Low-cost Educational Stochastic Optical Reconstruction Microscopy (eSTORM)

by

Ranbel F. Sun

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 2013

[SEPTEMBER 2013]

Author .....

Department of Electrical Engineering and Computer Science

August 30, 2013

Certified by .....

Edward S. Boyden, Associate Professor

Thesis Supervisor

August 30, 2013

Accepted by .....

Albert R. Meyer

Chairman, Masters of Engineering Thesis Committee

Low-cost Educational Stochastic Optical Reconstruction Microscopy (eSTORM)

by

Ranbel F. Sun

Submitted to the Department of Electrical Engineering and Computer Science on

August 30, 2013

In Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

Abstract

A stochastic optical reconstruction microscope was built and demonstrated for under $20,000, enabling hands-on learning of single-molecule localization concepts in teaching laboratories. This was accomplished by replacing the most expensive parts of $500,000 commercial instruments, namely the laser, camera, and objective, with lower cost alternatives. Since lower cost also comes with higher noise, we characterized the optical and noise characteristics of the microscope. A new sample protocol, consisting of microspheres labeled with streptavidin-Alexa 647 conjugates, was developed to test the system, compare the image quality of two reconstruction programs (QuickPALM and rapidSTORM), and evaluate trade-offs in camera selection. Finally, by imaging defined actin features in 3T3 cells, the instrument was estimated to have a sub-diffraction resolution between 70 -100 nm.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview of super-resolution microscopy

Microscopy is one of the most powerful tools in biological research, enabling key insights into cellular structure and function. In particular, fluorescence light microscopy has become the modality of choice for imaging cells due to its non-invasive nature, the development of selective labeling probes, and high contrast between signal and background [1]. However, as first derived by Abbe, optical microscopes are inherently limited in spatial resolution by the diffraction of light waves. The maximum resolution is approximately half the wavelength of the light used, or 250 nm laterally [2]. Electron microscopy can achieve atomic resolution since the wavelength of an electron is shorter than that of visible light [3]. Unfortunately, it requires low pressure and extensive sample preparation which can be problematic for imaging cells. Samples can also be damaged by the radiation from energetic electrons [4].

To break the diffraction barrier, research teams have developed several super-resolution (SR) techniques which increase the fluorescence microscopy resolution by an order of magnitude or more and bring subcellular processes into focus. One way to bypass the diffraction limit, proposed in 1928 [5] and demonstrated with visible light in 1984 [6, 7], is to place the aperture within one wavelength of light from the emitter to achieve sub-diffraction resolution before the light spatially diverges. Since the strict distance requirement of this technique, called near-field microscopy, limits its applica-

tions to surface studies, current research efforts in the field have focused on developing far-field techniques which can be categorized into two basic approaches. The first approach uses patterned illumination to spatially isolate fluorescent emissions within a diffraction-limited region. This class includes structured illumination microscopy (SIM) [8] and stimulated emission depletion microscopy (STED) [9]. The second category uses photoswitching fluorophores, i.e. molecules that can be switched between a fluorescent "on" and a non-fluorescent "off" state, to temporally isolate close light sources. In 2006, three groups independently demonstrated single-molecule localization microscopy and named it stochastic optical reconstruction microscopy (STORM) [10], photoactivated localization microscopy (PALM) [11], and fluorescence photoactivation localization microscopy (FPALM) [12].

1.2 Motivation

Commercial versions of SR microscopes are now available and can be expected to have a broad impact on biological research. SR microscopes have already "provided new insights into the organizations of proteins associated with plasma membranes and intracellular membrane organelles...[and] have also been used to investigate the molecular architecture of synapses." [13] The tremendous potential of SR microscopy prompted *Nature Methods* journal to select it as method of the year in 2008 [14]. The emergence of SR microscopy indicates that is important for students and scientists to become familiar with SR principles and practice. This motivates our idea of developing a classroom SR microscope to facilitate teaching the instrument's underlying principles, strengths/weaknesses, and image processing methods.

The original STORM developed by the Zhuang laboratory uses activator-reporter pairs of fluorophores switched by laser pulses of two alternating wavelengths. This technique was licensed by Nikon, which sells commercial instruments for between $500,000 and $750,000. The goal of this work is to demonstrate SR microscopy with a custom educational STORM setup (eSTORM) built for under $20,000. This is accomplished by building a system based on a variant of STORM called dSTORM,

or direct STORM, developed by Heilemann[15]. This method only requires one fluorophore and a constant laser pulse of a single wavelength, decreasing the hardware cost and complexity. Instead of trying to achieve the best resolution, eSTORM takes a new approach by using less expensive components while still achieving sub-diffraction resolution. In addition, it employs a modular construction style which permits students to assemble the microscope themselves to better understand its operation, and develop a robust and inexpensive sample protocol.

Chapter 2

Theory

2.1 Classical resolution limit

2.1.1 Point spread function

The resolution in conventional optical microscopy is limited by the diffraction of light as it passes through a finite aperature. Light from a point source, approximated by a single fluorescent molecule, forms a finite blurry focal spot known as a point spread function (PSF). The intensity distribution of this spot defines the PSF of the microscope and has a size determined by the wavelength of light (λ) and the numerical aperature of the objective (NA).

The theoretical PSF of a wide-field microscope is also known as an Airy disc. Its intensity *I(r)* at a distance r from the light source position in the image plane is given by [16]:

$$I(r) \propto \left(\frac{J_1(r)}{r}\right)^2 \tag{2.1}$$

where J_1 denotes the Bessel function of the first kind and first order. Since it is difficult to compute Bessel function parameters to fit an intensity profile [17], the small outer rings of the Airy pattern are often ignored and the central lobe of the PSF is approximated by a Gaussian:

$$I(r) \propto exp\left(\frac{-r^2}{2\sigma^2}\right) \tag{2.2}$$

where σ denotes the Gaussian RMS width in one dimension. Figure 2-1 compares the Besselian PSF (solid line) with its Gaussian approximation (dashed line). This approximation has been used and verified [19, 20] and enables practical image processing in STORM.

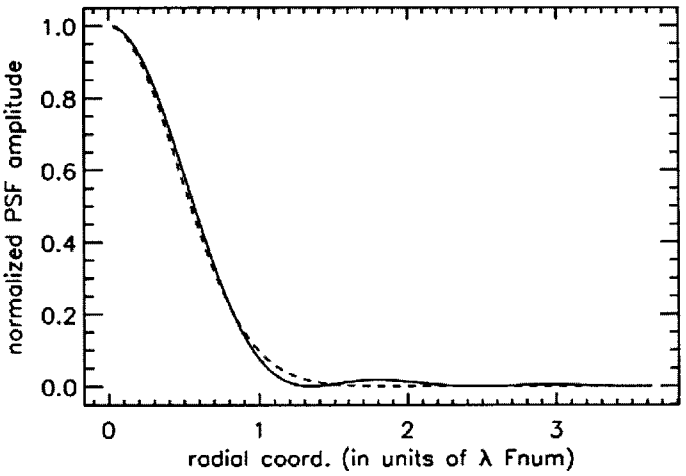


Figure 2-1: Cross section through an Airy diffraction pattern (solid line) and its Gaussian approximation (dashed line). Image credit: [18]

The final image of an extended object as viewed through the microscope is represented by the convolution of the PSF with the object [21]. Note that the blurring of the image represented by the theoretical PSF is the best case and does not take into account lens aberrations or optical inhomogeneity in the object.

2.1.2 Resolving power

The resolving power of a microscope relates to its ability to distinguish point emitters which are in close proximity. More specifically, the resolution is given by the distance between the center maximum of the PSF and its first minimum, known as the Rayleigh length *d(Rayleigh)*:

$$d(Rayleigh) = \frac{0.61\lambda}{NA} \tag{2.3}$$

Figure 2-2a shows how the two PSFs are well-separated when the two point sources are further than d(Rayleigh) apart. The overlap in Figure 2-2b is still considered resolvable, but the points become unresolvable when they are separated by less than d(Rayleigh) (Figure 2-2c).

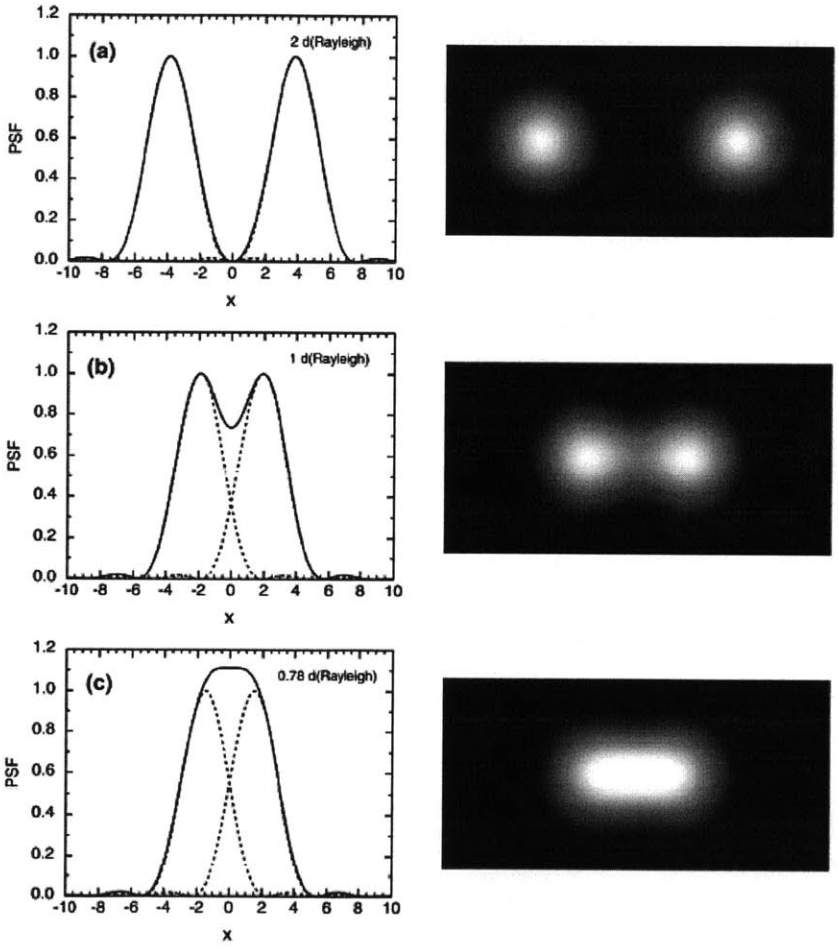


Figure 2-2: Resolving two point light sources with various separation distances. The left column shows a cross-section through y = 0. Dotted line indicates PSF for individual particles and solid lines indicates the sum of the two PSFs. The right column shows PSFs as a linear grayscale plot. (a) Separation by two Rayleigh lengths, (b) Separation by one Rayleigh length, and (c) Separation by 0.78 Rayleigh lengths. Image credit: [21]

