDonald, Baldini, and Storrie 2015). The problem with this method is that it is also restricted by the diffraction limit of light in the form of the illumination field (Huang, Bates, and Zhuang 2009). The resultant image has a resolution limit two times better than wide field microscopy (Leung and Chou 2011).

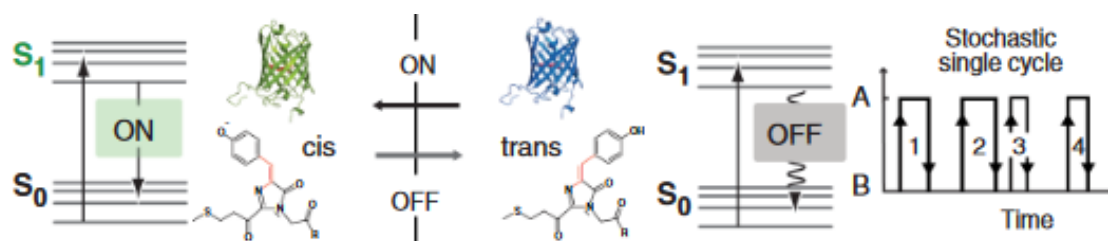
Spatially patterned excitation techniques involve depleting the generated PSF in order to extract sub-diffraction limit features (Huang, Bates, and Zhuang 2009). An example of this method is stimulated emission depletion (STED) microscopy. In STED an excitation beam is first sent to the specimen. This excites the fluorophores in an area (see fig 2.5a and fig 2.5c). A secondary, doughnut shaped depletion beam with a higher wavelength than the original excitation beam is then shone in the same location. This instantly deexcites fluorophores in that area to a lower, off state (see fig 2.5d and fig 2.5e) (Leung and Chou 2011). The centralised area is now more focused and of a higher resolution. The resolution may be increased by increasing the intensity of the depletion beam (Willig et al. 2007). Two other techniques that also employ spatially patterned excitation techniques are Reversible Saturable Optical Linear Fluorescence Transitions (RESOLFT) and saturated structured illumination microscopy (SSIM).



**Figure 2.5: A figure showing the effect the depletion beam has in STED microscopy** These are GFP-tagged caveolin 1 imaged by a STED microscope. Scale bars are 1  $\mu\text{m}$  in (a) and (b) and 200nm in (c) and (d). (a) shows a larger image of excited fluorophores without depletion, (c) shows a zoomed area of this. (b) shows a larger image of excited fluorophores after depletion, (d) shows a zoomed area of this. Figure taken from Leung and Chou 2011.

The final class of SRM techniques are single-molecule localisation based methods. These techniques involve localisation of the individual molecules that make up the imaged structure (Leung and Chou 2011). This is done using the principle of reversible fluorophores; that is fluorophores that can be switched on and off over a given time period when given some form of stimulation or destimulation event (see fig2.6) (Hell 2007). The two main kinds of localisation based methods are Photoactivated Localisation Microscopy (PALM) and STORM. Both use this previously described concept of photoswitching of reversible fluorophores and both have a lateral resolution of approximately 20nm in the lateral plane and 20nm in the axial plane(Huang,

Babcock, and Zhuang 2010). However the primary difference between the two methods is their use of fluorophores and the resultant way by which photoswitching is used; in STORM the photoswitching is stochastic, i.e one fluorophore may transfer from the emitting, excited state to the non-emitting, ground state more than once over the period of the imaging, whilst in PALM subsets of fluorophores are continuously lit over a period of time until they become photobleached, hence they may no longer emit again. The focus here will be on STORM.



**Figure 2.6:** A figure demonstrating the excitation events of a photoswitchable fluorophore. When the fluorophore is excited as demonstrated in the left the fluorophore switches to a cis state and emits light. Upon returning to the ground state demonstrated next, the fluorophore switches to a trans state and stops emitting light. The graph at the end demonstrates that as time goes on these fluorophores may switch back and forth between the two states interchangeably. Figure taken from Hell 2007.

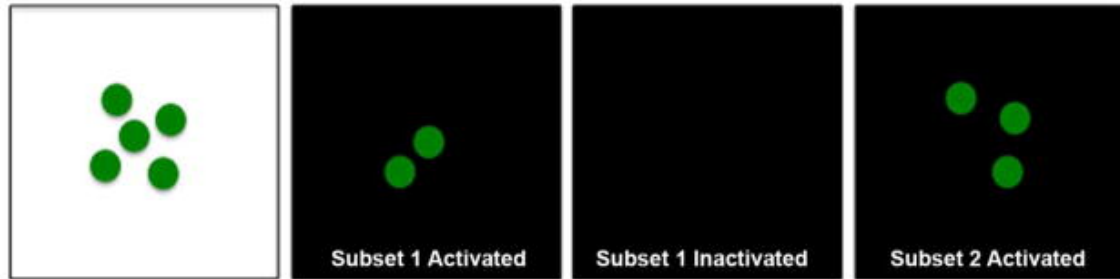
## 2.4 STORM

STORM microscopy was first outlined in the 2006 paper by Rust, Bates and Zhang and was featured alongside PALM as Methods of the Year in 2008 by the Nature Methods journal. Akin to other methods the technique requires a tagging, imaging and a reconstruction phase (Rust, Bates, and Zhuang 2006). The tagging phase is similar in principle to other SRM techniques but fundamentally requires a different kind of fluorophore. This is due to the requirements of the imaging phase wherein the stochastic nature of the technique is revealed. The reconstruction phase then takes the results of the imaging phase and attempts to, true to its namesake, reconstruct the ground truth locations of the molecules present in the specimen such that a resultant structure can be determined (Leung and Chou 2011).

Although similar in principle to other fluorescence based techniques the tagging phase of STORM microscopy is distinctly different because it requires the fluorophore to be switchable between excitation states. In order to effectively resolve the final image these photoswitchable fluorophores, named a 'switch' by Rust, Bates and Zhang must have high switching cycles and must not photobleach quickly. Commonly the switches are a combination of two fluorophores together as this enables the fluorophores to effectively switch between states (Rust, Bates, and Zhuang 2006). Rust, Bates and Zhang's original switch was Cy3-Cy5 and since then there have been a large number of switches developed with different switching cycles, photon output and labelling efficiency; the choice of which is dependant on the experiment being undertaken (Hell 2008). There are however three main concerns regarding the development of a new switch as highlighted by Huang, Babcock and Zhuang in 2010: firstly, the switch must have two definable states, an on and off state; secondly, in order for the switch to be clearly identifiable as being on and contrasted enough from background noise it must release a large amount of light in its emitting state compared to its dark state; and thirdly the excitation rate of the fluorophore must be low such that there is less chance of two close fluorophores from emitting at the same time.

The imaging phase of STORM involves switching of subsets of the fluorophores present in the image on and off. This is achieved through photoactivation. This leads to a blinking effect; at any particular time within the specimen a random, small, sparse subset of the fluorophores will be in the excited state (see fig2.7) (Huang, Babcock, and Zhuang 2010). A fluorophore may move between states several times over the course of the imaging process (Hell 2007). Note that due to the nature of the medium it is likely that as time progresses the fluorophores will move

in a radius equal to their size around their point of adhesion with their molecule. A number of frames equating to the iterative process of activation, capture, deactivation are generated from this imaging process at sequential time frames and these are then used in the reconstruction process (Huang, Babcock, and Zhuang 2010).



**Figure 2.7: A figure showing the stochastic flickering nature of STORM.** The ground truth image is demonstrated in the first image. In the second image the first subset of fluorophores in the image are active. They are then inactive in the third image. In the fourth image the second subset of fluorophores are active. As demonstrated, as time progresses different subsets of the fluorophores in the specimen emit and go dark. Note that the fluorophores may emit more than once and they may also appear in different subsets of fluorophores at every emitting state. Figure taken from Huang, Bates, and Zhuang 2009.

The reconstruction phase involves taking these individual frames and locating the individual molecules thus creating an overall ground truth image reconstruction. Molecule localisation is achieved by taking a frame and identifying all the present PSFs and reducing the effect of any background noise that may be present (Huang, Bates, and Zhuang 2009). For each PSF in the image a PSF model is fit to it and thereby molecule location is determined. As a result of the stochastic blinking of the fluorophores present and given enough time each individual fluorophore present in the image will be able to be processed individually; that is separate from the influence of any close, neighbouring fluorophores, as they will statistically at some point in the imaging process be off when the fluorophore in question is emitting. An overview of current reconstruction techniques is discussed in chapter 3.4. From a collection of processed frames an overall list of ground truth locations can thereby be determined (Huang, Babcock, and Zhuang 2010).

## Chapter 3

# Research

### 3.1 Introduction

In order to effectively model the STORM process it is important to review previous literature in relation to key concepts that require simulation. It is important for the results of the simulator to be accurate enough such that it may be used to train machine learning models for the reconstruction process in the future; additionally, there is a need to fit as closely as possible the real world situation such that anyone using this piece of software as a training or education tool may get a full understanding of how the technique functions and how changing various parameters such as the fluorophore's features could affect the potential final outcome of the technique.

Foremost the most important aspect of effectively modelling the technique is an effective model of the PSF, indeed as outlined in chapter 2 it is the diffraction limit that is the primary need for the existence of this technique. Thereby, it is clearly very important that this aspect of the simulation is as accurate as possible. The secondary area of focus is the photoswitching model, the model that dictates the way that the photoswitching fluorophores switch between states as outlined in chapter 2.4 and demonstrated in figure 2.7.

The other important aspect of STORMSim is how the data it creates may be used in the reconstruction process. It is important to see how traditional reconstruction techniques may be used on the data both to test how robust the model is at creating reliable data but also as a potential future metric to quickly evaluate future reconstruction techniques in the absence of real world STORM data. ThunderSTORM is another imageJ plug in that performs these reconstruction techniques and additionally has a limited STORM simulator within it (Ovesny et al. 2014). Indeed, research into the limitations of ThunderSTORM helped shape the design of STORMSim. In order to evaluate the success of the STORMSim platform with regards to traditional reconstruction techniques it was clear that a number of evaluation metrics needed to be considered that compared the ground truth and the reconstruction predictions. Understanding of these processes enabled research into the avenue of using deep learning techniques to aid in the reconstruction process.

### 3.2 PSF Modelling

There are a variety of different models that exist that seek to model the way that a point object reacts to and diffracts light. These various models account for a number of different factors such as whether the imaged specimen is in focus, the parameters of the particular microscope being used and the dimensions of the model. The Richards & Wolf PSF models and the Gibson & Lanni PSF models (see appendix B) both account for differences in the microscope parameters and how that would potentially affect the produced PSF (Gibson and Lanni 1989; Richards and Wolf 1959; Kirshner et al. 2013).

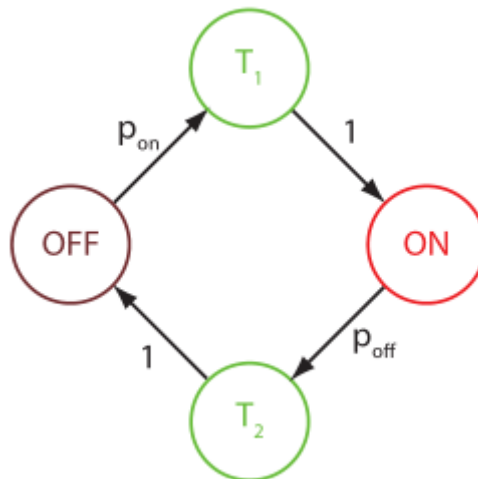
However the most common form of PSF modelling takes the form of a Gaussian model. This is highlighted by ThunderSTORM's use of Gaussian models in its simulations and within its reconstruction techniques, where the PSF is fit to some kind of Gaussian model (Ovesny et al. 2014). This can be represented both in 3D and in 2D, it is described in 2D as follows:

$$G(x, y) = e^{-\frac{((x-x_0)^2 + (y-y_0)^2)}{2\sigma^2}} \quad (3.1)$$

where  $x$  and  $y$  represent the location of the current location,  $x_0$  and  $y_0$  represent the location of the point,  $e$  is Euler's constant and  $\sigma$  is some variance that describes the width of the PSF. This formulae may be expanded to many variants such as elliptical variants (Kirshner et al. 2013; Ovesny et al. 2014). It has been proven that this model of PSF is a valid approximation of the real world PSF (Small and Stahlheber 2014; Stallinga and Rieger 2010).

### 3.3 Photoswitching Modelling

As previously discussed it is important to effectively model the way by which the photoswitchable fluorophores move between states as depicted in figure 2.6. ThunderSTORM's approach in this case is to assign a number of molecules present within some kind of area and then sample from a Poisson distribution to get the actual number that may appear and then render from this (Ovesny et al. 2014). The overt problem that lies in this approach is that it does not take into account any transitional changes between an off and an on state and it ignores the fact that a fluorophore may stay on for longer than one frame (Hell 2007; Rust, Bates, and Zhuang 2006). An alternative model is that which is described by a Markov chain model. A hidden Markov approach has been outlined by two separate papers that both seek to use it to characterise fluorophore behaviour. In both cases it has shown to be a highly robust model at predicting the behaviour both with real world data and simulated data (Patel et al. 2017; Frahm 2016). An example of this Markov model is demonstrated in figure 3.1. Additionally to this, a paper in May of this year used a Markov model to successfully model photoswitching of fluorophores in order to build a Deep learning guided Bayesian inference for the reconstruction phase of SRM and a separate simulator of STORM data TestSTORM also utilises this approach (Li et al. 2018; Sinkó et al. 2014).



**Figure 3.1: A Markov model of photoswitching.** Similarly to the photoswitching state diagram in figure 2.6 the photoswitching model has two main states: an on and an off state. This is enhanced by the presence of two transition states  $T_1$  and  $T_2$ . As can be seen movement from the transition states to the main two states is definite. The transition probabilities between the primary on and off states and the transition states is dependant on the fluorophore being observed. Figure taken from Frahm 2016.

### 3.4 Reconstruction Techniques

In general the concept of reconstruction relies on the idea that given an image with an unknown PSF model if one can fit some PSF model to it then the location of the point object that is emitting the light source can be determined. In general a PSF model is fit either by maximum likelihood estimate or least squared criterion with respect to the measurement error defined by  $\Delta_{loc} \approx \frac{\Delta}{\sqrt{N}}$  for an image with  $N$  photons (as we take  $N$  measurements) where  $\Delta$  is the size of our PSF (Huang, Bates, and Zhuang 2009). The resultant formulae  $I(x, y) = I_0 h(x - x_0, y - y_0) + b$ , where  $I_0$  is the peak intensity,  $h$  is the PSF of the imaging system,  $(x_0, y_0)$  is the location of the point light source and  $b$  is the mean background noise of the image, gives a relative measure of the intensity of the light at location  $(x, y)$  with respect to the peak intensity. In general the maximum likelihood method of fitting is preferred as it is far more accurate than the least squared solution (Mortensen et al. 2010). A less computationally expensive method for localisation is to simply take the centroid of the PSF, this is done using a method such as radial symmetry, however this is less accurate (Thompson, Larson, and Webb 2002).

Although it is fairly straightforward to localise a single point light source from a PSF, as previously highlighted the challenge emerges when there are more than one point light source close together, causing PSFs to amalgamate. In such cases, algorithms that are involved with single fluorophore fits have a rejection criteria built in that attempts to recognise a PSF-like shape that could potentially consist of more than one PSF and ignore it (Small and Stahlheber 2014). This can be done by looking at the shape of the PSF and determining how elliptical the shape is or by the fit of the model to that particular PSF (Starr, Stahlheber, and Small 2012). In terms of STORM, this means that any estimators built using the single localisation techniques will likely ignore any PSFs containing more than one fluorophore in each frame but due to the nature of the technique the majority fluorophore over the course of the imaging should at some point emit with a distance greater than the FWHM from any other fluorophore and therefore almost all fluorophores will be located (Small and Stahlheber 2014). If for some reason a number of fluorophores were missed, due to the potential large numbers of fluorophores present in any one STORM instance the difference to the overall, imaged structure would be negligible.

The other way to tackle the problem is to build algorithms that are able to localise several point light sources close together. This is done in a similar manner to the above method but instead of fitting a single PSF model to the image, a multi PSF model is fit to the image using maximum likelihood estimate (Quan et al. 2011). This is more computationally expensive than the single localisation method and is also more complicated with more parameters that need tuning. This can lead to PSF modelling that overfits to PSFs with two or more point light sources present and as a result may ignore single point light source PSFs (Small and Stahlheber 2014).

ThunderSTORM utilises a single fluorophore localisation approach with both a least squared and maximum likelihood estimation approach as well as a centroid approach (Ovesny et al. 2014).

### 3.5 Evaluation Techniques

In order to effectively judge how well reconstruction techniques fare on generated data evaluation metrics must be considered. In general the best approach given that the results are two sets of Cartesian coordinates is to measure the difference or distance between two sets. There are a number of evaluation techniques that achieve this. Foremost the Jaccard index, defined as the union of two sets over the intersection of the sets or more formally as: let  $A$  and  $B$  both be non-empty sets then,

$$\text{Jac}(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (3.2)$$



is a measure of the similarity of two sets. The complementary metric is the Jaccard distance which measures the dissimilarity between two sets. It is defined as:

$$d_{\text{Jac}}(A, B) = 1 - \text{Jac}(A, B) \quad (3.3)$$

The Jaccard measure returns a number from zero to one where one defines the two sets as being the same. It can therefore be used to evaluate how many correct locations a reconstruction algorithm managed in comparison to the ground truth, thus measuring the success rate of the estimator.

The second measure is the Hausdorff Distance. It attempts to measure how far two subsets are from one another in terms of some measurable space. Let  $A$  and  $B$  be two non-empty sets on a metric space  $d$ :

$$\text{Hausdorff}(A, B) = \max \left\{ \sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(a, b) \right\} \quad (3.4)$$

The Hausdorff metric is often used in computer vision because it measures how close a model lies to the true values (Huttenlocher, Klanderman, and Rucklidge 1993). In a similar fashion it is applicable to comparing the how well a reconstruction technique is able to place predicted values with respect to the ground truth values. A problem that exists with the Hausdorff distance is that it is very sensitive to outliers in the data (Dubuisson and Jain 1994).

An issue that exists in both of the above metrics (Jaccard index and Hausdorff distance) is that when the sets are of disparate sizes the results are not accurate. If the results of the reconstruction process produces more results than is actually present due to the simulated movement of fluorophores, then the result of the above metrics will not be accurate (Gardner et al. 2014). As a result an optimal mapping algorithm to map the closest matching members of each set is required. This can be done by solving the linear sum assignment problem which takes the two sets and a weight matrix defining the cost of mapping the members of one set to every member of the other set and returns two sets mapped to each other (i.e index zero of set zero is mapped to index zero of set one) (Munkres 1957). Once optimally mapped the results can then more accurately be compared using the Jaccard index and Hausdorff distance.

With matched results in two equally sized sets it is possible to also apply more traditional error measurements. One such measurement is the root squared mean error (RMSE), defined as

$$\text{RMSE}(\hat{\theta}) = \sqrt{\frac{1}{N} \sum_{i=1}^N (\theta - \hat{\theta})^2} \quad (3.5)$$

where  $\hat{\theta}$  is the predicted value,  $\theta$  is the actual value and  $N$  is the size of the set. In the case where the predicted and actual values are Cartesian coordinates:

$$\text{RMSE}(\hat{\theta}) = \sqrt{\frac{1}{N} \sum_{i=1}^N \|\theta - \hat{\theta}\|_2^2} \quad (3.6)$$

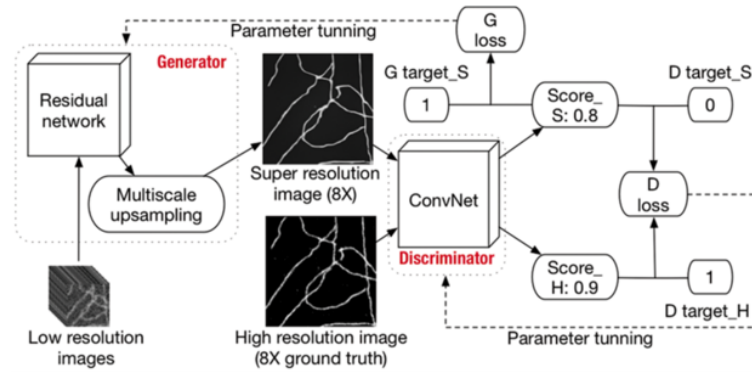
RSME gives an overall, average measure of the amount of error between matching members of two sets. In RSME the results are squared and square rooted this meaning that larger errors are penalised more.

## 3.6 Deep Learning techniques

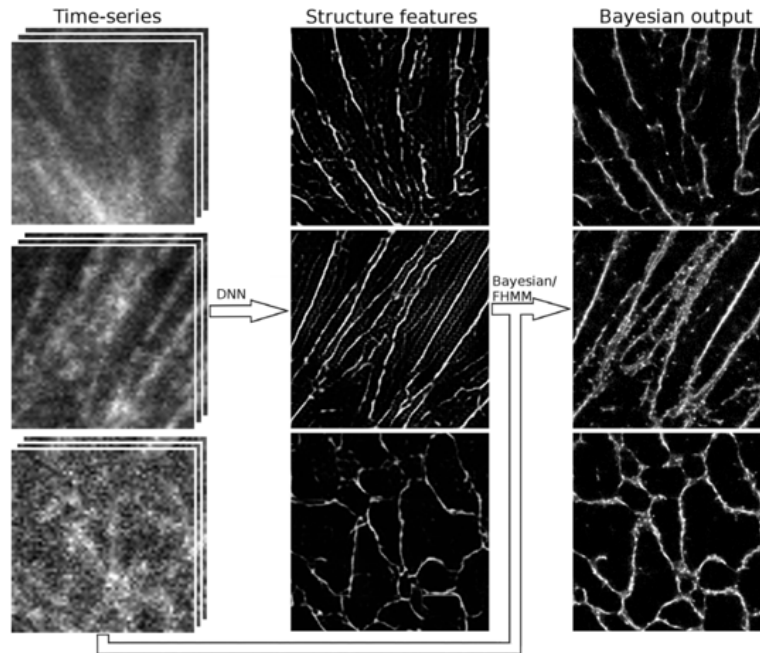
Deep learning has emerged as a leading form of machine learning particularly for image recognition tasks and in super resolution imaging (Lim et al. 2017). It is clear therefore that the power of these techniques could be harnessed and used for reconstruction purposes. Currently there

have been a number of groups that have attempted to tackle the problem of SRM reconstruction using deep learning techniques. These encompass three general approaches.

The first approach is to use a generative adversarial network (GAN); this involves the use of two neural networks, a generator network and an discriminator network that work against each other where the generator creates some form of data and the discriminator attempts to determine if the created data is correct or not (Goodfellow et al. 2014). Li et al. 2018 used a GAN to build a network that super resolves SRM image data and feeds the results into a separate model that performs Bayesian inference. The network architecture is highlighted in figure 3.2. The Bayesian inference model takes the final super resolved image from the GAN and the original low resolution input image and determines the location of the individual fluorophores present. Results of the network are promising as it is able to reproduce images better than established techniques such as 3B analysis (see figure 3.3).

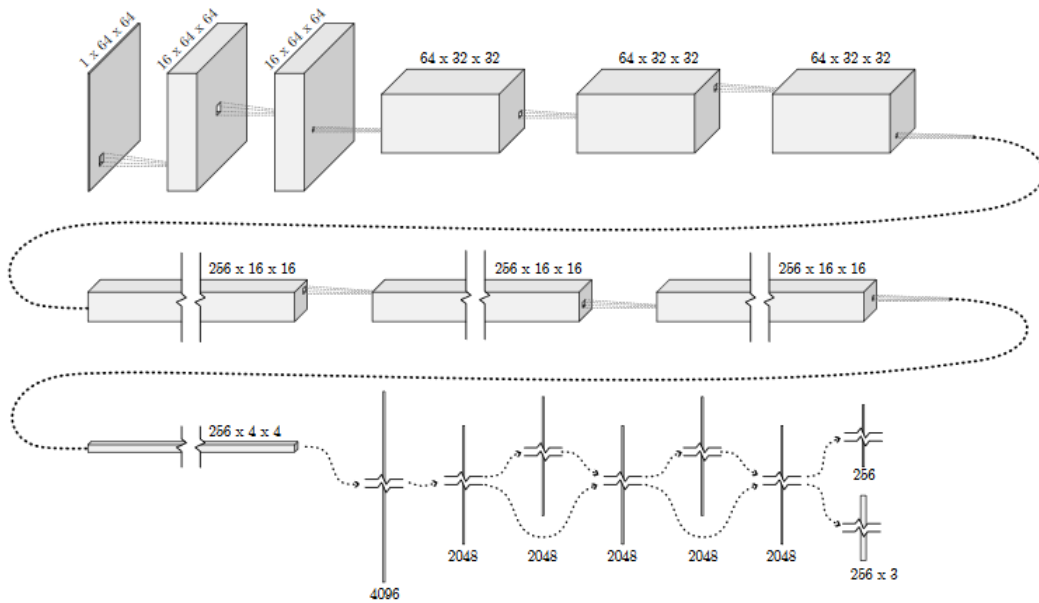


**Figure 3.2:** An overview of the network presented by Li et al. 2018. The generator network (G) takes low resolution SRM images and creates a super resolution image. The discriminator network (D) takes the super resolved image from G as well as the real ground truth high resolution image. D can compare these two images and feed the result back to G to tune it to create better super resolved images. In this way, the two networks work against one another. Figure taken from Li et al. 2018.



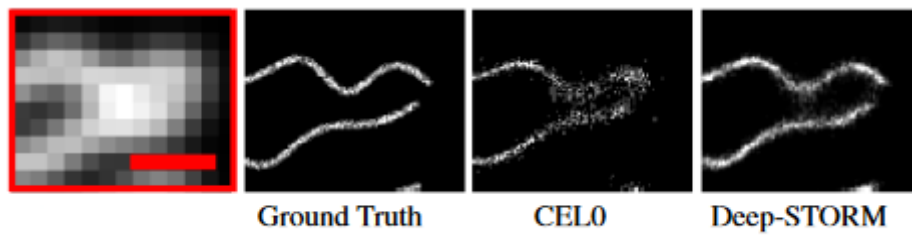
**Figure 3.3:** A depiction of the results from Li et al. 2018. The first column shows the original, low resolution input images, the second column shows the output of the GAN and the third shows the output of the Bayesian inference model. Figure taken from Li et al. 2018.

The second approach is outlined by Boyd et al. 2018 is to use a convolutional neural network with a section of dense layers at the end for STORM localisation. The architecture used in Boyd et al. 2018 is demonstrated in figure 3.4. The network uses residual connections in the dense layer in order to learn what is known as residuals which enables the network to be deeper; this is because unlike traditional networks learning does not become saturated or degrade in certain cases the deeper the network goes. This is achieved by creating sequential shortcuts in the network between layers (He et al. 2016). The output of the network used by Boyd et al. 2018 is twofold, the first output is a set of weights that equate to the number of fluorophores present in the image and the network's confidence of each fluorophore and the second output is a 2D (or 3D if working in 3D space) tensor with the locations of the fluorophores. Boyd et al. 2018 are keen to stress that the importance of their findings lie in their application of neural networks to the localisation problem in STORM and the loss function that they developed. Despite this when compared to established reconstruction algorithms in both 2D and 3D the network produces comparable results; it is clear that with tweaking of the architecture better results than established algorithms could be possible.



**Figure 3.4: A depiction of the architecture used by Boyd et al. 2018 for STORM localisation.** The network consists of layers of convolution followed by two-layer fully connected ResNet. Figure taken from Boyd et al. 2018.

The third approach is to use two convolutional networks stacked on top of one another in opposite directions such that an image is put in and an image comes out the other side. The primary idea behind these kinds of networks is to attempt to learn some features that are present in the data, to learn feature representation through unsupervised learning. The central area of the network, i.e the output of the first network and the input into the second network encodes the feature representation (Xu et al. 2016). For the task of localisation within the scope of STORM learning features of the image is similar to learning the PSF model as achieved using traditional reconstruction techniques. This is the technique employed by Nehme et al. 2018. The network takes in a set of small images and super resolves them to a single super resolved output image. The approach was compared to CEL0, a multi-emitter fitting algorithm much like those described in 3.4. In a number of situations the auto encoder performed better (see figure 3.5) (Nehme et al. 2018). As a result, an auto encoder approach is a viable reconstruction method.



**Figure 3.5:** An example of the results obtained by Nehme et al. 2018. Ground truth image is of simulated microtubules. The red boxed image shows the diffraction limited image. As can be seen images created by Deep STORM are much closer to ground truth than the baseline CEL0 method. Figure adapted from Nehme et al. 2018.

## Chapter 4

# Methods

### 4.1 Overview

The developed system, STORMSim, has been created in order to accurately model the chemical and physical processes that occur in the STORM technique. The system enables a user to enter in a selection of parameters that describe the STORM image they wish to create. STORMSim operates in nanometers and the resulting output is always a greyscale,  $400 \times 400 \times N$  pixel image with one colour channel where  $N$  is the number of frames in the video. The system generates from set parameters ground truth locations in Cartesian coordinates of molecules bounded within the specified ground truth size in nanometers set by the user. These molecules are then tagged by the system with some fluorophore, the features of which are defined by the user. Photoswitching of these fluorophores is then simulated over the  $N$  frames.  $N$  frames are then rendered by the program taking into account the specified PSF model. Additionally, a  $400 \times 400$  pixel frame is also created that shows the true ground truth locations of each of the molecules as a single delta function. The individual frames of the STORM simulation are sequentially, stitched together to create an overall image. A table of results is also created that shows the location of each of the molecules in the ground truth. The frame of ground truth locations, the overall image of the STORM simulation and the table of locations are then presented to the user of the plug in. A class diagram of the system is shown in figure 4.1.

A large number of simulations with singular varying parameters were created by the system and then were reconstructed using algorithms within ThunderSTORM. The results of these reconstructions were then compared with the ground truth values using a number of evaluation techniques, outlined in 3.5. A large amount of simulations with bounded, random parameters were then generated. These were then used to train an auto encoder neural network.