2.2 Fluorescence microscopy

2.2.1 Fluorophores

Fluorescence is characterized by the emission of light that occurs within nanoseconds after the absorption of light of a shorter wavelength. Molecules that exhibit fluorescent properties are called fluorophores. The outer electron orbitals in the fluorophore molecule determine the excitation and emission wavelengths as well as its fluorescent efficiency, also known as quantum yield. The difference between the excitation and emission wavelengths, known as the Stokes Shift [1], allows for the excitation light to be filtered out, leaving only emitted light from the objects of interest. This work utilized the Alexa Fluor 647 fluorophore (Figure 2-3) synthesized by Molecular Probes because it was tested by the Zhuang lab and has several desirable qualities. The Alexa Fluor dyes are generally more photostable (providing more time for image capture) and less pH-sensitive than other common dyes (fluorescein, tetramethyrhodamine) of comparable excitation and emission spectra [24]. Cy, ATTO, and Alexa fluorophores have been shown to be able to switch reversibly between an emissive and a dark state without an activator dye, but Alexa exhibits a relatively high photon yield, a low duty cycle (allowing greater labeling density), and a large number of switching cycles [26]. These properties make it a preferable choice for eSTORM, which is expected to have lower detector efficiency and more noise than commercial instruments. The excitation peak resides at 650 nm and the emission peak resides at 665 nm (Figure 2-4).

2.2.2 Excitation and emission

A useful tool for understanding the excitation and emission process is the Jablonski diagram (Figure 2-5), in which horizontal lines illustrate different energy levels and arrows indicate transitions between the states [25]. The current theory behind fluorescence excitation and emission is summarized as follows: Normal, nonexcited molecules reside in the ground state. Upon absorption of a photon with sufficient energy ($E_{photon} = \frac{hc}{\lambda}$: where h is Plank's constant, c is the speed of light, and λ is

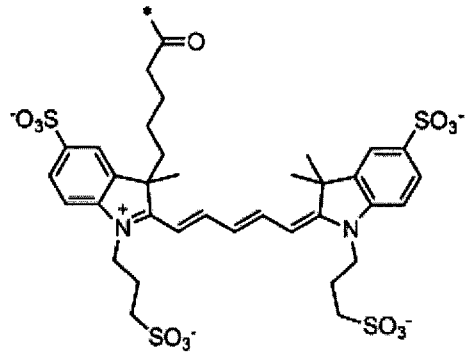


Figure 2-3: Alexa 647 chemical structure

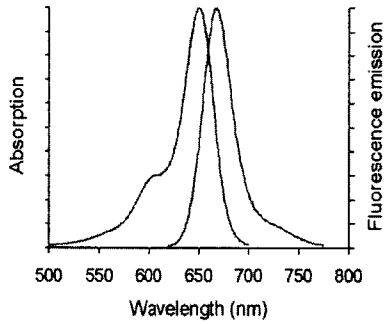


Figure 2-4: Alexa 647 emission and excitation spectra

the wavelength), an electron will quickly transition into a higher energy level. If the photon energy is greater than the energy required for fluorescence, the molecule will move into intermediate vibrational and rotational states.

Excited states are short-lived — the fluorophore will eventually return to the ground state via one of two main pathways. In the first case, the fluorophore relaxes directly into the ground state by emitting a photon with energy covering the difference in energy levels. This photon has a longer wavelength than the excitation photon, since some energy is inevitably lost via vibrational relaxation. Note that changing the excitation wavelength will not shift the emission spectrum. Also, photons are emitted in all directions, independent of the direction of the excitation photons, which is what allows epifluorescence microscopes to use the objective to simultaneously illuminate and image the sample.

In the other case, instead of fluorescing, the fluorophore can cross into a transient dark triplet state before stochastically returning to the ground state via phosphorescence. A molecule occupying the long-lived triplet state is unable to produce a fluorescence signal and can be considered temporarily deactivated. More importantly, triplet-state fluorophores also have the propensity to photobleach — that is, permanently lose their ability to fluoresce, resulting in a fading signal over time. One way bleaching can occur is when the triplet state reacts with oxygen to produce reactive oxygen species (ROS) which in turn can covalently alter the fluorescent molecule [27]. Since the spatial resolution of STORM is limited by the number of detected photons [28], photobleaching must be taken into account during instrument and protocol design.

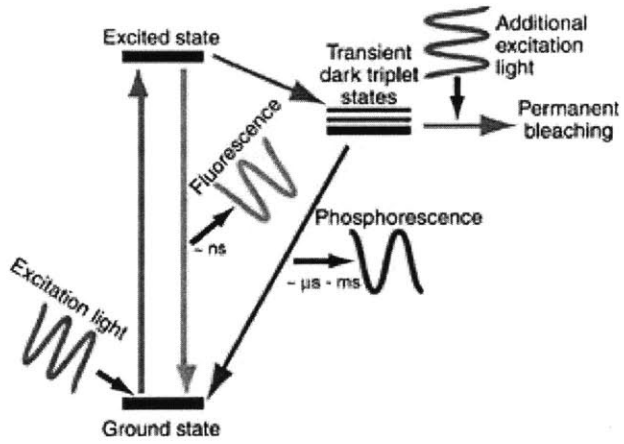


Figure 2-5: Jablonski diagram illustrating the energy states and transitions of a typical fluorophore. Image credit: [23]

2.3 STORM microscopy

2.3.1 Image formation

SR microscopy originiated from the idea that when point sources are separated by a distance less than the Rayleigh length, their PSFs will overlap and generate a single bright region where distinction of the individual sources is impossible (Figure 2-6). However, because the center of a point spread function can be localized down to the

nanometer scale [29], imaging individual molecules asynchronously can result in sub-diffraction limit resolution. Therefore, to obtain super-resolution, one can deactivate most of the fluorophores so that less than one molecule per diffraction-limited area can be in its "on state at a given time. Figure 2-7 illustrates the general principle behind photoswitching microscopy.

The sample is labeled with a photoswitching fluorophore at a density high enough to satisfy the Nyquist criteria. This means that the intermolecular spacing should not exceed twice the desired resolution [30]. The switching is controlled by a laser with low enough power to activate only a fraction of the fluorophores. The position of each individual fluorophore is determined by fitting a Gaussian distribution to its PSF. This process is repeated many times and the localizations are combined to create a single image. The main disadvantage of STORM over spatial confinement techniques like SIM and STED is the time it takes to acquire and process the image stack needed for producing a single super-resolution image.

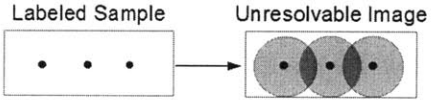


Figure 2-6: Conventional fluorescence microscopes cannot resolve fluorophores (black dots) spaced closer than the Rayleigh length because the PSFs (red circles) overlap.

2.3.2 Resolution

The resolution of a STORM microscope is described by the uncertainty in determining the position of a fluorophore from the captured images. Given an isolated fluorophore, the uncertainty in its localization is the uncertainty in fitting the center of the PSF [28]:

$$\Delta_{localization} \sim \frac{\Delta_{PSF}}{\sqrt{N}} \tag{2.4}$$

where $\Delta_{localization}$ is the localization precision, Δ_{PSF} describes the PSF width, and N is the number of detected photons. Many other potential factors can increase the

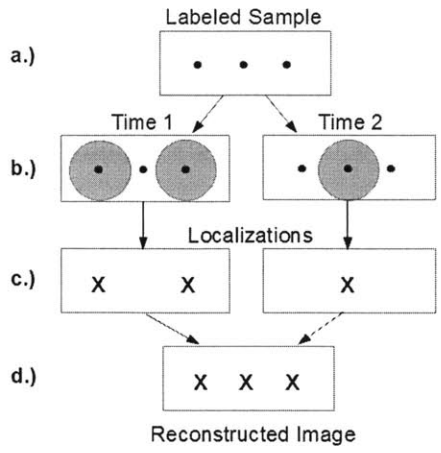


Figure 2-7: Principle of photoswitching microscopy: A sample is labeled with fluorophores

localization uncertainty: mechanical drift during image stack acquisition, noise, non-Gaussian PSFs [31], and nonuniformity in camera pixel quantum efficiency and gain [32].

Literature typically reports photoswitching microscopy resolution as the Full Width Half Maximum (FWHM) of the localization measurement distribution for a single fluorophore. An x-y resolution of 20 nm has been achieved with dSTORM [15] and 10 nm with STORM [33].

2.4 Estimating resolution

2.4.1 Photon budget

The feasibility of the eSTORM imaging system was examined by using a photon budget to estimate its achievable resolution, i.e. the FWHM of the localized fit. Our calculations were based off of Thompson's detailed analysis of single-molecule localization [20].

The energy flux Φ_{energy} at the sample was calculated by determining the area of the excitation laser beam at the sample A_{sample}, setting a laser power P_{laser} of 100 mW, and conservatively estimating a laser path power loss factor of 0.2 (Eq. 2.5).