STORMSim - Class Diagram

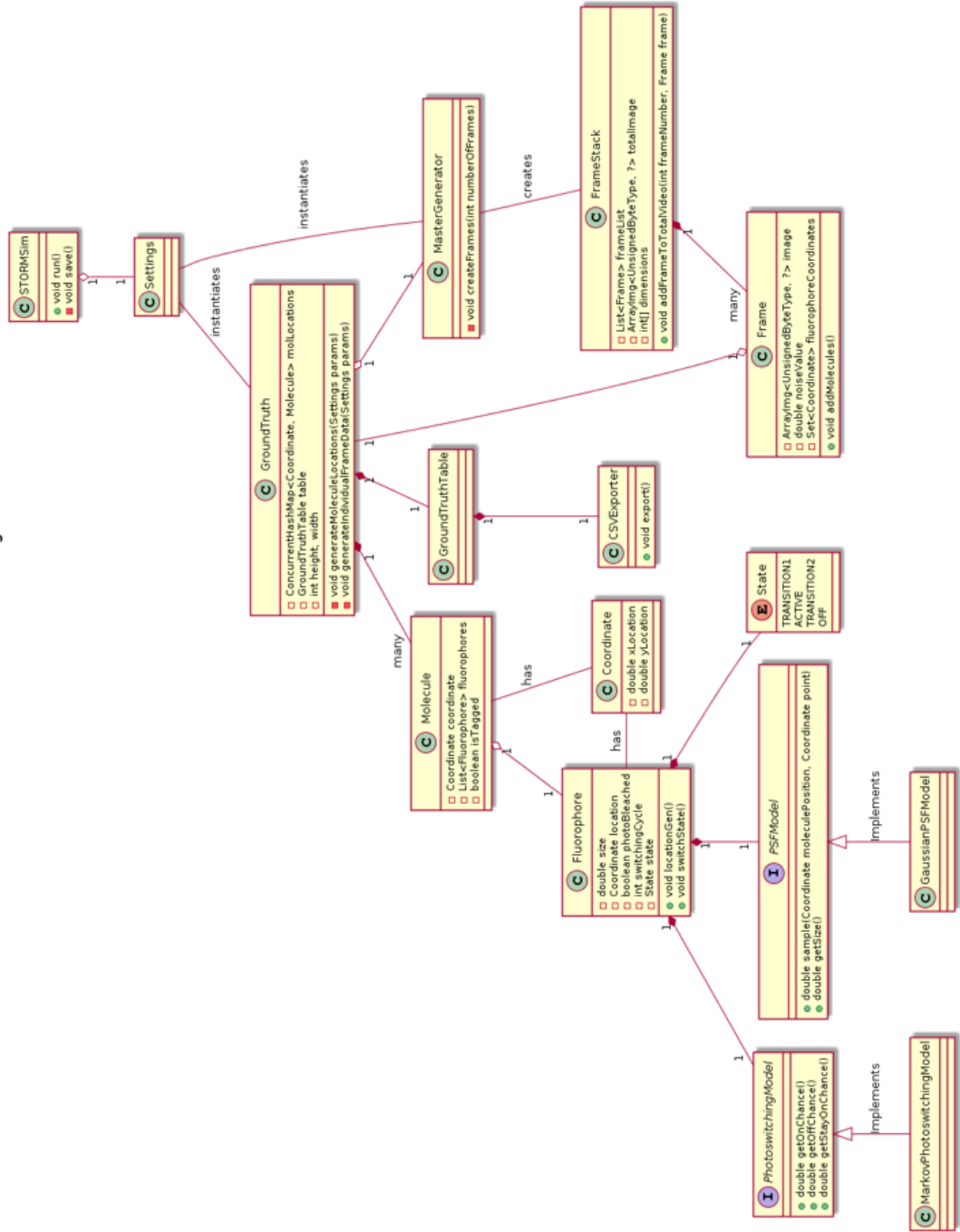


Figure 4.1: Class Diagram of STORMSim

## 4.2 STORMSim Requirements

### 4.2.1 Functional Requirements

1	General
1.1	The system will allow the user to generate STORM simulations dependant on a number of parameters
1.2	The system will allow the user to change these parameters
1.3	The simulations created will accurately reflect the physical and chemical processes in real STORM
2	Frame Rendering
2.1	The PSF will be simulated by the system
2.2	The system will model the photoswitching behaviour of fluorophores
2.3	Fluorophores will move between frames
2.4	The rendered frames will be in black and white
2.5	The system will allow the user to save the created image
2.6	Frames will be meaasured in nanometers
3	Results
3.1	The system will present users with the ground truth locations in the form of a table
3.2	The system will present users with the ground truth locations in the form of an image
3.3	The system will present users with the finalised stack of frames
3.4	The user may save the final image
3.5	The user may save the table in CSV format

### 4.2.2 Non functional Requirements

1	System
1.1	(Usability) The application will be in English
1.2	(Implementation) The system will be built with Java
1.3	(Implementation) The system will be built using the SciJava API
1.4	(Implementation) The system will be built as a plug in for ImageJ

## 4.3 Ground Truth generation

STORMSim generates ground truth locations in a random fashion. The user can define the number of molecules that they want in the ground truth as well as the chances of the molecules being tagged. The generation of the locations of the individual molecule locations as well as if they are tagged with a fluorophore or not is based on the algorithm described in Boyd et al. 2018 and is highlighted here in algorithm 1.

## 4.4 Fluorophores

Fluorophores are represented within the software as an object with a parent molecule which they are attached to, a location defined as a Cartesian coordinate, an intensity (photons), a PSF

**Algorithm 1** Molecule Location

---

```

for number of molecules do
   $m \leftarrow$  Molecule
   $x \leftarrow$  sample from uniform distribution bounded by width of frame
   $y \leftarrow$  sample from uniform distribution bounded by width of frame
  if not first iteration then
    while a molecule is already at the position  $(x, y)$  do
      generate new  $x$  and  $y$  as above
    end while
  end if
   $c \leftarrow$  a random number from 0 to 1
  if  $c \geq$  tag chance then
    set  $m$  as not tagged
  else
    tag molecule with fluorophore
  end if
  add molecule to ground truth
end for

```

---

model, a switching cycle and a state. The fluorophore states are on, off, transition one, transition two and photobleached.

## 4.5 Photoswitching Model

The photoswitching model used in STORMSim is a Markov model defined by two transitional states, an off state, an on state and a photobleached state. This follows from the research outlined in 3.3. The Markov model is depicted in figure . The user of the system is able to chance the on chance, the off chance and the switching cycle which determines the average number of cycles before photobleaching (see 2.1). As described in Frahm 2016 the transitional states are deterministic and therefore always move to the active or off state. The photoswitching model was chosen to be an interface in order that different kinds of photoswitching models such as a Poisson model may be implemented in the future. This model was chosen because it has been readily used as the standard model for photoswitching described in literature (Frahm 2016; Li et al. 2018; Sinkó et al. 2014). The algorithm for this process is described in algorithm 2.

## 4.6 PSF Model

The PSF model implemented in the software is the 2D Gaussian model that has height one, defined by the equation 3.1 (repeated below for clarity):

$$G(x, y) = e^{-\frac{((x-x_0)^2 + (y-y_0)^2)}{2\sigma^2}} \quad (4.1)$$

where  $x$  and  $y$  represent the location of the current location,  $x_0$  and  $y_0$  represent the location of the point,  $e$  is Euler's constant and  $\sigma$  is some variance that describes the width of the PSF. The variance is a parameter that can be changed by the user of the software. The PSF model is implemented as an interface so in a similar fashion to the Photoswitching model new models may easily be added. This model of PSF was chosen as it was described in literature as being a good approximation of the real PSF without being too numerically complex and therefore much quicker to calculate (Small and Stahlheber 2014; Stallinga and Rieger 2010).



---

**Algorithm 2** Markov Photoswitching
 

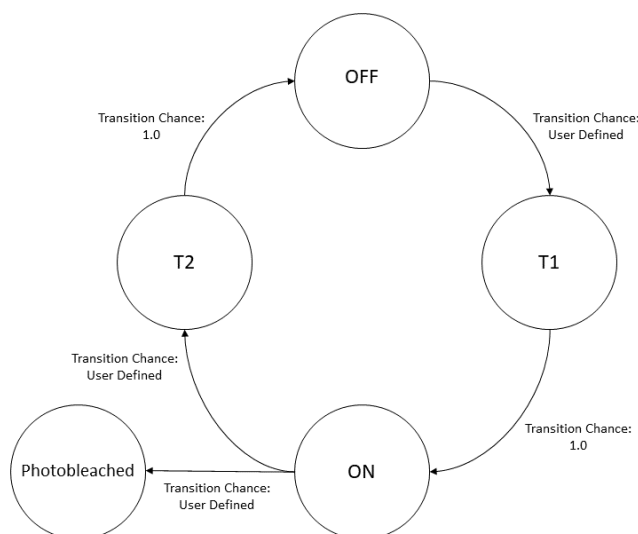
---

```

for frame in all_STORMframes do
  for tagged molecules do
    for fluorophores tagged to the molecule do
      if fluorophore is not photobleached then
         $s \leftarrow \text{fluorophore.state}$ 
        if  $s$  is active then
          if  $\text{off chance} \geq \text{generated random chance}$  then
            fluorophore.switch_state
            ▷ state is now T2
          end if
          Add copy of fluorophore to frame's render list
        else if  $s$  is off then
          if  $\text{on chance} \geq \text{generated random chance}$  then
            fluorophore.switch_state
            ▷ state is now T1
            Add copy of fluorophore to frame's render list
          end if
        else
          ▷  $s$  is either T1 or T2
          fluorophore.switch_state
          ▷ state is now either off or active
          if  $s$  is active then
            Add copy of fluorophore to frame's render list
          end if
          if  $s$  is off then
            if number of total switches == switching cycle then
               $p \leftarrow \text{Poisson distribution with mean at switching cycle}$ 
              get cumulative probability of  $p$  with current cycle number
              if cumulative probability  $\geq$  random chance then
                fluorophore is photobleached
              end if
            end if
          end if
        end if
      end if
    end for
  end for
end for

```

---



**Figure 4.2: The Markov photoswitching model used in STORMSim** This model is adapted from one presented in Frahm 2016. There are four states: on, off, T1, T2 and photobleached. The fluorophore may move between these states dependant on the chances shown next to each arrow. The transitional states are deterministic and will always move to the next state. The photobleached state is an end state and is irreversible.

## 4.7 Frame and Frame Stack construction

The frame is constructed by primarily utilising the imglib2 subset of the SciJava API. The frame class is the parent class of two classes, the GroundTruthFrame, which is the frame that shows the ground truth location of each of the molecules and the STORMFrame, which is the frame that makes up an individual frame within the overall STORM simulation.

The image of the frame is created using the ArrayImg factory class from SciJava; this generates a 400x400 size ArrayImg of type UnsignedByteType. The UnsignedByteType is a wrapper type that represents an unsigned byte (a type that is not possible in Java otherwise) bounded between 0 and 255. Every value is originally set to 0 in the ArrayImg. The value of the UnsignedByteType is equal to the level of white in that pixel, that is to say at a value of 0 that pixel is completely black and at a value of 255 the pixel is completely white; as a result the original, created ArrayImg is a 400x400 black square.

STORMFrames contain an id which corresponds to their location within the time frame, a noise value and a list of fluorophores (determined by algorithm 1) in addition to the aforementioned ArrayImg. The STORMFrame rendering algorithm is presented in algorithm 3

The algorithm described above takes each fluorophore within the frame and firstly generates a new location for the fluorophore; this is calculated using two normal distributions: one over the X axis with a central point at the molecule location with variance of the size of the fluorophore; and similarly over the Y axis with the same central point and variance. Each distribution is sampled to give an X and Y coordinate in a radius around the central molecule location, this is then set as the new fluorophore coordinate.

After generating a location, originally the algorithm iterated over the every pixel in the ArrayImg however this caused performance issues given that a large number of fluorophores may be present in one frame and there may also be a large number of frames. Due to the limitation of the SciJava API it was not possible to index into the ArrayImg like traditional arrays. As a result, this challenge was tackled by calculating a SciJava interval that can be used alongside an iterator to iterate over a specified section of the ArrayImg. The interval is calculated by firstly

**Algorithm 3** STORMFrame rendering

---

```

for  $f$  in frame_fluorophores do
  generate new a location for  $f$ 
   $f\_loc \leftarrow f.location$ 
  while  $f\_loc$  is already occupied do
    generate new a location for  $f$ 
  end while
  interval  $\leftarrow$  calculate interval to iterate
  for pixel in interval do
     $p\_loc \leftarrow$  convert current pixel location to nm
    intensity_factor  $\leftarrow$  calculate PSFmodel with  $f\_loc$  and  $p\_loc$ 
    expected  $\leftarrow f.intensity * intensity\_factor$ 
    actual  $\leftarrow$  sample from Poisson distribution with mean at expected
    pixel  $\leftarrow$  convert actual to a pixel value rather than intensity
  end for
end for
if noise present then
  for pixel in image do
    noise_sample  $\leftarrow$  sample Poisson distribution with mean at noise_value
    pixel  $\leftarrow$  convert noise_sample to pixel value + pixel
  end for
end if

```

---

locating the fluorophore in pixel space by converting its nm location to a pixel location. The PSFModel variance, and thus its width is taken and converted from nm to pixel. This value is then square rooted to get the FWHM of the PSF. This value is then subtracted from the central fluorophore pixel position in both the X and Y axis, this is then the starting point of the interval as it is top left corner of the pixels that encompass the PSF. The length of the interval in the X and Y dimension is then set to equal the width of the PSF in pixels. Thus a square interval area that equates to the area in which the PSF operated is generated.

This interval is then iterated over and the intensity factor for the pixels is calculated using the PSF function described in 3.1. This intensity factor is then multiplied by the intensity in photons of the fluorophore, which is normalised in relation to its state, to get an expected intensity for that pixel and that fluorophore. A Poisson distribution with mean of this expected value is then sampled from in order to account for background interference from out-of-focus fluorophores or the imaging medium's inaccuracies (Small and Stahlheber 2014). This is then converted to an UnsignedByteType value and added to the current pixel's value (values are normalised with respect to the bounds of the UnsignedByteType, if the value is over 255 it is just set to 255). The GroundTruth frame's construction works in a similar manner to the STORMFrame's rendering. The location of each molecule rather than its tagging fluorophore is converted from nm to pixel. This value is rounded to its closest value representing the pixel closest to the molecule. This pixel is then set to an UnsignedByteType value of 255.

The FrameStack is created also as an ArrayImg with a Z dimension equal to the number of frames present. Optimisation here was also an issue, originally it was a slow, limiting step to stitch together the individual frames of the simulation because it was done at the end of the process. However this was fixed by giving each frame an id that corresponded to its location in a sequential time frame and running the creation of each of these frames on separate threads. Once a frame is rendered in a thread the thread then writes its result to the correct location pertaining to its id in the final frame stack, e.g if the frame had an id number of 4 it would be written to the fourth layer in the Z dimension in the frame stack. As each thread only writes to an particular Z dimension in the frame stack there is no need to make this method synchronised as concurrent access of the same location will never happen. The frame stack counts up the number of frames

that are completed and once this has reached the total number of frames specified by the user, the frame stack rendering process is marked as done.

## 4.8 Results Table

The results table is represented also by a built in SciJava class, the DefaultTable class. This allows ImageJ to display a table like set of data. The results table every time a molecule is generated in the ground truth takes that molecule and adds its information to its columns. The table has 3 columns: Molecule number, x Coordinate (nm) and y Coordinate (nm). The problem encountered with ImageJ is that there is no built in way for ImageJ to save the results unlike with images. A CSV exporter was built instead, this takes the table and iterates over it extracting the information, delimiting this with commas and generating a CSV file representing the table.

## 4.9 User Interface

The user interface is demonstrated in figure 4.3.

The screenshot shows the STORMSim user interface window. It is organized into several sections with settings for simulation parameters. The 'Fluorophore Settings' section includes 'Size of fluorophore (nm)' (30), 'Switching Cycle' (10), and 'Intensity (Photon count)' (300). The 'STORMFrame Settings' section includes 'Height & Width of frame (nm)' (32,000), 'Number of molecules' (10), 'Number of frames' (10), and 'Noise value (mean photon count)' (30). The 'PSF Settings' section includes 'Type of PSF model to use' (Gaussian) and 'PSF FWHM (nm)' (200). The 'Photoswitching Model Settings' section includes 'Type of photoswitching model to use' (Markov), 'On chance' (0.9), 'Off chance' (0.8), and 'Tag chance' (0.99). At the bottom, there is a 'Saving' section with checkboxes for 'Save STORMImage?' and 'Save results?', and a 'Save location' field with a file explorer icon. An 'OK' button is at the very bottom.

**Figure 4.3: The user interface of STORMSim** As seen there are a large number of changeable parameters present within the plugin. These can be changed within sensible limits, e.g you cannot have a chance greater than 1.0.

## 4.10 ThunderSTORM reconstruction

ThunderSTORM reconstruction techniques were run on data generated by STORMSim to test the effectiveness of the simulation by bench-marking it against tested reconstruction techniques. The first step was to generate experimental data with STORMSim with altered parameters. The

base parameter values are shown in table 4.5. The parameters of molecule number, fluorophore size, noise value and PSF FWHM were changed incrementally in separate experiments and the image data in addition to the results data was saved.

Reconstruction with ThunderSTORM was done using a Gaussian fit model with a least squares method and MLE method with a fitting radius of 3 pixels. The centroid fitting approach was not used as it is much less effective than the other methods (Small and Stahlheber 2014). The camera settings were set to a pixel size that matched the size of the frame, for 32000nm frame size this was 80nm a pixel, the photoelectrons per A/D count was set to 1 and the base level in A/D counts was set to 0. The data was fed to ThunderSTORM using an ImageJ macro, this greatly increased the speed of getting results. The results were then exported and evaluation techniques were used to evaluate the effectiveness of the reconstruction process.

Parameter	Value
Frame Size (nm)	32000
Number of molecules	100
On chance	0.25
Off chance	0.8
Number of frames	10
Tag chance	0.95
PSF FWHM (nm)	200
Fluorophore size (nm)	10
Noise value (intensity)	50
Switching Cycle	10

Table 4.5: A table showing the baseline parameter values used for experimentation

## 4.11 Evaluation Techniques

The evaluation techniques were implemented within Python because of the large number of preexisting libraries that are designed for the metrics used and for data manipulation. The evaluation methods described in chapter 3.5 were used. They are Jaccard Index, Hausdorff distance and RSME. The Jaccard index and Hausdorff distance were calculated without an optimal mapping approach first. However due to the fact that the produced reconstruction results contained more members than the ground truth, as described in 3.5, the metric results were inaccurate. Additionally, with two, unmatched sets of unequal size RSME was not possible. As a result, using the work described in Gardner et al. 2014 optimal mapping was performed using linear sum assignment from the SciPy library using the cost function,

$$c(x, y) = \min \left[ 1, \frac{m_D(x, y)}{v(A)} \right] + \min \left[ 1, \frac{m_D(x, y)}{v(B)} \right] + \min \left[ 1, \frac{m_D(x, y)}{v(A \cup B)} \right] \quad (4.2)$$

defined by Gardner et al. 2014 where  $x$  and  $y$  are coordinates,  $A$  and  $B$  are sets of coordinates,  $v$  is the size of, and  $m_D$  is the euclidean distance between  $x$  and  $y$ . RSME, Jaccard and Hausdorff were then calculated using the results of the optimal mapping.

### 4.11.1 Jaccard Index

The Jaccard index defined in 3.2 was adapted in a similar fashion to one described in Boyd et al. 2018 where a threshold for an exact match is used instead of using exact match between points. The algorithm for calculating the intersection count between the mapped and non mapped sets is slightly different. These are demonstrated in algorithm 4 and algorithm 5 respectively.

The Jaccard index is then calculated using the equation outlined in 3.2.

---

**Algorithm 4** Mapped intersection

---

```

 $A \leftarrow$  array of ground truth coordinates
 $B \leftarrow$  array of reconstructed coordinates
count  $\leftarrow$  0
for  $i = 0; i < A.length < i = i + 1$  do
    if euclidean distance between  $A[i]$  and  $B[i] \leq$  threshold then
        count  $\leftarrow$  count + 1
    end if
end for
return count

```

---



---

**Algorithm 5** Non mapped intersection

---

```

 $A \leftarrow$  set of reconstructed coordinates
 $B \leftarrow$  set of ground truth coordinates
count  $\leftarrow$  0
for reconstructed in  $A$  do
    best  $\leftarrow$  inf
    for ground truth in  $B$  do
        distance  $\leftarrow$  euclidean distance between ground truth and reconstruction
        if distance  $\leq$  threshold and distance < best then best  $\leftarrow$  reconstruction
        end if
    end for
    if best  $\neq$  inf then count  $\leftarrow$   $i + 1$ 
    end if
end for
return count

```

---

**4.11.2 Hausdorff Distance**

The Hausdorff distance described by the equation 3.4 was used as the second metric for comparison. The algorithm used is demonstrated in algorithm 6.

---

**Algorithm 6** Hausdorff Distance

---

```

 $A \leftarrow$  set of reconstructed coordinates
 $B \leftarrow$  set of ground truth coordinates
distance list  $\leftarrow$  list
for reconstructed in  $A$  do
    shortest  $\leftarrow$  inf
    for ground truth in  $B$  do
        distance  $\leftarrow$  euclidean distance between reconstructed and ground truth
        if distance < shortest then
            shortest = distance
        end if
    end for
    append shortest to distance list
end for
return maximum of distance list

```

---

### 4.11.3 RSME

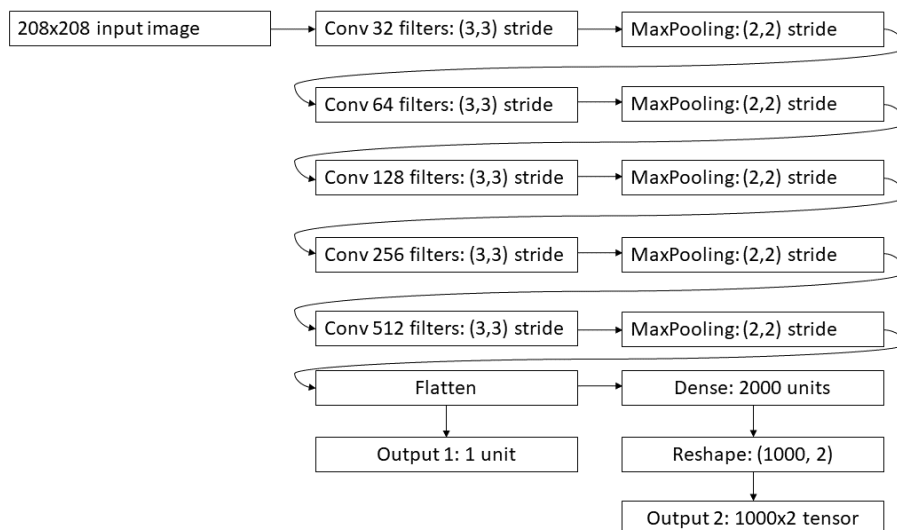
RSME calculation follows the formulae 3.5. The error is measured as the euclidean distance between the two matching points resolved by the optimal mapping. The error is squared, summed and then square rooted.

## 4.12 Deep Learning

Although the focus of this project was to build an accurate STORM simulator, there was scope to develop deep learning methods to perform the reconstruction phase of STORM. Two approaches were taken with this: a convolutional neural network approach with dense later layers following the architecture outlined in Boyd et al. 2018 and a auto encoder approach with a similar architecture to Nehme et al. 2018. Data for training and valudation of the neural networks was generated by STORMSim. The data created had the variables of molecule number, noise, PSF FWHM, on chance, off chance, tag chance, fluorophore size and switching cycle randomly changed to random, variable specific bounded values. Data was split into train and test with 75% of the data for training and 25% for testing purposes.

### 4.12.1 Convolutional Neural Network

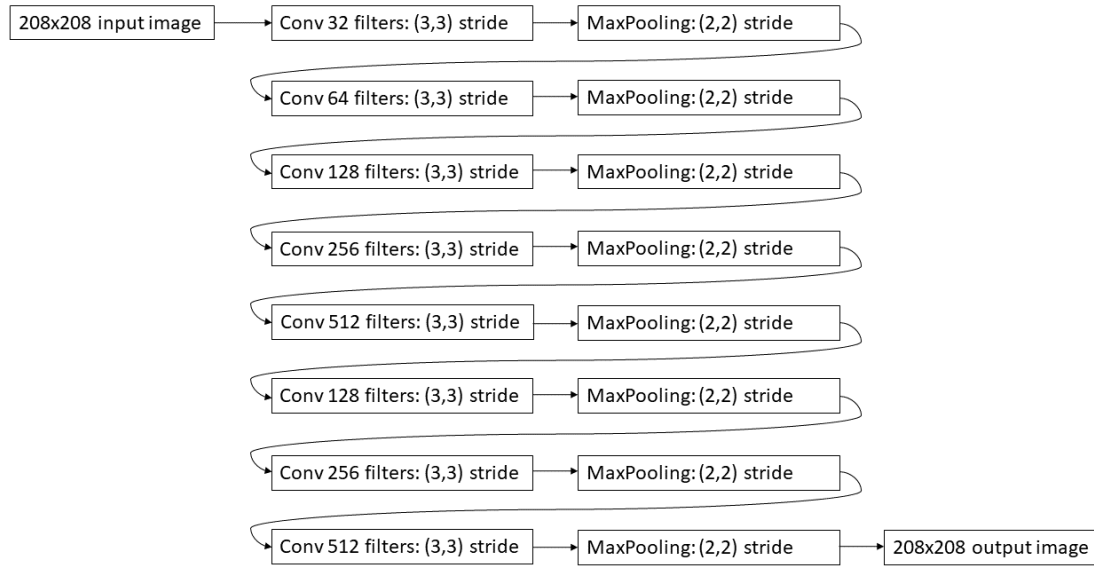
The architecture of this network is demonstrated in 4.4. As an input it takes a 208x208 single channel image for the  $x$  axis, on the  $y$  axis there are two inputs the first is the molecule count of the image, the second is the Cartesian locations of the present molecules. Although the output of the network has 1000 locations in practice there was not always 1000 molecules present in the image, in such a case the input would read (0, 0). The network was trained using Adam. The loss function used was a tensor implementation of Cartesian RSME described earlier in 4.11.3.



**Figure 4.4: A depiction of the convolutional neural network architecture** This architecture is loosely based on the architecture presented in Boyd et al. 2018. The network takes a 208x208 image with one colour channel as its input. It has two outputs, an output with 1 unit which is designed to output a count of the number of molecules present in an image and the second output is a 2x1000 tensor representing the Cartesian coordinate locations of the molecules in the frame. Each layer uses a ReLu activation function. After each convolutional layer batch normalisation is applied.

### 4.12.2 Auto Encoder Neural Network

This architecture is depicted in 4.5. This is based off of Nehme et al. 2018. The network's input is a 208x208 single channelled image and the output is the same. The network's x axis training data is a single frame from a STORMSim simulation and its y axis is the same frame. The idea here being that the network attempts to learn features within the frame. Because the features in a STORMSim frame are models of PSF, i.e hopefully the network is learning the Gaussian noise PSF model. The network was trained using Adam with binary cross entropy as the loss function.



**Figure 4.5: The auto encoder architecture used** This architecture is adapted from one presented in Nehme et al. 2018. ReLu is used as the activation function between each layer. Between each convolutional layer batch normalisation is applied



# Chapter 5

## Results

### 5.1 STORMSim

An example of the data that STORMSim generates is presented in figure 5.1. This figure shows the kind of STORM data that can be produced by STORMSim using the baseline parameters outlined in figure 4.5. The stochastic, photoswitching nature of the fluorophores is demonstrated here as between frames the individual fluorophores can be observed changing intensity. This is further highlighted by figure 5.2 which demonstrates up close a single fluorophore during the 4 emission states outlined in chapter 4.5.

In terms of modelling PSF, figure 5.3 shows a series of frames with a height and width of 2000nm and the PSF model is clearly visible here. Moreover, the PSF is also clearly visible in all other examples in this section. Figure 5.3 also highlights how close together molecules are indistinguishable from one another to the naked eye. Additionally, the PSF is shown as a clearly visible Gaussian blur function at both this resolution and also at the wider resolution demonstrated in 5.1. Figure 5.4 demonstrates the handling of noise by the system and how that affects the way that affects the overall image. As seen in figure 5.4 noise is not a uniform value across the entirety of the frame, this follows algorithm 3 which models noise from a Poisson distribution.

The effect of changing size is demonstrated by the difference between the images in figure 5.1 and figure 5.3. The effect of changing the FWHM is shown in figure 5.5. This shows that increasing the FWHM of the fluorophores increases the size of the rendered PSF. Figure 5.6 demonstrates the outputs of the system including the results table.

### 5.2 Reconstruction Techniques

All raw data for the results displayed here can be seen in appendix C.

#### 5.2.1 MLE

When the ThunderSTORM MLE reconstruction technique was applied to a series of simulations where the size of fluorophore was changed there was a reduction in the accuracy of the reconstruction techniques according to the Jaccard Index from 0.22 with a fluorophore size of 20nm to 0.10 with a fluorophore size of 150nm, see 5.7. This is true both for when the results are optimally mapped (where they peak at 0.8 at a fluorophore size of 40nm and fall to 0.29 at a fluorophore size of 150nm) and for the when they are not mapped (where they peak at 0.24 at 40nm fluorophore size and reach their lowest point of 0.1 at 150nm fluorophore size). However, there does not appear to be a clear trend with the Hausdorff distance when it is unmapped, fluctuating around an average error of 3152.32. Similarly there is no discernible trend for RSME

either. The mapped metric versions of the Jaccard and Hausdorff performed better than the non mapped versions.

The same approach as above was taken with noise and the results are presented in 5.8. As demonstrated, the optimal mapping approaches as expected performed better than the non mapped approaches for Jaccard and Hausdorff. Although there is no trend in the optimal mapping versions of Jaccard and Hausdorff, they either fluctuate a lot in the case of Jaccard, with an average value of 0.80, or generally remain low in the case of Hausdorff, there is a negative trend in non optimal Jaccard and a positive trend in the non mapped Hausdorff; both of these imply that the metric is returning a worse result. RMSE like optimised Jaccard and Hausdorff does not have a trend.

This was repeated with the FWHM value of the PSF, the results of which are displayed in figure 5.9. It appears that as the size of the PSF is increased when the results were not mapped there was generally a trend for the reconstruction to be worse, demonstrated by the positive trend in Hausdorff and the sharp negative drop from 0.23 at a FWHM size of 240nm to the reading of 0.14 at a FWHM size of 290nm, in Jaccard. However, when the results are optimally mapped there is no clear trend, shown by the fluctuating data present for RSME, Jaccard (optimal) and Hausdorff (optimal).

The results of applying ThunderSTORM MLE reconstruction to data with varying numbers of molecules are shown in figure 5.10. The data shows that as the number of molecules increased the Jaccard index when optimally mapped and when not both demonstrated a negative trend. This trend was shared by the optimally mapped Hausdorff which also showed as more molecules were present the estimator produced more error; however, the non optimally mapped Hausdorff distance and the RSME both have trends which suggest that the estimator got more accurate as more molecules were present.