The photon flux Φ_{photon} is the energy flux divided by the energy of an excitation photon E_{photon} (Eq. 2.6).

$$\Phi_{energy} = \frac{0.2 * P_{laser}}{A_{sample}} \tag{2.5}$$

$$\Phi_{photon} = \frac{\Phi_{energy}}{E_{photon}} \tag{2.6}$$

The photon emission rate from the sample Ph_{em_rate} depends on the dye's quantum yield Q_{dye} (defined as the ratio of absorbed to emitted photons) and its photon cross section σ (measure of the probability of absorption).

$$Ph_{em\_rate} = \frac{\Phi_{photon} Q_{dye}}{\sigma} \tag{2.7}$$

The number of photons detected $Ph_{detected}$ by the camera is the photon emission rate multiplied by the exposure time t_{exp} and the quantum efficiency of the camera Q_d. Photon losses were estimated as 70% due to detection angle, 20% due to filters and 50% due to the objective [20]. A lower NA objective was accounted for with an additional arbitrary 25% loss. Therefore,

$$Ph_{detected} = 0.3 * 0.8 * 0.5 * 0.75 * Ph_{em\_rate} t_{exp} Q_d \tag{2.8}$$

2.4.2 Camera noise

The process of representing the photons striking each pixel as a numerical value is subject to many sources of noise, resulting in unwanted variations in the final image. Shot noise, pixelation noise, readout noise, and dark current are the primary sources of noise in a CCD camera. By definition, photon shot noise is due to the fundamental quantum nature of light in which the probability of the photon arriving in a given time period is governed by a Poisson distribution. Shot noise increases with the square root of the signal. Pixelation noise arises from the uncertainty in where the photon arrived within a pixel. Readout noise is generated by the camera's output amplifier and is specified by the CCD manufacturer. Lastly, dark current is the build up of

thermally generated electrons over the timespan of one acquisition and is of particular concern in low light applications such as STORM imaging.

2.4.3 Localization uncertainty

Eq. 2.10 describes the localization uncertainty of a particle per imaging cycle $< (\Delta x)^2 >$, accounting for shot noise, pixelation noise, and background noise [20]. Here, s is the standard deviation of the microscope PSF and a is the pixel size in the sample plane. The total background noise b is taken as the sum of CCD camera readout noise N_r, CCD dark current N_d, and additional spurious noise N_o (Eq. 2.9). In Eq. 2.11, $FWHM$ is the Full Width at Half Maximum of the localized fit for n_{cycles} number of imaging cycles for that particular fluorophore.

$$b = N_r + (N_d + N_o)t_{exp} \tag{2.9}$$

$$< (\Delta x)^2 >= \frac{s^2 + \frac{a^2}{12}}{N} + \frac{8\pi s^4 b^2}{a^2 P h^2_{detected}} \tag{2.10}$$

$$FWHM = \frac{2\sqrt{2ln(2)} < (\Delta x)^2 >}{\sqrt{n_{cycles}}} \tag{2.11}$$

We considered the following three cameras because they fall in different price tiers and can be easily obtained: Allied Vision Technologies Manta 032-B, Hamamatsu Orca Flash 2.8, and Hamamatsu Orca ER. The Manta is the standard camera used in our lab, the Orca-Flash seems like a good price-quality compromise for this project, and the Orca-ER is a more ideal but expensive cooled camera.

Because dark current for the Manta and the Orca-Flash are unspecified in the camera documentation, we estimated them from the relative increase in operating temperature compared to the Orca-ER. From [34], the dark current roughly doubles for every 6°C increase in temperature.

We set the output laser power to 100 mW, that of a typical mid-tier laser. The output beam diameter was 1.1 mm, and the beam diameter at sample plane was 7.7

mm (given a beam expansion factor of 7x and a 100x objective). The dye parameters are those of Alexa 647.

Figure 2-8 presents our estimates for the FWHM of the localization fit for the three different cameras in different levels of spurious light N_o. Figure 2-9 keeps the noise parameter constant and varies the number of imaging cycles for the fitted particle. Despite having a higher readout noise than the Orca-Flash, the Orca-ER's low dark current and higher quantum yield give it the best estimated resolution of the three cameras. Our estimates show that given spurious background noise of approximately 1 or 2 orders of magnitude of readout noise, and given a long enough acquisition to capture multiple imaging cycles, it is theoretically possible to achieve sub-diffraction resolution with the proposed system components (detailed in the "Methods" chapter).

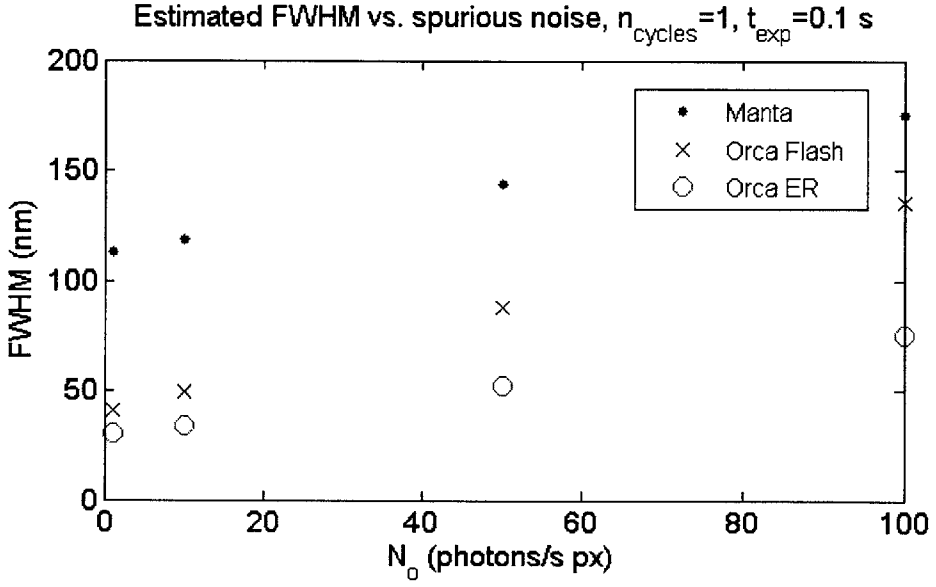


Figure 2-8: Estimated Full Width Half Maximum of the localization fit increases with increasing values of spurious noise.

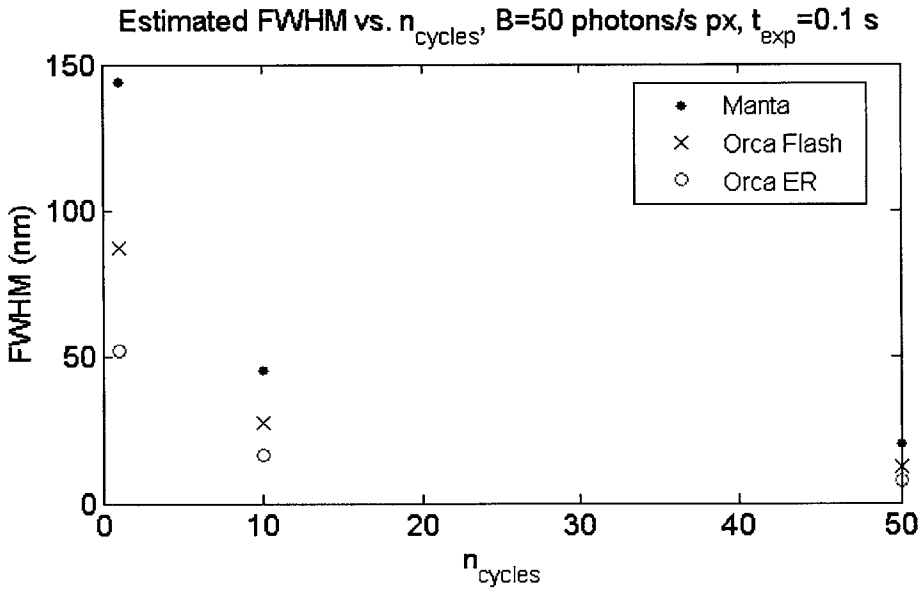


Figure 2-9: Estimated Full Width Half Maximum of the localization fit decreases for increasing numbers of imaging cycles.

Chapter 3

Methods

3.1 System Overview

The overall eSTORM imaging process is summarized in Figure 3-1. This is the same general idea as STORM/(F)PALM. First, a target structure is tagged with photoswitching fluorophores and immersed in an imaging buffer. Next, the sample is secured on the microscope stage, brought into focus under brightfield illumination, and then illuminated with 5 mW minimum of excitation light (determined experimentally). Up to 3000 images are captured using acquisition software to limit the size of the data file. Finally, the image stack is analyzed using reconstruction software.

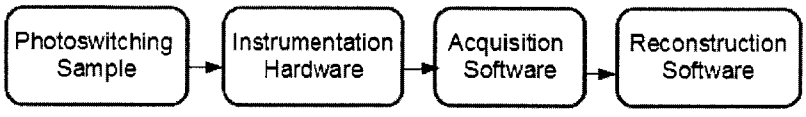


Figure 3-1: Overview of eSTORM imaging process

3.2 Samples

3.2.1 Coated microspheres

A simple, reliable, and easily identifiable sample was needed to test the eSTORM system and verify that the reconstruction software produced correct images of the sample. To this end, we designed a sample where 7.2 μm polysterene microspheres (Bangs Laboratories) were coated with an Alexa 647-streptavidin conjugate (Molecular Probes). The microsphere-to-protein ratio needed to achieve surface saturation is given by Eq. 3.1 [36].

$$S = \frac{6}{\rho D} C \tag{3.1}$$

where S is the amount of protein needed to coat 1 g of microspheres with a monolayer, ρ is the density of microspheres (1.05 g/cm^3 for polysterene), D is the diameter, and C is the capacity of microsphere surface for a given protein. we estimated C for streptavidin from the value for bovine serum albumin (3 mg/m^2) since they are of similar size and molecular weight.

The 7.2 μm microspheres were vortexed and then diluted to 1% solids using phosphate buffered saline (PBS). Since 3-10x the monolayer amount of protein is recommended for achieving full saturation, the microspheres were added to between 2.5-25 mg of Alexa 647-streptavidin (1x-10x the monolayer amount). The mixture was incubated for 2 hours in the dark at room temperature. Then, it was centrifuged for 6 minutes at 3000 g and the microsphere pellet was resuspended to 10 mg/mL. The centrifugation step was repeated and the pellet was resuspended in 0.5 M NaCl.