### 5.2.2 Least Squares

The results of using least squared reconstruction method for simulations with changing fluorophore size are presented in 5.11. These show that as fluorophore size increases Jaccard index both when mapped and when not decreases. Hausdorff in both forms and RSME all show no particular trend. This is very similar to the trend observed in the corresponding least squares reconstruction shown in figure 5.7.

Using least squared reconstruction techniques and changing the amount of noise is correlated with increased error when measured without optimal mapping. When the data is mapped there is no obvious trend for all three mapped metrics (Jaccard, Hausdorff and RSME). The data is shown in figure 5.12.

Changing the size of the PSF by increasing the FWHM and using the least squared metric for reconstruction shows that in both the optimal mapping and the non mapping approach there is little change with respect to the similarity of the real, ground truth and the reconstructed values. This data is presented in 5.13.

The outcome of altering the number of molecules in the image is depicted in figure 5.14. These results show that as the number of molecules increase the accuracy of the reconstruction appears to decrease with respect to the Jaccard index in both the optimal and non optimal cases. The non optimal Hausdorff however peaks at the reading of 2665 for 100 molecules and decreases from here to 0.21 at 400 molecules. This is not reciprocated by the corresponding optimally mapped Hausdorff which shows a positive trend, however the RSME similarly shows a decrease in error as the number of molecules increases.

### 5.2.3 Comparison

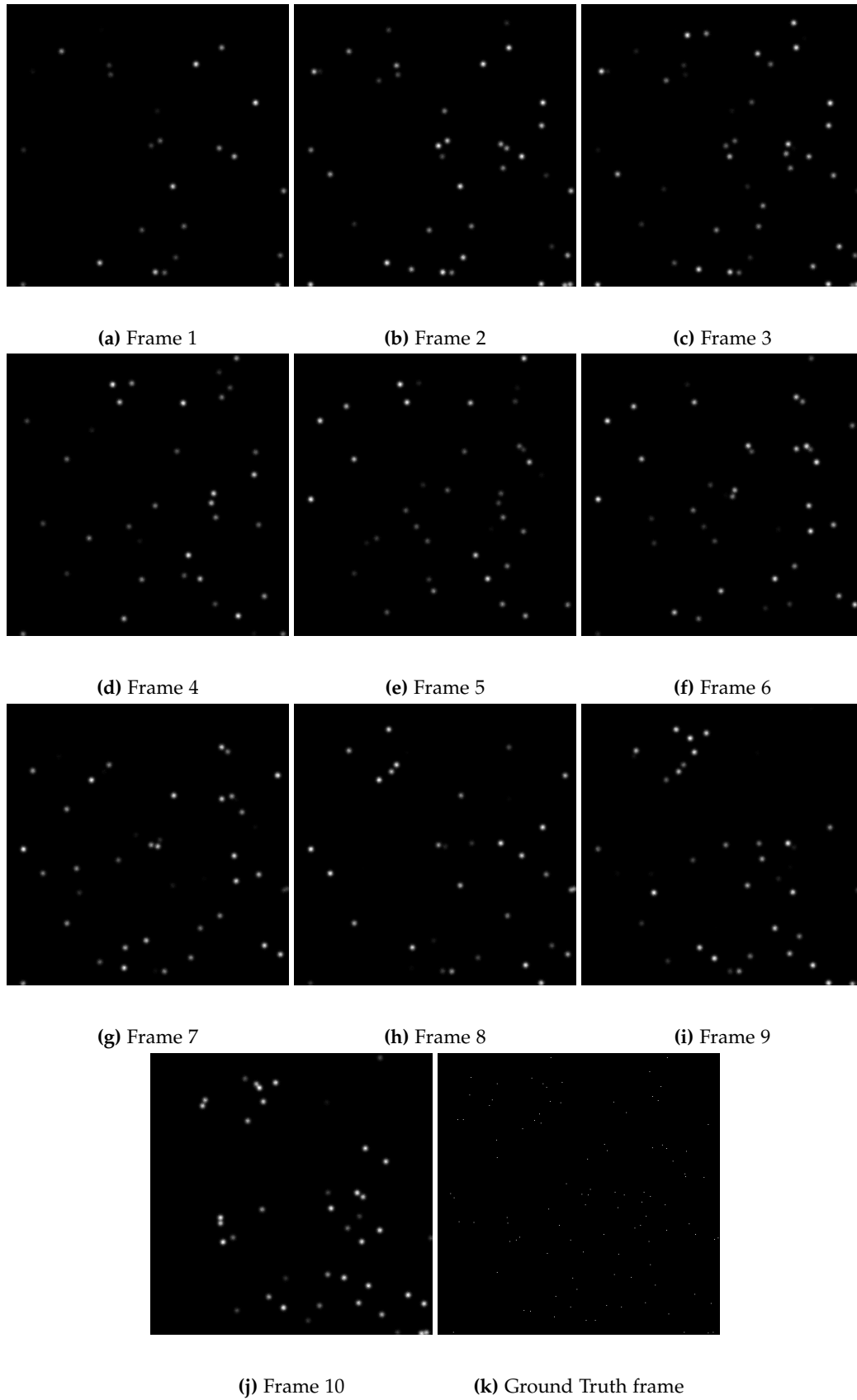
Both the least squared and MLE methods show considerable difference in the errors with respect to when the metrics were optimally mapped or not. The average non mapped Jaccard error across both the MLE (0.20) and least squared (0.18) was 0.19 which indicates a very low similarity between the reconstructed values and the ground truth values. In comparison when these values

are mapped the average Jaccard score increases to 0.75, 0.74 for the MLE values and 0.76 for the least squares values, this suggests that there is a similarity of 75% between the ground truth and reconstructed values. Equally, the average Hausdorff error is reduced from 2227.30 to 319.90 when optimal mapping occurs.

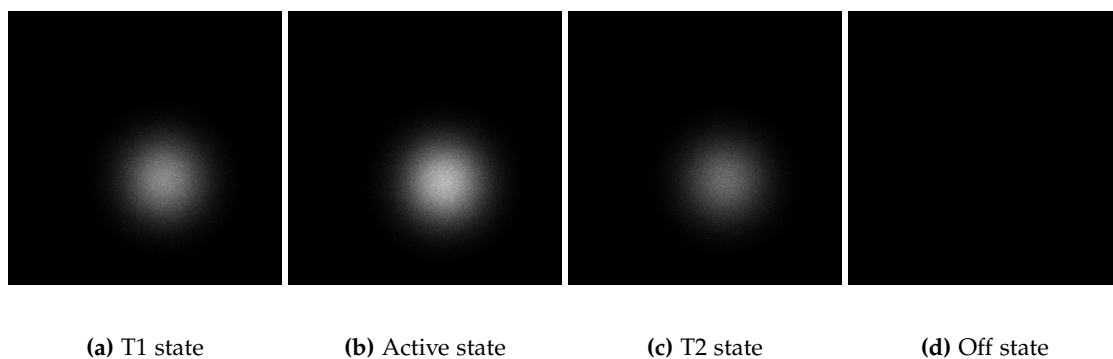
Although the MLE reconstruction method has higher error with respect to Hausdorff when mapped and when not in comparison to least squares, in general it is the better metric having a higher average Jaccard Index in both the mapped and non mapped versions, similarly for RSME, see table 5.15. However the difference between these two reconstruction methods is slight. This is highlighted by the only very slight improvements observed in the raw average error data shown in appendix C.

### 5.3 Deep Learning

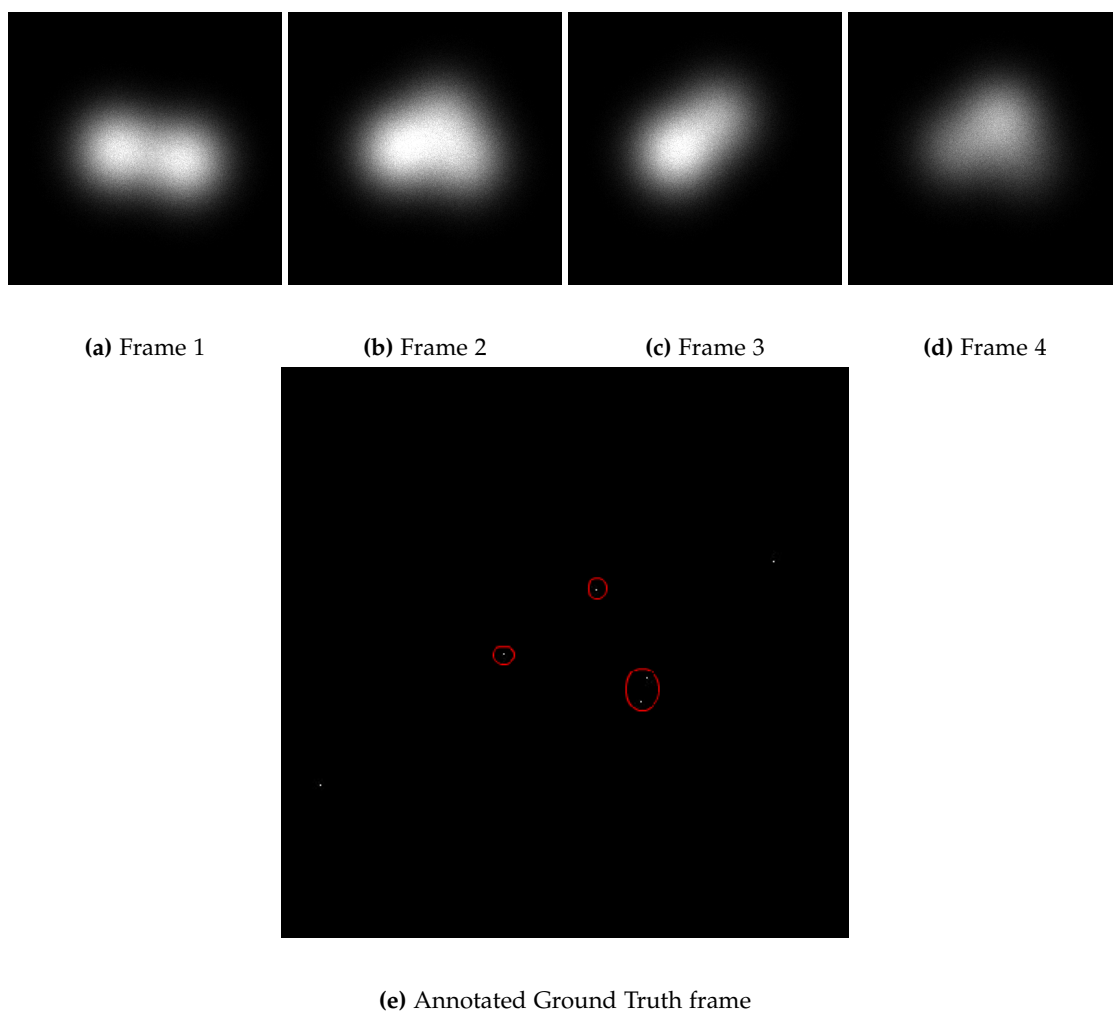
Due to the fact that the main contribution with this project lies in the creation of a simulator, the deep learning methods employed were limited. Both approaches outlined in chapter 4.12 were used. The first network used was an auto encoder approach using a learning rate of 0.01 with Adam. The structure of this network is highlighted in 4.5. This network had a loss of 0.501 with binary cross entropy and after training for 100 epochs with 400 steps per epoch the network produced images like those shown in figure 5.16. These results suggest that the network has learnt representations of the Gaussian PSF spread function. The network architecture was then adapted and a dense later set of layers replaced the deencoder end of the network with an output being a 2d tensor. The output layer originally had a sigmoid activation function but this became saturated and the model would not learn due to saturation and therefore it was changed to a ReLu output. Using a Cartesian RSME loss function this model was trained with Adam and a learning rate of 0.01 and was trained with a batch size of 32 with 50 epochs. Although the model learnt, loss was very high, at 23622.



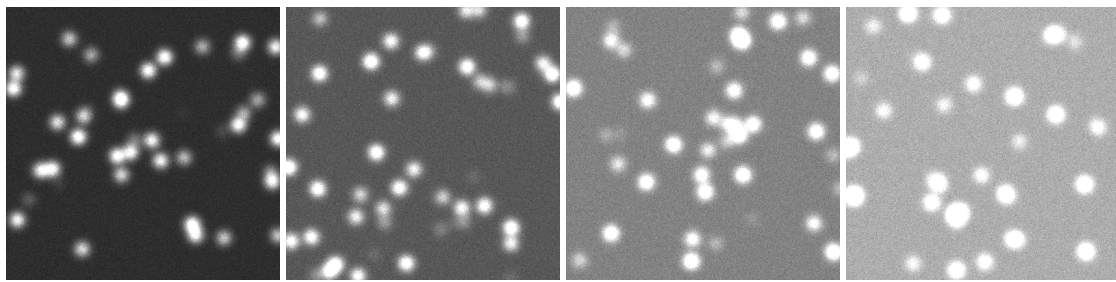
**Figure 5.1: An example STORMSim simulation** The parameters used to generate this data are those described in 4.5 expect mean noise count is set to 0 here. The frames are ordered sequentially in time. It can be clearly seen that there is stochastic blinking of fluorophores across frames.



**Figure 5.2: Example data demonstrating the photoswitching modelling** The parameters used to generate this data is the same as 4.5 expect mean noise count is set to 0, frame size is set to 2000 and number of molecules is set to 1. The sequential frames show how the different states in the fluorophore occur over a time period. As time goes on the fluorophore follows the Markov model proposed in figure 4.2 with 4 states: transition 1, on, transition 2 and off. This is demonstrated in this order in these frames from (a) through to (d)

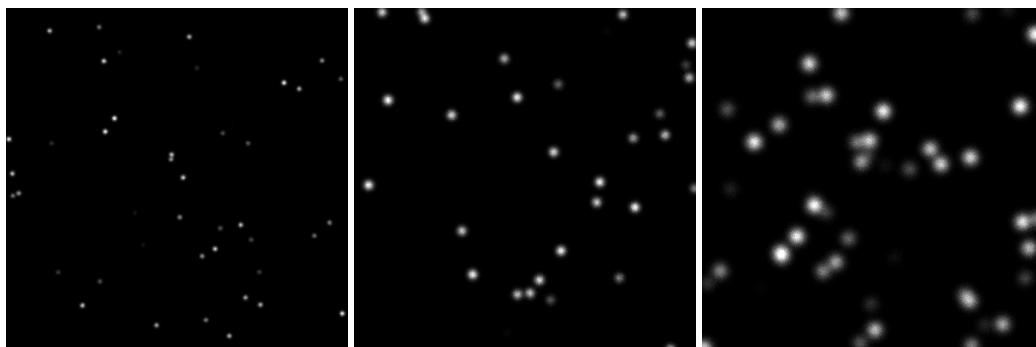


**Figure 5.3: An example of the PSF model** This shows a simulation made at a frame size of 2000nm with 6 molecules with a PSF FWHM of 200nm. The frames are ordered sequentially in time. It shows that molecules close together even at this close resolution are sometimes indistinguishable from one another with simply the human eye whilst also demonstrating the systems ability to represent overlapping PSFs. The 4 molecules highlighted in the central area are flickering on and off between frames 1 and 4 It is unclear which of the two molecules highlighted together on the right hand side of the ground truth frame are emitting in 2 and 4



(a) 50 mean photons      (b) 100 mean photons      (c) 150 mean photons      (d) 200 mean photons

**Figure 5.4: A figure showing the effect noise has on the simulation** This shows a simulation made at a frame size of 10000nm with 100 molecules with changing amount of noise. The shown frames were taken from a number of different simulations with varying amounts of noise present. It clearly can be seen that as the mean noise value increases through (a) to (d) from 50 to 200 mean photon count the effect becomes more pronounced. The non uniform nature of the noise is also demonstrated as there is a stippling effect across the images and not just a total white blur.