We found that adding the NaCl helped promote the adsorption of the microspheres to glass surfaces. We constructed a standard flow cell by adhering a No. 1.5 coverslip to a microscope slide with two pieces of 3M double-sided scotch tape spaced 5 mm apart (Figure 3-2). With the coverslip side down, the flow cell was filled with 10 μL of the microsphere in NaCl solution and left in a dark humidifier chamber for 1 hour to let the beads adhere to the coverslip. After incubating, the flow cell was washed

3x with PBS. Finally, the flow cell was filled with imaging buffer (Section 3.2.3) and the ends of the chamber were sealed with nail polish.

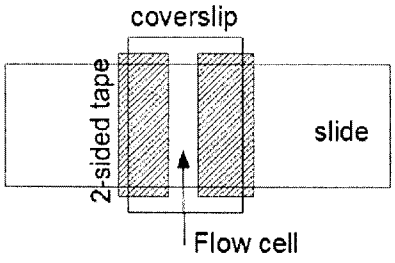


Figure 3-2: A flow cell constructed by placing two strips of double-sided scotch tape approximately 5 mm apart between a slide and a coverslip. The flow cell thickness is thus less than a few tens of microns. Before imaging, the flow cell is sealed with nail polish to prevent evaporation.

3.2.2 Actin in cells

During the development of super-resolution microscopy, subcellular structures such as microtubules, actin, and mitochondria were imaged as proof-of-concept [13]. These structures were chosen for their well-characterized morphology. We wanted a tried and true biological target so that more time could be spent on the instrument, and we selected 3T3 cells because they are known to be relatively robust and easy to culture. Therefore, the F-actin filaments of 3T3-Swiss albino mouse embryo fibroblast cells were stained with phalloidin-Alexa 647 to visualize the structure of the cytoskeleton. The cells were cultured at 37°C in 5% CO_2 in DMEM++. The day prior to actin staining, fibroblasts were plated on 35 mm glass-bottom cell culture dishes (MatTek) because they allow us to easily perform the staining protocol and add imaging buffer. We found that the ideal cell confluency should ideally reach about 60% before staining — too crowded and the filaments would not have space to stretch properly.

The staining protocol was derived from Invitrogen's phalloidin-Alexa 647 protocol. First we aspirated the culture medium and washed the cells twice with PBS (pre-warmed to 37°C to avoid temperature shock). Since the actin cytoskeleton is a dynamic structure in living cells, it was fixed by adding pre-warmed 3.7% formalde-

hyde for 15 minutes in the dark. After rinsing twice with PBS, the cell membrane was permeated by incubating in 0.5% Triton X-100 for 5 minutes. The cells were washed twice with PBS and a staining solution consisting of 200 μL PBS and 7μL Alexa 647 was added to the well. The dish was incubated overnight in the dark at 4°C. Finally, the cells were washed once with PBS and 400 μL of imaging buffer were added to the well (Section 3.2.3).

3.2.3 Imaging buffer

To promote photoswitching in a fluorophore population, eSTORM samples were imaged in a buffer which provides an alternative redox path for the triplet state fluorophores to quickly return to the ground state. If reactions with oxygen can be avoided, photobleaching can be prevented. The imaging buffer ("TN Buffer") is commonly used in molecular biology and consists of 50mM Tris pH 8.0, 10mM NaCl, and 10% glucose. For every 98 μL of TN Buffer, we add 1 μL of 143mM β-mercaptoethanol and 1 μL of glucose oxidase solution (50 mM Potassium Phosphate Buffer, 0.5 mg/mL glucose oxidase, and 40 μg/mL catalase) [32]. By adjusting the concentration of the reducing agent β-mercaptoethanol (BME), the rate of photoswitching can be controlled. The mixture of glucose, glucose oxidase, and catalase serves as an enzymatic oxygen scavenger system to reduce photobleaching.

Another oxygen-scavenging system to consider is based on pyranose oxidase instead of glucose oxidase. As shown in [35], pyranose oxidase can increase the photostability and lifetime of Alexa fluorophores in single-moleule experiments. The eSTORM samples currently use glucose oxidase because it is less expensive and well-tested in previous literature. However, pyranose oxidase may be worth testing in future work to reduce the noticeable photobleaching which occurs over long image acquisitions.

3.2.4 Actin in plants

Our idea for a low-cost eSTORM sample was to stain the actin filaments in onion cells. Plant cells are inexpensive, easy to obtain, and do not require upkeep like mammalian cells. We were unable to find literature describing photoswitching microscopy with plant cells. However, Olyslaegers [37] describes a protocol for staining actin in onion cells. The staining protocol underwent several iterations and modifications, but none of the samples resulted in observable photoswitching. We experimented with variations of the following protocol: we extracted a thin onion slice with forceps and incubated in actin buffer (100 mM PIPES, 10 mM EGTA, 5 mM $MgSO_4$, and 0.6 M Mannitol at pH 6.9) plus various concentrations of either glycerol or NP-40 detergent for 30 minutes in order to permeabilize the cell walls. Then, the onion slice was incubated in the dark at room temperature in actin buffer plus phalloidin-Alexa 647 for up to 90 minutes.

We observed that the samples permeabilized with glycerol showed no dye penetration into the cells. The samples made with NP-40 resulted in brief flashes of fluorescent filaments which bleached immediately and showed no photoswitching. The main drawback of NP-40 is that extraction is not limited to the cell membrane, resulting in the fast deterioration of the sample and image blurring. With glycerol, the intensity and quality of the actin stain should be much higher [37]. However, in our trials, NP-40 produced more promising stains than glycerol because we could actually see some fluorescent actin. Further experimentation with reagent concentrations and incubation times are needed before we can perform STORM imaging on onion actin filaments.

3.2.5 Control samples

Some control samples were needed to test the microscope and to assist in developing the sample protocols. We created a reference dye slide by pipetting 5 μL of stock concentration phalloidin-Alexa 647 onto a glass slide, letting the suspension medium evaporate, and attaching a coverslip with vacuum grease. We made a PSF mea-

surement slide using the same technique, using 175 nm deep red fluorescent beads (Invitrogen) diluted to a 1:50000 ratio of the stock concentration with PBS (ratio determined experimentally). We also created a photoswitching test slide consisting of pipetting streptavidin-Alexa 647 in NaCl into a flow cell (Figure 3-2, letting the particles passively adsorb onto the coverslip for at least 2 hours, washing away unbound particles with PBS, and adding imaging buffer (Section 3.2.3).

3.2.6 Other samples

Microtubules

Cytoskeleton targets are promising potential samples because they have a known size, shape, and distribution. Microtubules are cylindrically shaped with an outer diameter of about 25 nm and have been studied extensively using single-molecule fluorescence microscopy [40, 41, 38, 39]. Unlike actin in mammalian cells, microtubules do not require live cell culture, and their lengths and labeling ratio can be adjusted via the polymerization protocol. Gell [40] recommends binding the microtubules to a silanized goverglass surface via a spacer protein that attaches nonspecifically to the glass but specifically to the microtubule . The spacer protein holds the microtubule away from the surface, reducing unwanted surface interactions. For simplicity, we attempted to polymerize the microtubules, mount them on regular slides, and see if we could observe photoswitching. GTP is added to polymerize the mixture of Alexa 647-labeled tubulin and unlabeled tubulin. Then, we added taxol to stabilize the microtubules, in increasing concentrations spaced by 15 minute incubation periods at 37°C (see appendix for full protocol).

Rather than observing photoswitching microtubules in the samples, we mostly saw a dense field of switching fluorophores. We observed very sparse, non-switching microtubule strands in a few of the samples. This indicates that the tubulin did not polymerize reliably, possibly due to its temperature sensitivity. This protocol was put on hold in favor of optimizing actin and microsphere protocols.

DNA

Another sample we considered was DNA with fluorescent end modifications, a sample commonly used for demonstrating STORM [42, 26]. In short, complementary oligos are biotinylated on one end and labeled with Alexa 647 on the other end. After purification, the complementary strands are annealed to form biotinylated dsDNA. Biotinylated bovine serum albumin (b-BSA)is added to etched coverslips to prevent nonspecific binding. Next, the coverslip is coated with a layer of streptavidin. Since streptavidin has a high affinity for biotin, the biotinylated dsDNA is then immobilized on the streptavidin surface. The number of base pairs can be specified when ordering the oligos, therefore allowing one to specify the spacing between fluorophores. However, because the custom oligos are relatively expensive and the protocol more complex than actin staining, DNA was deemed a less than ideal sample for eSTORM.

3.3 Instrument

The first iteration of the eSTORM instrument was built by lab members Drago Andres Guggiana-Nilo and Brian Ross. In addition to documenting the current instrument's configuration, we shall note the major changes between the initial and the present design. For a summary of the most recent setup, see the optics block diagram in Figure 3-3 and the hardware photos in Figure 3-4.

3.3.1 Optomechanics

The microscope frame is constructed using readily accessible 30 mm and 60 mm cage system parts from Thorlabs. The cage system was originally supported by optical posts but was later lowered with cage plate mounting bases to increase stability. As seen in Figure 3-4, support is provided near the alignment mirrors, next to the dichroic cage cube, at the camera, and in between the spatial filter and the 200 mm condenser.

The Nikon STORM instrument uses an expensive, integrated motorized stage

with encoders. To reduce costs, we opted for a manually controlled stage. Among the sample stages available in our lab, we selected the 3-axis Thorlabs NanoMax-TS because it supposedly offers more long-term stability compared to the other stages. However, the sample holder base we had available was too wide for the MatTek dish cell samples. We solved this problem by adding a washer adapter that attached to the stage using the slide holder spring clips. For taking PSF images, the z-axis stage drive was replaced by a stepper motor actuator. We drove the stepper motor by interfacing a stepper motor driver carrier from Pololu to a NI-DAQ box.

Finally, the instrument was protected from light an enclosure made from cutting pieces of black foam and taping them to Thorlabs anodized beams.

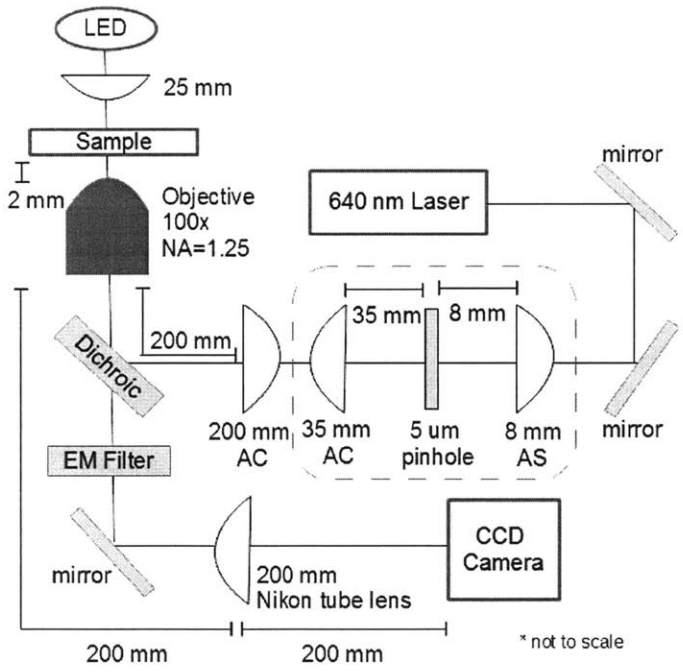
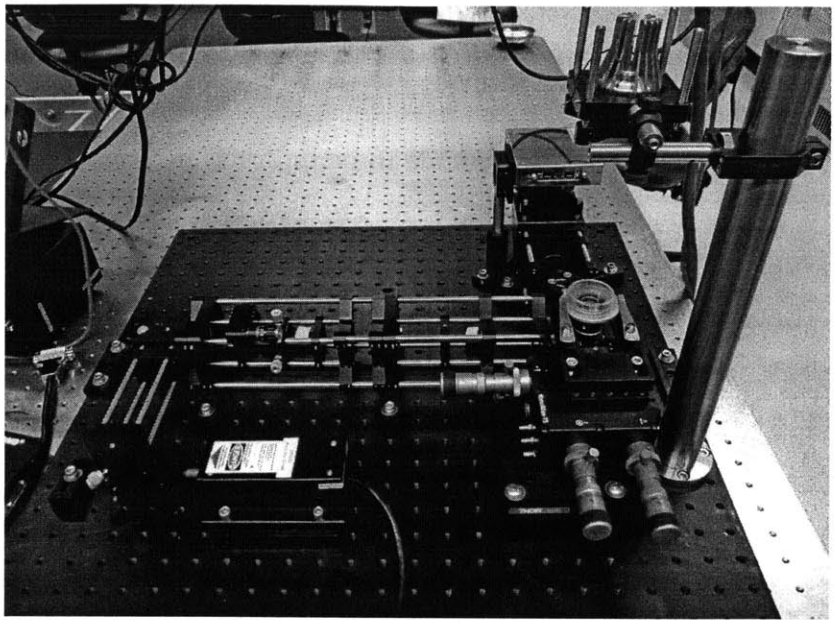


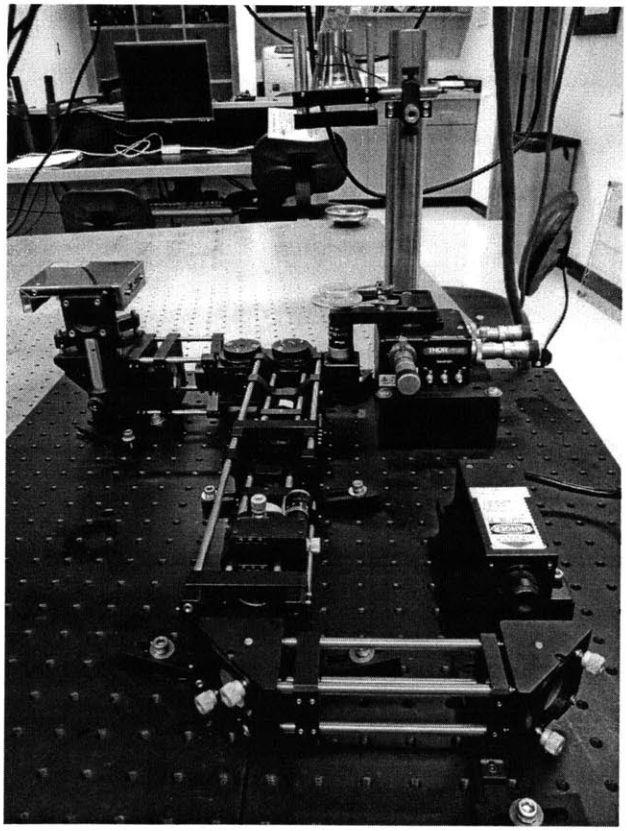
Figure 3-3: Block diagram of eSTORM optical design: The components in the dotted box comprise the spatial filter, AC = achromatic, AS = aspherical

3.3.2 Transillumination

To image in bright-field, a 25 mm plano convex lens condenses the light from a near infrared LED (@ 3 V, 400 mA) from above the sample.



(a)



(b)

Figure 3-4: a.) Top view of eSTORM hardware b.) Side view of eSTORM hardware

3.3.3 Epifluorescence illumination

The first version of eSTORM used a 5 mW 632.8 nm Melles-Griot laser borrowed from another lab. Because it was impossible to perform STORM imaging on a large sample area with this limited laser power, we changed to a 150 mW 640 nm laser (OEM Diode System). This mid-tier laser was selected because it was seen as a good price/quality compromise. The upgrade allows more flexibility in the optical design, allowing us to expand and filter the beam while still providing enough power to switch the fluorophores. The original setup incorporated a 7x Galilean telescope to expand the 1 mm laser beam to cover the FOV for regular fluorescent imaging. The OEM laser had a beam diameter of 3 mm, so we reduced the beam expander ratio to 3x (100 mm plano convex and -30 mm plano concave). However, the beam was nonsymmetric in shape and had noticeable intensity variations. This variation was most likely occurring because of optical scattering due to air particles and lens imperfections. We addressed this problem by adding a spatial filter and replacing the spherical lenses with aspheric/achromatic lenses.

Spatial filters improve the uniformity of a laser beam's intensity profile, which subsequently improves resolution. The intensity variations are removed by observing that upon focusing the beam, the interference/noise will remain defocused in an area surrounding the focused laser output. Thus, by placing a pinhole at the condenser focal point, the laser beam will pass through while the interference will be blocked. The optimal pinhole diameter is given by [43]:

$$P = \frac{8}{\pi}\frac{\lambda F}{D} \tag{3.2}$$

P is the pinhole diameter, λ is the wavelength, F is the focal length of the condenser, and D is the beam diameter. We opted for a 5 μm pinhole, an 8 mm aspherical condenser, and a 35 mm beam expansion lens based on what components were manufactured by Thorlabs. Manually aligning the spatial filter was not an easy task. The pinhole and 8 mm lens were mounted on a Thorlabs XY-translator and Z-translator respectively. We sought to maximize the power measured at the filter output - the

best alignment we managed yielded 30% of the beam power at the filter output. After losses throughout other parts of the optical path, the power at the sample plane was measured to be approximately 20% of the laser source output.

After passing through the spatial filter, a 200 mm aspheric lens focused the laser onto the back focal plane of the objective via a dichroic mirror (Chroma ZT647rdc). We used a 100x, 1.25 NA oil immersion objective from Nikon instead of expensive 1.4 NA or greater objectives which may also come with built-in correction for aberrations and flatness of field.. A high numerical aperature objective would produce a smaller PSF and thus better resolution.

The extra dichroic cage cube visible in Figure 3-4 was installed to potentially accommodate a UV laser to help control fluorophore photoswitching dynamics.

3.3.4 Camera path

The emitted light from the sample passes through the objective and is partially filtered by the dichroic, has a cutoff of 661 nm. An emission (EM) filter (Chroma ET700/75m) eliminates the rest of the extraneous wavelengths. The dichroic and EM filter cutoffs were chosen to accommodate the Alexa 647 spectra. Originally, a 200 mm plano convex lens focused the light onto the CCD. We replaced this with a specialized 200 mm Nikon tube lens (Edmund) to reduce aberrations. A custom tube lens mount was machined in order to accommodate its M38 x 0.5 threads. We acquired images with either a Hamamatsu Orca Flash 2.8 camera or a Hamamatsu Orca ER camera, which were two accessible cameras at different price/quality tiers.

3.4 Imaging Software

3.4.1 Acquisition

We capture image sequences using Hamamatsu's HCImage Live version 3.0.2.1 software, which came bundled with the Orca Flash 2.8 camera. The gain, exposure times, and total number of images varied with each acquisition. We limited the total

number of images to $\leq$3000 so that the data files were kept under 20 GB. In order to speed up the reconstruction process, we exported the image stack to multi-page tiffs approximately 200 x 200 pixels in size before running them through localization analysis.

3.4.2 Analysis

There are a number of open-source software packages available for converting raw data into a reconstructed image. In 2011, the original eSTORM processed the images using QuickPALM 1.1, an open source ImageJ plugin for particle detection and compiling STORM/(F)PALM images. The other ImageJ reconstruction plugin, Octane 1.2, has less documentation and does not show the reconstruction image during analysis. In 2012, a paper was published in Nature Methods describing rapidSTORM, a stand-alone open source Java program which has been able to reconstruct STORM images in real-time [53]. We tested out rapidSTORM 2.21, the most recent stable version.

3.5 Simulator

3.5.1 Overview

Some free, general simulator packages exist, such as the FluoroSim module in Microscope Simulator 1.3.1 [54]. However, they do not model photoswitching behavior, so a more applicable program is needed for STORM. Therefore, early in the project, we wrote a MATLAB simulation with three goals in mind: (1) To help understand the effect of key parameters in SR microscopy, (2) To generate image stacks to test reconstruction software, and (3) To help evaluate performance/cost tradeoffs by simulating the expected image produced in different eSTORM setups.