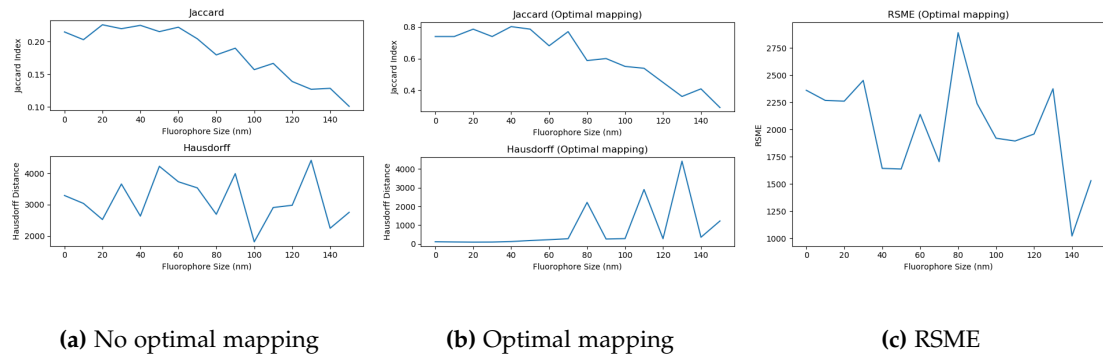
(a) 150nm      (b) 300nm      (c) 500nm

**Figure 5.5: A figure showing the effect changing the FWHM has on the appearance of the PSF** Parameters used are the same as 4.5 except from the fact that mean photon count for noise was set to 0. It can clearly be seen that increasing the FWHM from (a) to (c) increases the size of the PSF.

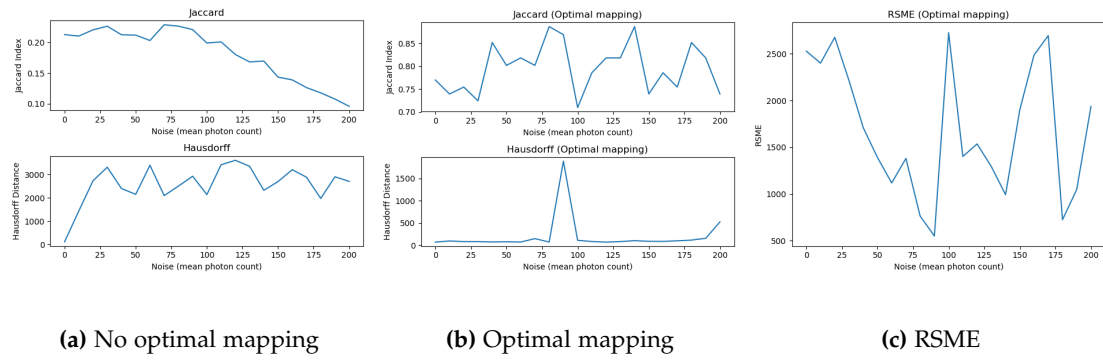


(a) STORM Frame      (b) Ground Truth      (c) Results table

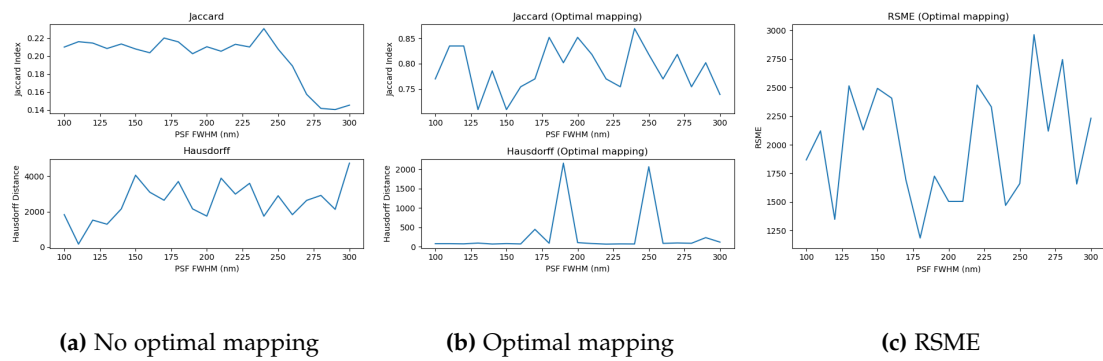
**Figure 5.6: The three outputs of STORMSim** Parameters used are the same as those shown in table 4.5 except from the fact that mean photon count for noise was set to 0. The three kinds of output from the system is demonstrated: (a) a frame from the STORM simulation (in reality a series of frames in a video will be generated), (b) the ground truth image and (c) a table of ground truth values.



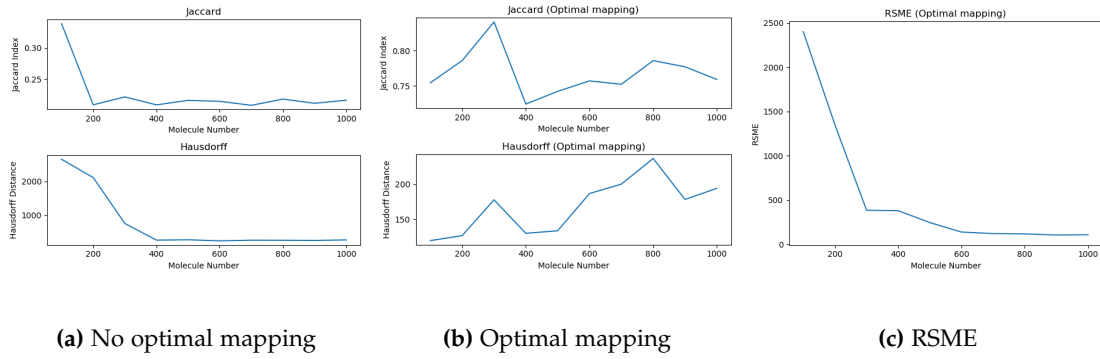
**Figure 5.7:** A figure describing the impact that changing the size of the fluorophore has on the MLE reconstruction technique in ThunderSTORM Parameters used are the same as those shown table 4.5 except from the fact that the fluorophore size has been changed between the separate simulations.



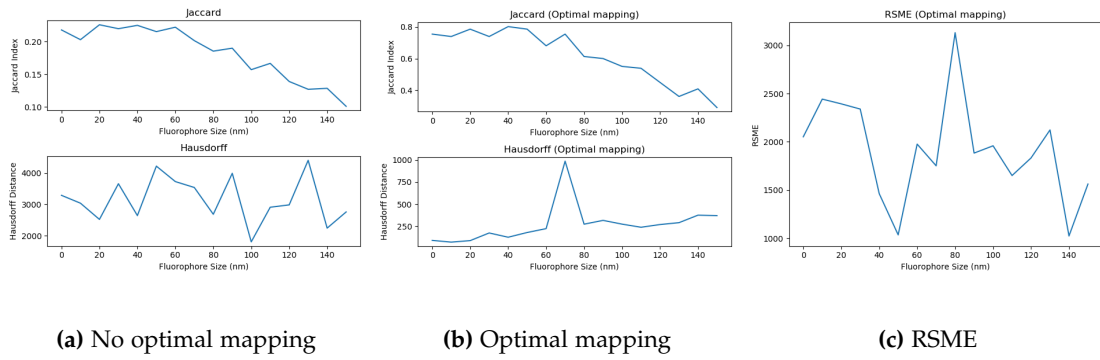
**Figure 5.8:** A figure describing the impact that changing the mean photon value of the noise has on the MLE reconstruction technique in ThunderSTORM Parameters used are the same as those shown in table 4.5 except from the fact that the mean noise value has been changed between the separate simulations.



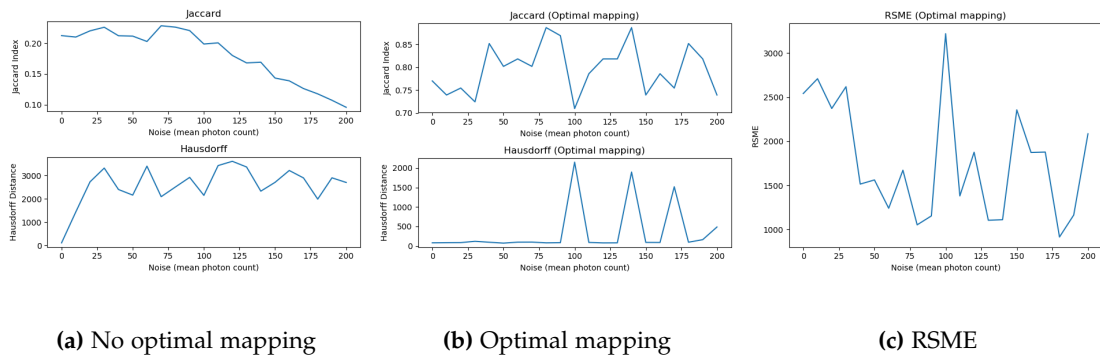
**Figure 5.9:** A figure describing the impact that changing the PSF FWHM has on the MLE reconstruction technique in ThunderSTORM Parameters used are the same as those shown in table 4.5 except from the fact that the PSF FWHM has been changed between the separate simulations. .



**Figure 5.10:** A figure describing the impact that changing the number of molecules has on the MLE reconstruction technique in ThunderSTORM Parameters used are the same as those shown in 4.5 except from the fact that the number of molecules has been changed between the separate simulations.

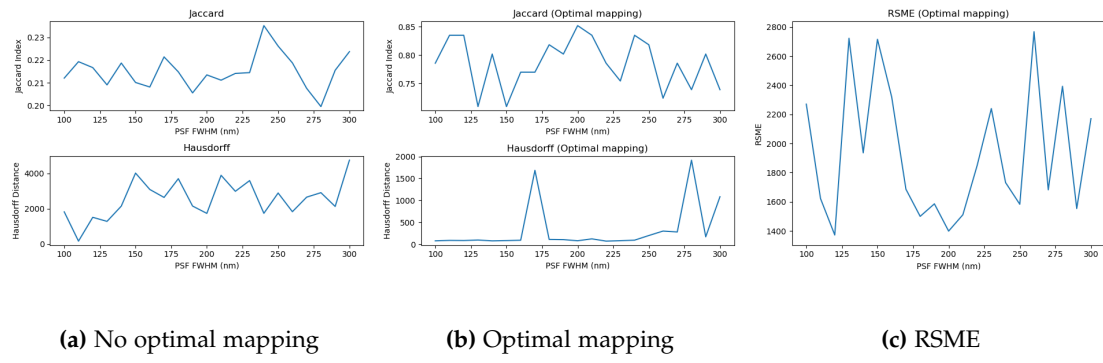


**Figure 5.11:** A figure describing the impact that changing the fluorophore size has on the least squared reconstruction technique in ThunderSTORM Parameters used are the same as 4.5 except from the fact that the fluorophore size has been changed between the separate simulations.

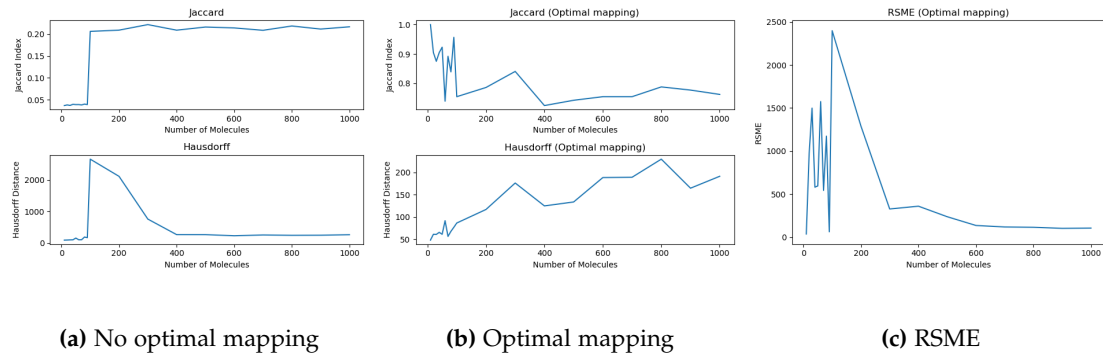


**Figure 5.12:** A figure describing the impact that changing the mean noise value has on the least squared reconstruction technique in ThunderSTORM Parameters used are the same as those shown in 4.5 except from the fact that the mean noise value has been changed between the separate simulations





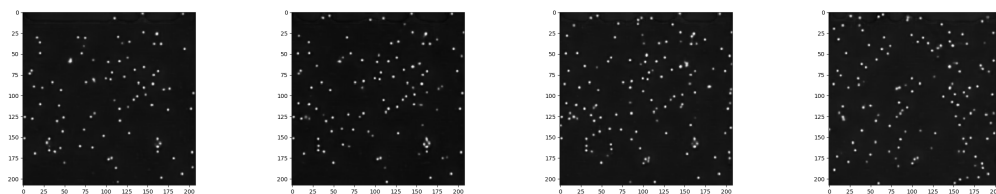
**Figure 5.13:** A figure describing the impact that changing the PSF FWHM has on the least squared reconstruction technique in ThunderSTORM. Parameters used are the same as those shown in 4.5 except from the fact that the PSF FWHM has been changed between the separate simulations.



**Figure 5.14:** A figure describing the impact that changing the number of molecules has on the least squared reconstruction technique in ThunderSTORM. Parameters used are the same as those shown in 4.5 except from the fact that the number of molecules has been changed between the separate simulations.

	MLE (average)	Least squares (average)	Total (average)
Jaccard	0.20	0.17	0.19
Hausdorff	2270.72	2183.89	2227.30
Jaccard (mapped)	0.74	0.76	0.75
Hausdorff (mapped)	370.90	268.91	319.90
RSME	1553.71	1568.42	1561.06

**Figure 5.15:** A table comparing the average values of the evaluation metrics between the MLE and least squared reconstruction methods



**Figure 5.16:** A demonstration of the results obtained from a auto encoder neural network approach

## Chapter 6

# Discussion

### 6.1 STORMSim

The simulator presented in this report has demonstrated its capabilities to effectively model the chemical and physical processes that occur within STORM. This has been proven by its ability to accurately model the areas of importance highlighted in chapter 3, namely the photoswitching behaviour and the PSF model. Both of the implemented versions of these systems were based in current research; in the case of the PSF model, research regarding the effectiveness of using a Gaussian model and in the case of the photoswitching model, research that suggests that a Markov model is the best way to simulate photoswitching (Patel et al. 2017; Stallinga and Rieger 2010).

Implementation of these methods was achieved successfully by building the software as a plug in on top of ImageJ using the SciJava API. Optimisation problems were solved, such as during image generation, which vastly increased the speed of the system. The system utilises a number of algorithms to generate its final data which are outlined in chapter 4.

The success of the simulator is proven by the images presented in the results section. It is clear that the simulator is able to produce visually accurate STORM data. This is shown through a number of examples: figure 5.3 which demonstrates the systems ability to model the PSF, figure 5.2 which demonstrates the system's ability to model photoswitching and figure 5.4 which demonstrates the systems ability to model noise. The number of different parameters that can be changed make this tool an effective way to teach newcomers about the technique and as a way to visualise the effect changing different parameters may have on real STORM data.