A fluorescent microscope can be modeled as a linear system where an image I is the convolution of the true fluorescent signal S with the PSF of the instrument. Noise N can come from various sources, including the CCD camera electronics and background photons in the system, and it is added to the convolved image (Eq 3.3).

Figure 3-5 summarizes the main features of the simulator, Figure 3-6a is an example of a single frame produced by the simulator, and Figure 3-6b shows a 100-image stack reconstructed using QuickPALM.

$$I = S \otimes PSF + N \tag{3.3}$$

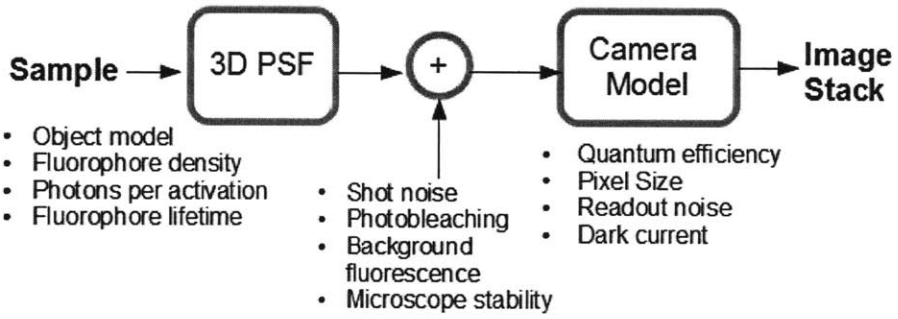


Figure 3-5: Simulator overview

3.5.2 Sample model

First, a sample's shape is modeled by specifying dimensional bounds on a 3D grid in which the sample resides. For ease of implementation, we represented actin filaments by rectangular bars spaced a certain distance apart. We set the fluorophore locations by setting their density and randomly assigning 3D Cartesian coordinates (with up to 2 decimal points precision) which fall within the bounding dimensions. When making real life samples, one can control the concentration of fluorophores but not exactly where they bind. Each modeled fluorophore also has an on/off state and a counter for the number of cycles remaining before photobleaching. The following fluorophore parameters are modeled: survival fraction after 400 seconds, number of photons emitted per switching event, and lifetime (switching cycles expected over 1000 seconds). The parameters for Alexa 647 and other photoswitchable dyes were experimentally determined by Dempsey [26].

We model photobleaching effects by randomly deleting a set of fluorophores at each imaging cycle, this set defined by the survival fraction which roughly decreases linearly

over time. Each remaining fluorophore is switched to the on state with probability given by the fraction of fluorophores expected to be on during the exposure time window.

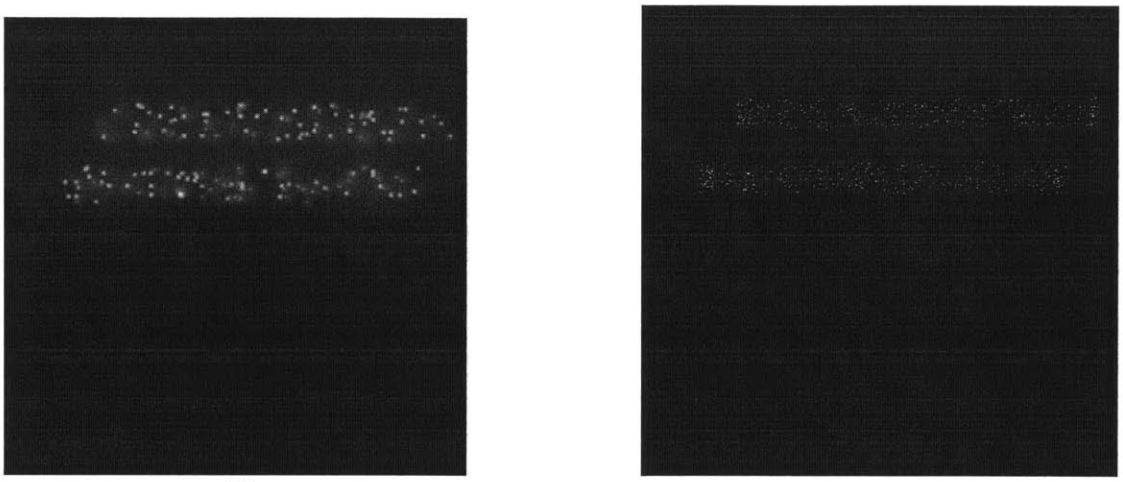
3.5.3 Instrument model

Since fluorophore locations have a finer distance grid than the camera pixel grid, we mapped the turned-on fluorophores onto an empty pixel grid with intensity values based on the distance from the pixel. For example, if a fluorophore falls at the intersection of 4 pixels, its photons will be split evenly between the 4 pixels. We designed a triangle filter to accomplished this objective and applied it to the fluorophore location 3D array. Finally, we binned the filtered fluorophore array into the pixel intensity array.

The 3D PSF of the microscope is an image stack generated by the PSF Generator ImageJ plugin [44], which generates various 3D models of a microscope PSF incuding the Gaussian model, the simulated defocus, the scalar -based Born & Wolf model [45], the scalar-based Gibson & Lanni model [46], and the vectorial-based Richards & Wolf model [47]. We imported the PSF stack into MATLAB and convolved it with the pixel intensity array. Since 3D convolution is computationally intensive, the operation is performed in C++ using an algorithm from Numerical Recipes in C [48].

3.5.4 Noise model

We model microscope instability by adding the same location shift for all fluorophores. Since shot noise and the camera's dark current and read noise depend on the random arrival of discrete photons and electrons, we draw the numbers from different Poisson random distributions and add them to the final output of the simulator. Since shot noise scales with the square root of the intensity, the shot noise added to a particular pixel is drawn from a Poisson distribution with $\lambda = \sqrt{pixel\ intensity}$. Dark current/thermal noise is known to increase linearly with exposure time, so λ is proportional to the exposure time. The readout noise is taken from a distribution where λ scales



(a) (b)

Figure 3-6: a.) Example image from simulator output stack b.) Stack reconstructed with QuickPALM

with the pixel intensity range.

Chapter 4

Results and Discussion

4.1 Basic Characterization

4.1.1 Field of view

We verified the Field of View (FOV) in the sample plane by imaging a 600 line pairs/mm sample in brightfield, giving us a FOV of 69.12 μm x 51.84 μm. This is consistent with the CCD camera's pixel count and the expected pixel size in the sample plane.

4.1.2 Flat field

We took a fluorescent image of a reference dye slide to check the beam profile and see if there are any pixel-to-pixel variations in the detector(Figure 4-1). The beam diameter was about half of what we expect given the laser diameter and the theoretical beam expansion factor of 4.375x. The spatial filter was most likely not aligned perfectly and cut off the outer edges of the beam. However, the beam profile was definitely more uniform and symmetrical than the profile without the spatial filter. We focused on taking STORM images in the center of the FOV.

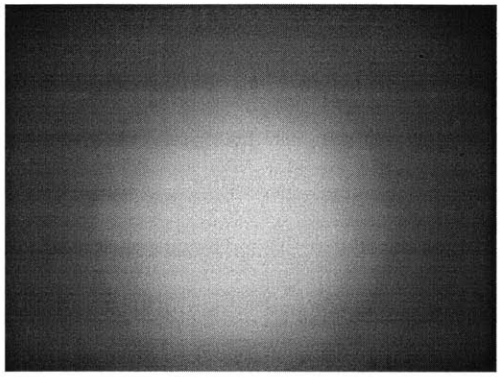
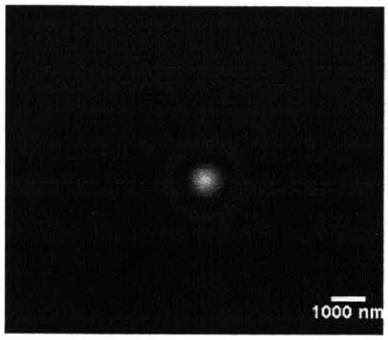


Figure 4-1: Image of reference dye slide, Gain=0 and Exposure = 1 sec

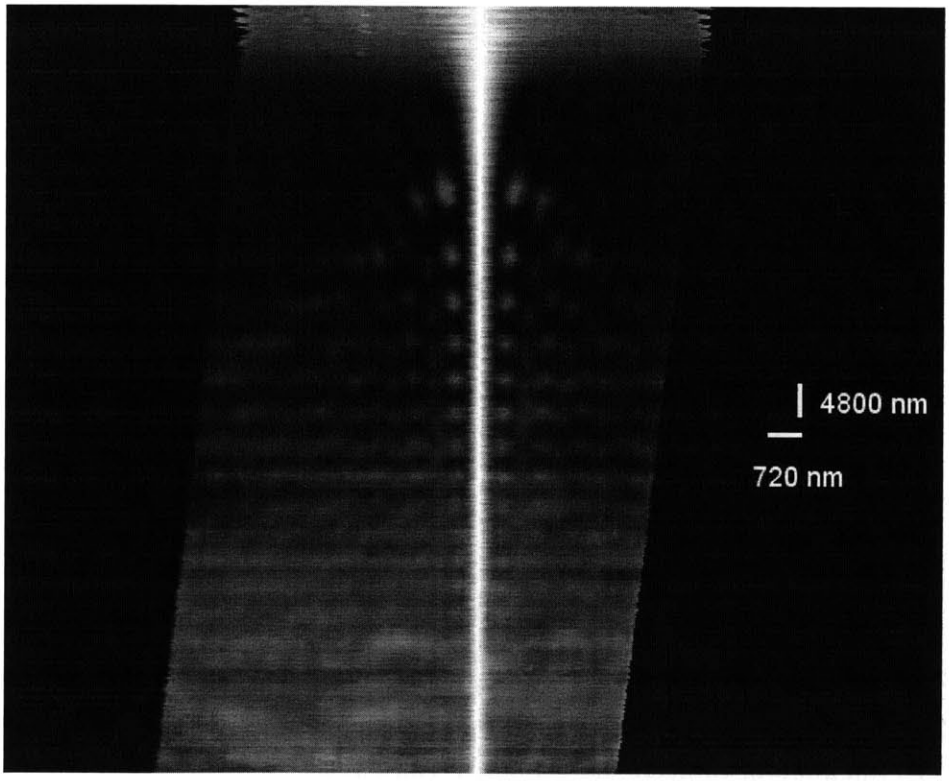
4.1.3 Point spread function

We measured the microscope's 3D PSF by taking an image stack of sub-diffraction fluorescent beads. Based on the capabilities of our stage axis actuator, we imaged at 500 distance steps along the z-axis, spaced 240 nm apart, with the first image corresponding to the furthest distance from the objective. The airy disk pattern is clearly visible near the focal plane, as shown in Figure 4-2a. Since the sample might not have been fixed perfectly in the x-y plane, and also because of stage drift, the bead centroid appeared to change x-y position as we scrolled through the stack. For display purposes, we wanted to align the PSF centroids of every stack. We tried readily available ImageJ plugins, but none of them worked to our satisfaction. Therefore, we designed an algorithm to register the image by low-pass filtering each image, finding the maximum intensity coordinates, and comparing this location to the initial centroid position in the stack. The result of this alignment is shown in Figure 4-2b.

One of the main utilities of measuring the PSF is finding the FWHM of the diffraction-limited spot. The theoretical Rayleigh resolution of the eSTORM instrument is 320 nm. To measure the actual resolution, the focal plane PSF is fitted with a 2D Gaussian (Eq. 4.1). The calculated best fit parameters generate the Gaussian function plotted in Figure 4-3b. The FWHM can be estimated from the standard deviation using Eq. 4.2, resulting in a FWHM of 11.75 pixels in both the x and y dimensions, or 420 nm given the Orca Flash pixel size. The symmetry in the PSF



(a)



(b)

Figure 4-2: a.) Slice of 3D PSF in x-y plane b.)Registered PSF in x-z plane

speaks to the quality of the objective lens as well as the quality of the microsphere sample slide. The measured FWHM is ~30% greater than the theoretical because of lens aberrations and sample non-uniformities. For a high-resolution optical system with a high-quality objective lens, the PSF FWHM from a sub-resolution fluorescent microsphere should be within 10%−40% of the theoretical resolution of the microscope [49].

$$f(x,y) = Aexp\left(-\left(\frac{(x-x_o)^2}{2\sigma_x^2} + \frac{(y-y_o)^2}{2\sigma_y^2}\right)\right) \tag{4.1}$$

$$FWHM = 2\sqrt{2ln2}\sigma \approx 2.35\sigma \tag{4.2}$$

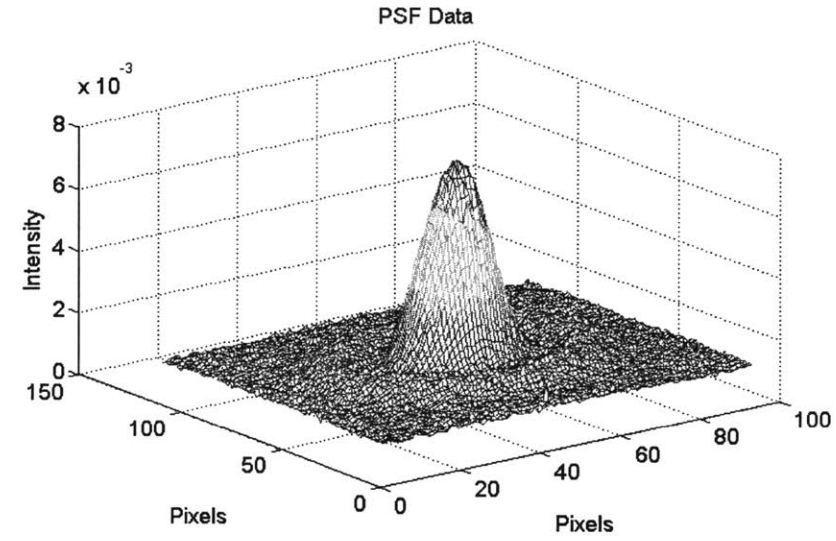
4.2 Noise

4.2.1 Dark current

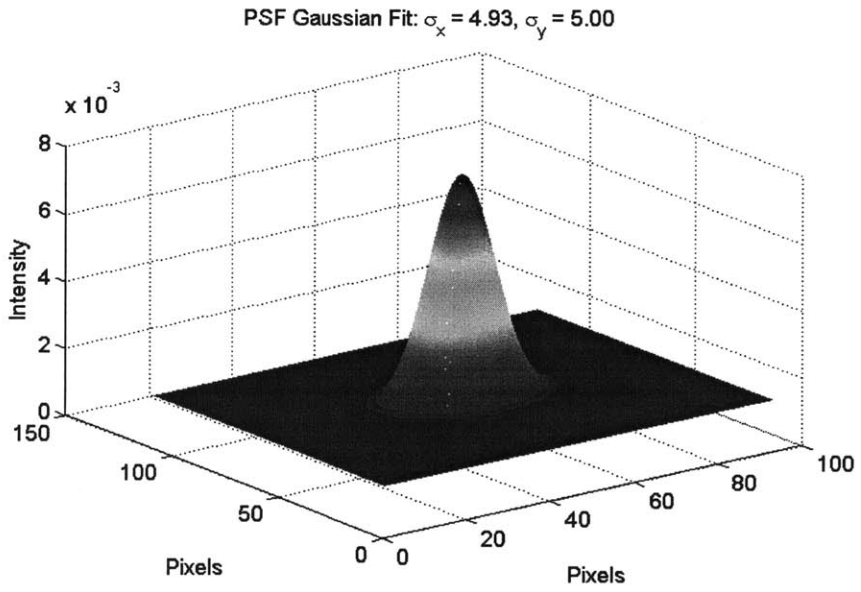
When taking an image with the shutter closed, photons are detected even though theoretically there should be none. This is known as the dark current and the bias of the CCD, arising from electrons created by thermal agitations. The mean dark noise value should increase linearly with exposure time as long as the pixels are not saturated. We measured the dark current because we expected our system to require longer exposure times to compensate for the low fluorescent light levels. The instrument was set up as close to experimental conditions as possible with the laser turned off and a sample mounted on the stage. The camera was uncapped in order to measure the amount of extraneous light in the setup.

Based on the linear increase of dark current I_D with exposure time t and the standard definitions of gain g and bias$B$, we wrote Eq. 4.3 to model the observed pixel intensity M:

$$M = \frac{1}{g}(I_D t + B) \tag{4.3}$$



(a)



(b)

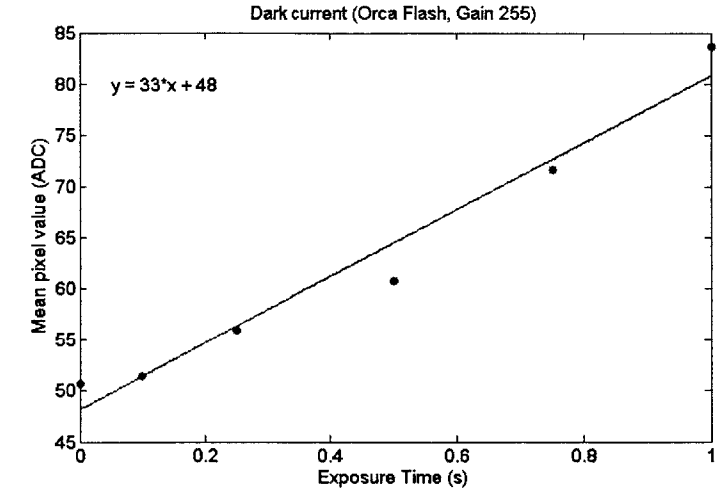
Figure 4-3: a.) PSF image with height representing intensity b.) 2D Gaussian fitted to PSF

The images are taken with either the Orca-Flash or the Orca-ER at maximum gain and varying exposure times. The Orca-Flash has a maximum analog gain of 8x, while the Orca-ER has a maximum gain of 10x. The mean pixel value of each images is graphed in Figure 4-4 along with a linear fit. Dividing the slope of the line by the gain results in a dark current of 4 electrons/pixel/second and 0.15 electrons/pixel/second for the Orca-Flash and Orca-ER respectively. Dividing the y-intercept by the gain gives us biases of 6 and 20 electrons. The measured Orca-ER dark current is comparable to the manufacturer specification of 0.1 electrons/pixel/second, indicating that the setup is well-shielded from external light sources. The Orca-Flash documentation does not provide a value for dark current, but the measured value is on the same order of magnitude as a temperature-extrapolated value of 2 electrons/pixel/second [34].

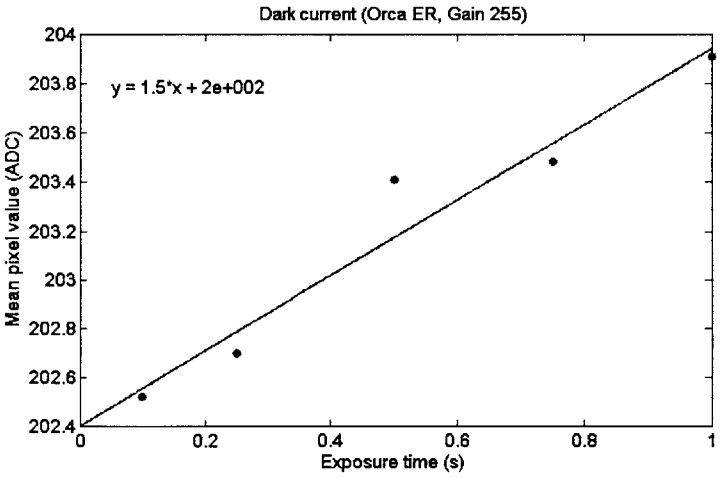
4.2.2 Mechanical instability

In addition to photon noise sources, the resolution of the microscope can also be limited by mechanical vibration and drift because they increase the uncertainty in fluorophore location. We can quantify microscope instability by measuring how the locations of fixed bead samples change over time. Since we did not have any non-PSF beads in the Alexa 647 spectra range, we swapped out the emission filter and imaged 1 μm beads with 580 nm/605 nm excitation/emission wavelengths. The red LED illuminator was used as an excitation source — we set the current to a low 100 mA and the camera gain and exposure were reduced such that the fluorescent beads were much brighter than the background. Two beads in the same FOV were imaged for 3 minutes for a total of 1800 frames (using the Orca-Flash).