### 6.2 Reconstruction

This tool can also be used for more than just as a visual aid for people learning the technique. Because of its robustness in its design and its ability to output a table of ground truth locations it may be used in the future as a tool to test new reconstruction techniques. This robustness and ability to be used for testing reconstruction techniques was proven by its success at reconstruction using ThunderSTORM. Using MLE and least squares methods in ThunderSTORM on generated STORMSim data for reconstruction produced good results. This is demonstrated in the results section. Additionally, the ability for data created by STORMSim to be reconstructed successfully using established, widely utilised reconstruction techniques further proves the effectiveness of its ability to simulate STORM.

The ability for STORMSim data to be reconstructed within ThunderSTORM also enables testing of the limits of ThunderSTORM by generating sets of data where only one parameter is changed, this is a lot more difficult with real world data as its harder to consistently create data with a single, changed parameter due to the more complex nature of imaging systems. Although ThunderSTORM's reconstruction techniques were applicable to the data, they were

only measurably successful after an optimal mapping problem was solved between the ground truth values and the reconstruction values. This is demonstrated in table 5.15. Once this optimal mapping was found, the error as measured as the Jaccard distance and the Hausdorff distance was relatively low. The results suggest that after optimal mapping the ThunderSTORM reconstruction algorithms handled the changes in noise (figure 5.12 and figure 5.8) and PSF FWHM (figure 5.13 and figure 5.9) effectively.

It was changes to the size of the fluorophore (figure 5.11 and 5.7) and the number of molecules (figure 5.14 and figure 5.10) present that appeared to most effect the reconstruction algorithms. This is likely a follow on from the reason as to why without an optimal mapping approach the reconstruction techniques are not effective, it is because the estimators are very prone to overestimating the number of emitting sources present in the frames. ThunderSTORM likely overestimates the number of fluorophores because it does not handle the movement of the fluorophores between frames effectively, despite the fact that this movement is inherent within the technique due to fluorophore drift. This is modelled within STORMSim and, as a result, it is likely the main cause for error when STORMSim data is applied to Thunderstorm estimators; this follows from the fact that the largest observed error arose from increasing the number of molecules and increasing the size of the fluorophores; both of these parameters will lead to an increased amount of fluorophore movement within the frame either through directly increasing the amount of movement of the individual fluorophores or by increasing the number present.

Additionally, the results show that in many cases the evaluation metrics disagreed with one another as to whether error was increasing or decreasing. This highlights the importance of choosing the correct metric for evaluation and is especially important when building machine learning models. Using a wide variety of evaluation metrics and perhaps weighting them in some way may be a viable approach going forward.

### 6.3 Deep Learning

Due to this shortcoming there exists scope for better estimators. Deep learning is a particularly strong candidate to fill this gap. Recent papers as discussed in chapter 3.6 by Boyd et al. 2018, Nehme et al. 2018 and Li et al. 2018 have attempted to use deep learning techniques to tackle localisation problems, either specifically for STORM or for general SRM. Although the results from the deep learning in this report was not particularly effective, having high loss especially when using a convolutional network, there was still promise shown. This is especially demonstrated with the auto encoder model (see figure 5.16). This model appeared to learn the the feature representation of the Gaussian PSF function, this is demonstrated by figure 5.16 which demonstrates the network's output on an inputted frame that it had never seen before. The fact that it was able to reproduce, in some form, Gaussian noise that appears to be like the PSF suggests it has learnt some form of PSF model function. One would hope that the fact that the model is able to learn this function implies that it may be able to apply reconstruction through fitting of this PSF function. This suggests that STORMSim may be able to be used as a platform to generate large amounts of data quickly in order to train neural networks in the future. Whilst current neural network architectures have limited themselves to single frame inputs in general, the crux of STORM is that it is the multiple frames of stochastic blinking that enable more accurate localisation. As a result, future implementations of neural networks for the localisation task may be recurrent in nature.

### 6.4 Future Remarks

STORMSim has demonstrated its ability to produce accurate STORM data that has been used as a benchmark for comparing the ability of reconstruction estimators. This data may also be used to train neural networks or other such similar machine learning models in the future. Additionally, it may just also be used as a training and visualisation tool.

Additionally, given more time there is a large scope for improving STORMSim itself. For example there could be better modelling of imaging systems, this could be implemented alongside more refined and specific PSF models like those outlined in chapter 3. Additionally, it lacks features that are a part of similar software packages such as TestSTORM and ThunderSTORM that allow users to specify some kind of structure for the fluorophores to model, this being a more accurate representation of the way that STORM is employed on real world data (Ovesny et al. 2014; Sinkó et al. 2014). Despite this, STORMSim has achieved a large number of the goals that it set out to achieve.

# References

- Abbe, Ernst (1873). "Beiträge zur Theorie des Mikroskops und der mikroskopischen Wahrnehmung". In: *Archiv für mikroskopische Anatomie* 9.1, pp. 413–418.
- Boyd, Nicholas et al. (2018). "Deeploco: Fast 3D localization microscopy using neural networks". In: *BioRxiv*, p. 267096.
- Chapman, Sean, Karl J Oparka, and Alison G Roberts (2005). "New tools for in vivo fluorescence tagging". In: *Current opinion in plant biology* 8.6, pp. 565–573.
- Dubuisson, M-P and Anil K Jain (1994). "A modified Hausdorff distance for object matching". In: *Proceedings of 12th international conference on pattern recognition*. IEEE, pp. 566–568.
- Frahm, Lars (2016). "Stochastic modeling of photoswitchable fluorophores for quantitative super-resolution microscopy." PhD thesis. Georg-August-Universität Göttingen.
- Gardner, Andrew et al. (2014). "Measuring distance between unordered sets of different sizes". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 137–143.
- Gibson, Sarah Frisken and Frederick Lanni (1989). "Diffraction by a circular aperture as a model for three-dimensional optical microscopy". In: *JOSA A* 6.9, pp. 1357–1367.
- Goodfellow, Ian et al. (2014). "Generative adversarial nets". In: *Advances in neural information processing systems*, pp. 2672–2680.
- He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.
- Hell, Stefan W (2007). "Far-field optical nanoscopy". In: *science* 316.5828, pp. 1153–1158.
- (2008). "Microscopy and its focal switch". In: *Nature methods* 6.1, p. 24.
- Huang, Bo, Hazen Babcock, and Xiaowei Zhuang (2010). "Breaking the diffraction barrier: super-resolution imaging of cells". In: *Cell* 143.7, pp. 1047–1058.
- Huang, Bo, Mark Bates, and Xiaowei Zhuang (2009). "Super-resolution fluorescence microscopy". In: *Annual review of biochemistry* 78, pp. 993–1016.
- Huttenlocher, Daniel P., Gregory A. Klanderman, and William J Rucklidge (1993). "Comparing images using the Hausdorff distance". In: *IEEE Transactions on pattern analysis and machine intelligence* 15.9, pp. 850–863.
- Kirshner, Hagai et al. (2013). "3-D PSF fitting for fluorescence microscopy: implementation and localization application". In: *Journal of microscopy* 249.1, pp. 13–25.
- Leung, Bonnie O and Keng C Chou (2011). "Review of super-resolution fluorescence microscopy for biology". In: *Applied spectroscopy* 65.9, pp. 967–980.
- Li, Yu et al. (2018). "DLBI: Deep learning guided Bayesian inference for structure reconstruction of super-resolution fluorescence microscopy". In: *arXiv preprint arXiv:1805.07777*.
- Lichtman, Jeff W and Scott E Fraser (2001). "The neuronal naturalist: watching neurons in their native habitat". In: *nature neuroscience* 4, p. 1215.
- Lim, Bee et al. (2017). "Enhanced deep residual networks for single image super-resolution". In: *The IEEE conference on computer vision and pattern recognition (CVPR) workshops*. Vol. 1. 2, p. 4.
- MacDonald, Laura, Giulia Baldini, and Brian Storrie (2015). "Does super-resolution fluorescence microscopy obsolete previous microscopic approaches to protein co-localization?" In: *Membrane Trafficking*. Springer, pp. 255–275.

- MicroscopyU (2017). *Single-Molecule Super-Resolution Imaging*. URL: <https://www.microscopyu.com/techniques/super-resolution/single-molecule-super-resolution-imaging> (visited on 08/28/2018).
- Mortensen, Kim I et al. (2010). "Optimized localization analysis for single-molecule tracking and super-resolution microscopy". In: *Nature methods* 7.5, p. 377.
- Munkres, James (1957). "Algorithms for the assignment and transportation problems". In: *Journal of the society for industrial and applied mathematics* 5.1, pp. 32–38.
- Nehme, Elias et al. (2018). "Deep-STORM: super-resolution single-molecule microscopy by deep learning". In: *Optica* 5.4, pp. 458–464.
- Ovesny, Martin et al. (2014). "ThunderSTORM: a comprehensive ImageJ plug-in for PALM and STORM data analysis and super-resolution imaging". In: *Bioinformatics* 30.16, pp. 2389–2390.
- Patel, Lekha et al. (2017). "A hidden Markov model approach to characterizing the photo-switching behavior of fluorophores". In: *bioRxiv*, p. 223875.
- Quan, Tingwei et al. (2011). "High-density localization of active molecules using Structured Sparse Model and Bayesian Information Criterion". In: *Optics express* 19.18, pp. 16963–16974.
- Richards, B and E Wolf (1959). "Electromagnetic diffraction in optical systems, II. Structure of the image field in an aplanatic system". In: *Proc. R. Soc. Lond. A* 253.1274, pp. 358–379.
- Rust, Michael J, Mark Bates, and Xiaowei Zhuang (2006). "Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM)". In: *Nature methods* 3.10, p. 793.
- Saxena, Manish, Gangadhar Eluru, and Sai Siva Gorthi (2015). "Structured illumination microscopy". In: *Advances in Optics and Photonics* 7.2, pp. 241–275.
- Sciences, Royal Swedish Academy of (2014). *2014 Nobel Prize in Chemistry*. URL: <https://www.nobelprize.org/prizes/chemistry/2014/summary/> (visited on 08/31/2018).
- Sinkó, József et al. (2014). "TestSTORM: Simulator for optimizing sample labeling and image acquisition in localization based super-resolution microscopy". In: *Biomedical optics express* 5.3, pp. 778–787.
- Small, Alex and Shane Stahlheber (2014). "Fluorophore localization algorithms for super-resolution microscopy". In: *Nature methods* 11.3, p. 267.
- Stallinga, Sjoerd and Bernd Rieger (2010). "Accuracy of the Gaussian point spread function model in 2D localization microscopy". In: *Optics express* 18.24, pp. 24461–24476.
- Starr, Rebecca, Shane Stahlheber, and Alex Small (2012). "Fast maximum likelihood algorithm for localization of fluorescent molecules". In: *Optics letters* 37.3, pp. 413–415.
- Thompson, Russell E, Daniel R Larson, and Watt W Webb (2002). "Precise nanometer localization analysis for individual fluorescent probes". In: *Biophysical journal* 82.5, pp. 2775–2783.
- Willig, Katrin I et al. (2007). "STED microscopy with continuous wave beams". In: *Nature methods* 4.11, p. 915.
- Xu, Jun et al. (2016). "Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images". In: *IEEE transactions on medical imaging* 35.1, pp. 119–130.
- Yuste, Rafael (2005). "Fluorescence microscopy today". In: *Nature methods* 2.12, p. 902.

# Appendix A

## Running the code

The following steps are required to run the code:

1. Download the repository from <https://git-teaching.cs.bham.ac.uk/mod-msc-proj-2017/clw762>
2. Open the project using an IDE as a Maven project
3. Import dependencies
4. Run the main method in STORMSim
5. ImageJ should open in a separate window
6. Click on plugins and locate STORMSim
7. A UI that looks like 4.3 should appear

Alternatively, download the jar in the jar folder in the repository. Install the plugin into ImageJ by following the instructions [http://imagejdocu.tudor.lu/doku.php?id=howto:plugins:how\\_to\\_install\\_a\\_plugin](http://imagejdocu.tudor.lu/doku.php?id=howto:plugins:how_to_install_a_plugin).

## Appendix B

# Additional PSF models

### B.1 Gibson & Lanni PSF model

$$h(x, y, z) = \left| \frac{C}{z_d} \int_0^1 J_0 \left[ ka\rho \frac{(x^2 + y^2)^{0.5}}{z} \right] e^{jW(\rho)} \rho d\rho \right|^2$$
$$W(\rho) = k \left\{ n_s t_s \left[ 1 - \left( \frac{\text{NA}\rho}{n_s} \right)^2 \right]^{0.5} + n_g t_g \left[ 1 - \left( \frac{\text{NA}\rho}{n_g} \right)^2 \right]^{0.5} + n_i t_i \left[ 1 - \left( \frac{\text{NA} \cdot \rho}{n_i} \right)^2 \right]^{0.5} - \right. \\ \left. - n_g^* t_g^* \left[ 1 - \left( \frac{\text{NA}\rho}{n_g^*} \right)^2 \right]^{0.5} - n_i^* t_i^* \left[ 1 - \left( \frac{\text{NA} \cdot \rho}{n_i^*} \right)^2 \right]^{0.5} \right\}$$

**Figure B.1:** The Gibson & Lanni model of PSF. This takes into account immersion and cover slip as well as the structure of the microscope. Equation taken from Kirshner et al. 2013



## B.2 Richards & Wolf PSF model

$$\begin{aligned}
h(x, y, z) &= |I_0|^2 + 2|I_1|^2 + |I_2|^2 \\
I_0(x, y, z) &= \int_0^\alpha B_0(\theta, x, y, z) \left( t_s^{(1)} t_s^{(2)} + t_p^{(1)} t_p^{(2)} \frac{1}{n_s} \sqrt{n_s^2 - n_i^2 \sin^2 \theta} \right) d\theta \\
I_1(x, y, z) &= \int_0^\alpha B_1(\theta, x, y, z) \left( t_p^{(1)} t_p^{(2)} \frac{n_i}{n_s} \sin \theta \right) d\theta \\
I_2(x, y, z) &= \int_0^\alpha B_2(\theta, x, y, z) \left( t_s^{(1)} t_s^{(2)} + t_p^{(1)} t_p^{(2)} \frac{1}{n_s} \sqrt{n_s^2 - n_i^2 \sin^2 \theta} \right) d\theta \\
B_m(\theta, x, y, z) &= \sqrt{\cos \theta \sin \theta} J_m(k \sqrt{x^2 + y^2} n_i \sin \theta) e^{jW(\theta)} \\
W(\theta) &= k \left\{ t_s \sqrt{n_s^2 - n_i^2 \sin^2 \theta} + t_i \sqrt{n_i^2 - n_i^2 \sin^2 \theta} - t_i^* \sqrt{n_i^{*2} - n_i^2 \sin^2 \theta} + \right. \\
&\quad \left. + t_g \sqrt{n_g^2 - n_i^2 \sin^2 \theta} - t_g^* \sqrt{n_g^{*2} - n_i^2 \sin^2 \theta} \right\}
\end{aligned}$$

**Figure B.2:** The Richards & Wolf PSF model. This takes into account the structure of the microscope. Equation taken from Kirshner et al. 2013

### B.3 Parameters

Parameter	Description
NA	Numerical Aperture of the objective lens. This value is given by $NA = n_i \sin(\Theta)$ where $\Theta$ is one half of the angular aperture of the lens.
$n_s$	Refractive index of the sample layer.
$t_s$	Axial location of the point source within the sample layer. The corresponding input field in the plugin is "z".
$n_g$	Actual value of the refractive index of the coverslip. This value is assumed to be equal to $n_g^*$ and it is therefore not used by the plugin.
$n_g^*$	Nominal value of the refractive index of the coverslip. This value is assumed to be equal to $n_g$ and it is therefore not used by the plugin.
$n_i$	Actual value of the refractive index of the immersion layer (the paper by Gibson and Lanni uses the notations $n_{oil}$ ).
$n_i^*$	Nominal value of the refractive index of the immersion layer (the paper by Gibson and Lanni paper uses the notations $n_{oil}^*$ ). This value is assumed to be equal to $n_i$ and it is therefore not used by the plugin.
$t_g$	Actual value of the coverslip thickness. This value is assumed to be equal to $t_g^*$ and it is therefore not used by the plugin.
$t_g^*$	Nominal value of the coverslip thickness. This value is assumed to be equal to $t_g$ and it is therefore not used by the plugin.
$t_i$	Actual working distance between the objective and the coverslip (the Gibson and Lanni paper uses the notation $t_{oil}$ ).
$t_i^*$	Nominal working distance between the objective lens and the coverslip (the Gibson and Lanni paper uses the notation $t_{oil}^*$ ). This value is given by $t_i^* = t_i + \Delta$ where $\Delta$ is the stage displacement. When generating a z-stack, stage displacement values are determined by the "Axial resolution" and by the "slices" input parameters of the plugin. If the number of slices is odd, then the middle slice corresponds to $\Delta = 0$ .
$\lambda$	Wavelength of the light emitted by the point source.
$k$	Wavenumber in vacuum of the emitted light, $k = 2\pi/\lambda$ .
$x_d, y_d$	Lateral position for evaluating the PSF at the detector plane. The sampling points are $\{x_{d,n}, y_{d,m}\} = \{n\Delta, m\Delta\}_{n,m}$ where $\Delta$ is the value of the "Pixel size in object space" input parameter, $n = -(N-1)/2 \dots (N-1)/2$ and $m = -(M-1)/2 \dots (M-1)/2$ . $N$ and $M$ are the "width" and the "height" input parameters, respectively of the plugin. If $N$ and $M$ are odd, then the middle pixel is $(0,0)$ .
$z_d$	Axial distance between the detector and the tube lens. This value is assumed to be equal to the focal length of the tube lens, and for this reason the plugin substitutes $NA = a/z_d$ in $h(x, y, z)$ .
$W$	The Gibson and Lanni phase aberration.
$C$	A normalizing constant. The value $C/z_d$ that appear in $h(x, y, z)$ is determined by the "normalization" input parameter of the plugin.

**Figure B.3:** A table showing the parameters for equations B.1 and B.2. Table taken from Kirshner et al. 2013

## Appendix C

# Results Raw Data

### C.1 MLE

Size (nm)	Jaccard	Hausdorff	RSME	Jaccard (mapping)	Hausdorff (mapping)
0	0.21	3291.95	2361.92	0.74	112.46
10	0.20	3038.76	2268.64	0.74	101.30
20	0.23	2524.22	2261.88	0.79	93.06
30	0.22	3658.83	2452.35	0.74	96.84
40	0.22	2634.29	1643.67	0.80	123.03
50	0.22	4225.71	1637.98	0.79	178.63
60	0.22	3731.75	2140.01	0.68	222.52
70	0.20	3535.57	1705.43	0.77	274.69
80	0.18	2691.08	2891.25	0.59	2213.98
90	0.19	3990.47	2239.24	0.60	259.82
100	0.16	1809.89	1921.10	0.55	280.34
110	0.17	2907.88	1895.32	0.54	2901.90
120	0.14	2980.19	1959.59	0.45	276.09
130	0.13	4417.46	2375.68	0.36	4417.46
140	0.13	2245.44	1021.82	0.41	352.99
150	0.10	2753.55	1530.63	0.29	1222.91

**Figure C.1:** A table showing the MLE results for changing the size of the fluorophore

Mean photon count	Jaccard	Hausdorff	RSME	Jaccard (mapping)	Hausdorff (mapping)
0	0.21	117.99	2528.46	0.77	75.14
10	0.21	1440.91	2400.55	0.74	99.89
20	0.22	2740.18	2676.47	0.75	86.26
30	0.23	3320.28	2213.87	0.72	85.77
40	0.21	2404.78	1710.19	0.85	78.18
50	0.21	2157.50	1391.31	0.80	82.95
60	0.20	3403.72	1118.11	0.82	76.39
70	0.23	2102.57	1381.25	0.80	152.95
80	0.23	2510.17	763.58	0.89	76.61
90	0.22	2932.65	550.94	0.87	1890.33
100	0.20	2146.34	2725.83	0.71	114.96
110	0.20	3426.36	1401.98	0.79	87.84
120	0.18	3616.98	1536.04	0.82	74.56
130	0.17	3362.74	1293.87	0.82	86.85
140	0.17	2329.97	993.07	0.89	107.05
150	0.14	2704.62	1901.11	0.74	92.82
160	0.14	3214.02	2486.95	0.79	90.34
170	0.13	2890.72	2694.57	0.75	103.93
180	0.12	1980.95	724.69	0.85	119.93
190	0.11	2908.48	1048.58	0.82	160.27
200	0.10	2712.64	1936.66	0.74	529.21

Figure C.2: A table showing the MLE results for changing the mean noise value

Mol no	Jaccard	Hausdorff Distance	RSME	Jaccard (mapping)	Hausdorff (mapping)
100	0.34	2653.82	2402.19	0.75	119.36
200	0.21	2113.95	1349.40	0.79	126.50
300	0.22	754.77	384.31	0.84	177.59
400	0.21	260.08	378.81	0.72	129.76
500	0.22	270.63	244.93	0.74	133.43
600	0.21	238.38	137.37	0.76	186.62
700	0.21	257.42	119.99	0.75	200.03
800	0.22	255.73	116.50	0.79	236.73
900	0.21	250.64	104.02	0.78	178.12
1000	0.22	266.53	106.92	0.76	194.03

Figure C.3: A table showing the MLE results for changing number of molecules

FHWM (nm)	Jaccard	Hausdorff	RSME	Jaccard (mapping)	Hausdorff (mapping)
100	0.21	1828.80	1868.07	0.77	78.37
110	0.22	164.34	2121.48	0.83	78.54
120	0.21	1518.40	1348.08	0.83	74.10
130	0.21	1286.22	2515.13	0.71	94.27
140	0.21	2149.26	2129.87	0.79	69.59
150	0.21	4052.16	2493.16	0.71	80.30
160	0.20	3092.82	2407.57	0.75	71.74
170	0.22	2639.23	1692.10	0.77	447.16
180	0.22	3697.04	1184.62	0.85	89.72
180	0.22	3697.04	1184.62	0.85	89.72
190	0.20	2152.00	1725.50	0.80	2152.00
200	0.21	1743.96	1504.55	0.85	105.47
210	0.21	3884.30	1504.80	0.82	82.45
220	0.21	2986.37	2522.04	0.77	67.34
230	0.21	3595.84	2331.69	0.75	72.75
240	0.23	1738.41	1470.15	0.87	70.94
250	0.21	2892.25	1660.51	0.82	2059.15
260	0.19	1825.90	2962.33	0.77	86.44
270	0.16	2632.28	2119.58	0.82	96.93
280	0.14	2913.04	2745.02	0.75	89.22
290	0.14	2126.00	1657.02	0.80	234.62
300	0.15	4732.25	2231.93	0.74	118.86

**Figure C.4:** A table showing the MLE results for changing the size of the FHWM

## C.2 Least Squares

Mol no	Jaccard	Hausdorff	RSME	Jaccard (mapping)	Hausdorff (mapping)
10	0.04	87.92	38.39	1.00	47.93
20	0.04	90.81	976.31	0.90	61.01
30	0.04	95.14	1498.87	0.88	60.77
40	0.04	97.73	581.58	0.90	65.33
50	0.04	153.50	596.21	0.92	61.02
60	0.04	98.88	1574.62	0.74	91.49
70	0.04	99.17	544.20	0.89	56.25
80	0.04	185.48	1173.64	0.84	67.60
90	0.04	165.43	64.22	0.96	76.70
100	0.21	2655.02	2397.07	0.75	86.28
200	0.21	2110.01	1291.03	0.79	116.61
300	0.22	754.77	328.78	0.84	175.94
400	0.21	261.98	361.10	0.72	124.64
500	0.22	261.12	238.95	0.74	133.45
600	0.21	226.33	137.36	0.75	188.18
700	0.21	250.85	120.00	0.75	188.85
800	0.22	240.47	116.43	0.79	229.43
900	0.21	243.57	103.95	0.78	164.34
1000	0.22	258.90	106.92	0.76	191.05

Figure C.5: A table showing the least squared results for changing the number of molecules

Size (nm)	Jaccard	Hausdorff	RSME	Jaccard (mapping)	Hausdorff ( mapping)
0	0.22	3289.33	2052.00	0.75	89.55
10	0.20	3037.93	2442.58	0.74	70.08
20	0.23	2518.93	2393.46	0.79	87.03
30	0.22	3660.75	2339.37	0.74	173.68
40	0.22	2641.17	1462.60	0.80	125.48
50	0.22	4224.33	1034.82	0.79	178.91
60	0.22	3729.67	1975.91	0.68	222.84
70	0.20	3540.72	1751.02	0.75	986.84
80	0.19	2682.90	3131.07	0.61	273.23
90	0.19	3994.58	1882.93	0.60	316.26
100	0.16	1800.01	1958.48	0.55	273.76
110	0.17	2908.56	1650.06	0.54	238.29
120	0.14	2983.98	1832.43	0.45	269.19
130	0.13	4411.06	2122.97	0.36	290.22
140	0.13	2240.20	1023.16	0.41	375.32
150	0.10	2757.14	1561.97	0.29	369.86

Figure C.6: A table showing the least squared results for changing the size of the fluorophore

Photon count	Jaccard	Hausdorff	RSME	Jaccard (mapping)	Hausdorff (mapping)
0	0.21	112.98	2542.49	0.77	79.10
10	0.21	1437.18	2710.20	0.74	82.26
20	0.22	2728.39	2372.86	0.75	84.60
30	0.23	3318.32	2619.56	0.72	117.81
40	0.21	2399.53	1515.28	0.85	93.78
50	0.21	2159.96	1562.38	0.80	71.38
60	0.20	3400.17	1241.30	0.82	95.90
70	0.23	2092.23	1672.51	0.80	97.31
80	0.23	2508.15	1053.34	0.89	78.35
90	0.22	2919.09	1154.50	0.87	82.79
100	0.20	2149.48	3220.41	0.71	2149.48
110	0.20	3426.18	1381.32	0.79	89.03
120	0.18	3608.67	1876.45	0.82	76.92
130	0.17	3364.91	1105.15	0.82	78.45
140	0.17	2329.03	1111.30	0.89	1895.45
150	0.14	2706.58	2357.03	0.74	89.59
160	0.14	3215.26	1873.79	0.79	88.02
170	0.13	2895.84	1878.39	0.75	1517.65
180	0.12	1987.53	915.24	0.85	94.42
190	0.11	2900.70	1164.66	0.82	160.34
200	0.10	2701.68	2085.50	0.74	484.73

**Figure C.7:** A table showing the least squared results for changing the mean noise value

FWHM (nm)	Jaccard	Hausdorff	RSME	Jaccard (mapping)	Hausdorff (mapping)
100	0.21	1824.97	2270.14	0.79	76.16
110	0.22	171.58	1622.36	0.83	86.11
120	0.22	1517.46	1372.41	0.83	83.30
130	0.21	1287.95	2722.10	0.71	94.38
140	0.22	2150.11	1936.37	0.80	75.30
150	0.21	4018.12	2715.08	0.71	81.64
160	0.21	3091.86	2319.48	0.77	90.10
170	0.22	2641.82	1686.46	0.77	1685.50
180	0.21	3700.84	1500.32	0.82	108.15
190	0.21	2154.20	1586.21	0.80	103.77
200	0.21	1739.83	1399.26	0.85	77.81
210	0.21	3892.31	1511.30	0.83	121.35
220	0.21	2987.76	1848.04	0.79	69.69
230	0.21	3593.58	2239.87	0.75	79.13
240	0.24	1743.79	1732.27	0.83	91.41
250	0.23	2891.55	1583.38	0.82	197.21
260	0.22	1835.35	2767.19	0.72	299.11
270	0.21	2657.20	1682.70	0.79	278.36
280	0.20	2910.81	2392.52	0.74	1918.06
290	0.22	2131.49	1554.20	0.80	168.05
300	0.22	4749.33	2170.70	0.74	1082.66

**Figure C.8:** A table showing the least squared results for changing the PSF FWHM



### C.3 Averages

Parameter	Metric	MLE	LS	Total
Noise	Jaccard	0.181646359	0.1816012	0.18162378
	Hausdorff	2591.646302	2588.659861	2590.153082
	Jaccard (Mapped)	0.796372444	0.796372444	0.796372444
	Hausdorff (Mapped)	203.4407208	362.255203	282.8479619
	RSME	1689.43312	1781.602745	1735.517933
PSF	Jaccard	0.199170009	0.215007544	0.207088777
	Hausdorff	2606.722241	2556.757006	2581.739623
	Jaccard (Mapped)	0.792683651	0.785982118	0.789332885
	Hausdorff (Mapped)	291.3490479	327.0119658	309.1805069
	RSME	1971.809651	1933.922906	1952.866279
Mol Number	Jaccard	0.226322996	0.130519964	0.17842148
	Hausdorff	732.1969236	438.7952742	585.4960989
	Jaccard (Mapped)	0.767766364	0.827190041	0.797478203
	Hausdorff (Mapped)	168.2172428	115.0995827	141.6584127
	RSME	534.4449883	644.7177141	589.5813512
Fluoro Size	Jaccard	0.182367776	0.182743342	0.182555559
	Hausdorff	3152.314796	3151.329687	3151.822242
	Jaccard (Mapped)	0.614121815	0.615705042	0.614913428
	Hausdorff (Mapped)	820.5014293	271.283677	545.8925532
	RSME	2019.157342	1913.426935	1966.292138

**Figure C.9:** A table showing the average metric values for all the different tested parameters

## Appendix D

# Project Management

### D.1 Overview

The project was managed in iterative stages, these tended to occur weekly where key tasks were outlined through a meeting with Dr.Iain Styles and tasks from the previous week were reviewed. These tasks were then tackled each week. A weekly report was kept of this in the form of emails.

This management enabled the work to be broken down into sizeable chunks that could be evaluated and tackled weekly. If I were to repeat this process again I would use a similar process.

### D.2 Week 1, meeting on 14/06/2018:

Research

### D.3 Week 2, meeting on 22/06/2018:

Further research, proposal and working environment set up

### D.4 Week 3, meeting on 28/06/2018:

Modelling of the PSF of a single molecule

### D.5 Week 4, meeting on 03/07/2018:

Location generation algorithm and modelling of multiple PSFs

### D.6 Week 5, self directed:

Photoswitching modelling

### D.7 Week 6, 18/07/2018:

Improvement of photoswitching algorithm, modelling labelling efficiency and utilising ThunderSTORM reconstruction

**D.8 Week 7, 23/07/2018:**

Modelling intensity of fluorophores and noise. Starting to build evaluation techniques for reconstruction techniques.

**D.9 Week 8, self directed:**

UI building and evaluation metrics created

**D.10 Week 9, 07/08/2018:**

Experiments run on ThunderSTORM evaluating reconstruction

**D.11 Week 10, self directed:**

Code clean up and experiments

**D.12 Week 11, self directed:**

Building a neural network approach and experiments continued

**D.13 Week 12 onwards:**

Performing final code clean up and writing up report