In order to improve ease of centroid detection, we converted the particle images to black and white followed by erosion. Then, the particles' weighted centroids were calculated and assembled into two particle tracks. One of the particle tracks is plotted in Figure 4-5— we noted that the particle drifts approximately 10 pixels in the y-dimension (360 nm) and 4 pixels (144 nm) in the x-dimension. A typical 3000-image eSTORM stack may take 25 minutes to acquire, so drift is a major design issue to



(a)



(b)

Figure 4-4: Mean pixel intensity of a.) Orca-Flash dark images and b.) Orca-ER dark images

contend with.

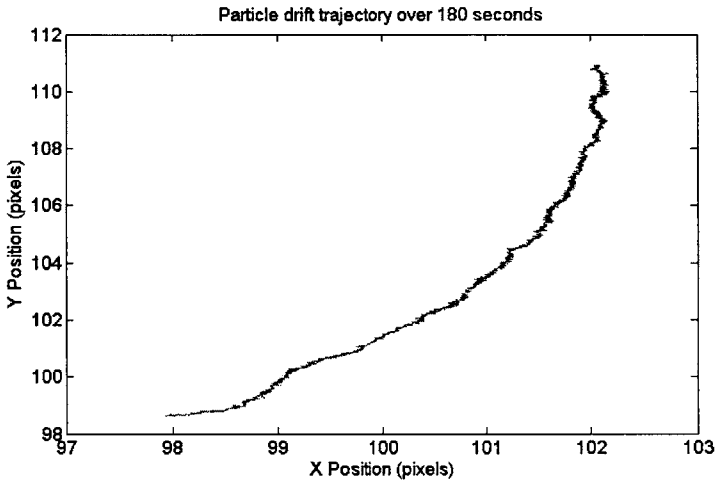
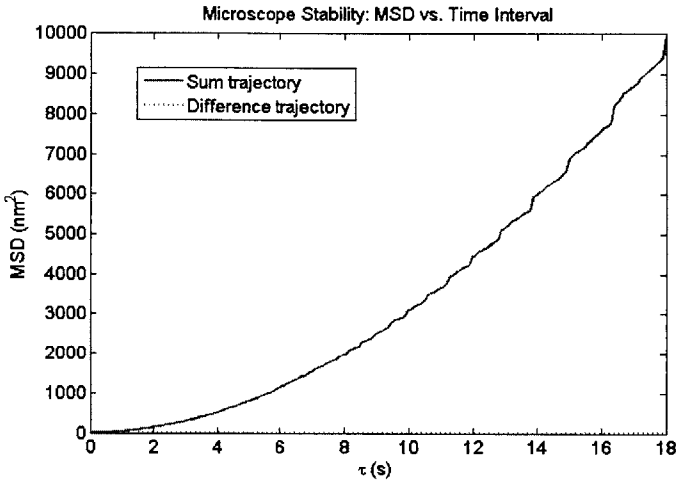


Figure 4-5: Particle trajectory over a 1800 second interval

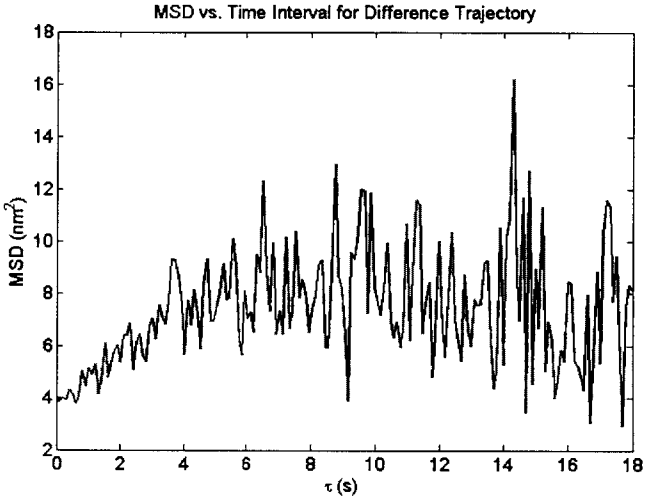
Because mechanical vibration is a stochastic process, we are interested in the averages of the observable physical data. An averaged quantity that is often calculated is called the mean squared displacement (MSD), given by Eq. 4.4:

$$MSD(\tau) =< \Delta r(\tau)^2 >=< |r(t+\tau) - r(t)|^2 > \tag{4.4}$$

where $r(t)$ is the position at time t and τ is the lag time between the two positions. We compute the sum and difference of the trajectories for the two particles. Then, we plot the MSD of the sum and difference trajectories for values of τ up to 18 seconds (Figure 4-6). The plot shows how the sum trajectory MSD increases steadily while the difference trajectory MSD which stays relatively constant at around 10 nm^2. This discovery indicates that the effect of mechanical drift is much greater than the effect of mechanical vibration, especially over long imaging times. It also means that if we can incorporate fixed fluorescent beads or quantum dots into the eSTORM samples, we can correct for drift and significantly improve the localization accuracy.



(a)



(b)

Figure 4-6: Particle Tracking: a.) Mean squared displacement of sum and difference trajectories b.) Mean squared displacement of difference trajectory only

4.3 STORM

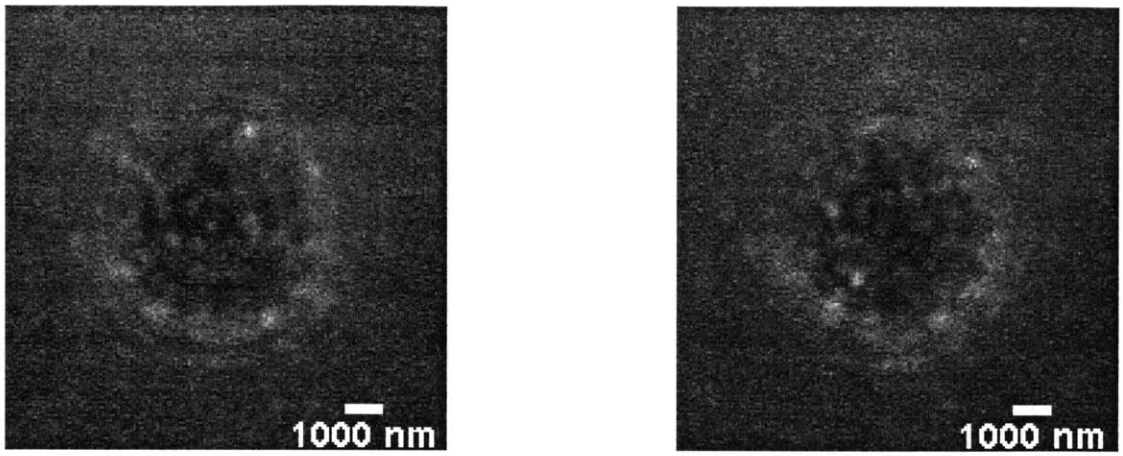
4.3.1 QuickPALM parameters

The QuickPALM software is equipped with 4 main user parameters which affect the number and quality of localizations it finds: (1) Minimum SNR, the minimum signal-to-noise ratio tolerated, (2) Maximum FWHM, the largest tolerated FWHM of a spot, (3) Local Threshold, the minimum tolerated intensity as a fraction of the maximum intensity, and (4) Minimum Symmetry, the minimum tolerated symmetry of the spot. We analyzed the effects of these parameters on the reconstructed image by varying one parameter at a time.

The data set consisted of a 1000-image stack of a 7.2 μm microsphere coated with 10 mg of streptavidin-Alexa 647. We adjusted the stage z-axis to focus just below the bead's maximum diameter plane, since focusing on the maximum diameter ring resulted in too many axially overlapping fluorophores. We measured the laser power as 10 mW at the sample plane. Images were taken with the Orca-ER camera at the maximum gain setting of 255 (10x analog gain) and an exposure of 0.2 seconds. These settings were selected to fill the dynamic range without saturating. Figures 4-7 show example diffraction-limited images randomly selected from two different time points within the stack. The rapidly changing locations of bright spots within the stack demonstrate evidence of photoswitching. The pixels around the dye-labeled bead are not black, indicating a significant amount of background noise in the system.

Generally speaking, more stringent parameters result in both fewer detected particles and less false positives from noise. Ideally, we would have a test sample with known fluorophore locations in order to calculate localization error with respect to truths. However, given the samples we had, we could only quantitatively compare the reconstruction results by tabulating the total number of localizations for each reconstructed image.

Higher SNR and higher FWHM settings produce fewer localizations (Figure 4-8), and the reconstructed image is very sensitive to these parameters. For this stack, we found that an SNR of 5 produced a decent compromise between detection and noise.



(a) (b)

Figure 4-7: a.) t=99.6 seconds, b.) t=124.6 seconds

FWHM is estimated at around 5 pixels, or 322.5 nm. Although one might expect that increasing the Maximum FWHM *increases* the number of accepted candidates, the algorithm implements a noise filter which will filter out spots much smaller than the specified FWHM.

QuickPALM calculates spot intensity by integrating the signal over an area equivalent to the given FWHM after background subtraction [50]. As the Local Threshold is increased, we see that the algorithm rejects more particle candidates which fail the intensity criteria (Figures 4-9). Higher Minimum Symmetry results in fewer localizations as expected. However, as shown in Figures 4-10, changing the symmetry parameter does not affect the reconstruction as drastically as changing the SNR, FWHM, or Threshold. We believe the optimal reconstruction parameters will depend heavily on the signal-to-noise ratio of the particular data set and the optical characteristics of the instrument (such as laser power and PSF)

4.3.2 rapidSTORM parameters

The rapidSTORM software comes with 3 levels of operation complexity: casual, normal, and expert. The expert level incorporates the greatest number of adjustable parameters, but we discovered that PSF FWHM and Amplitude Discarding Thresh-

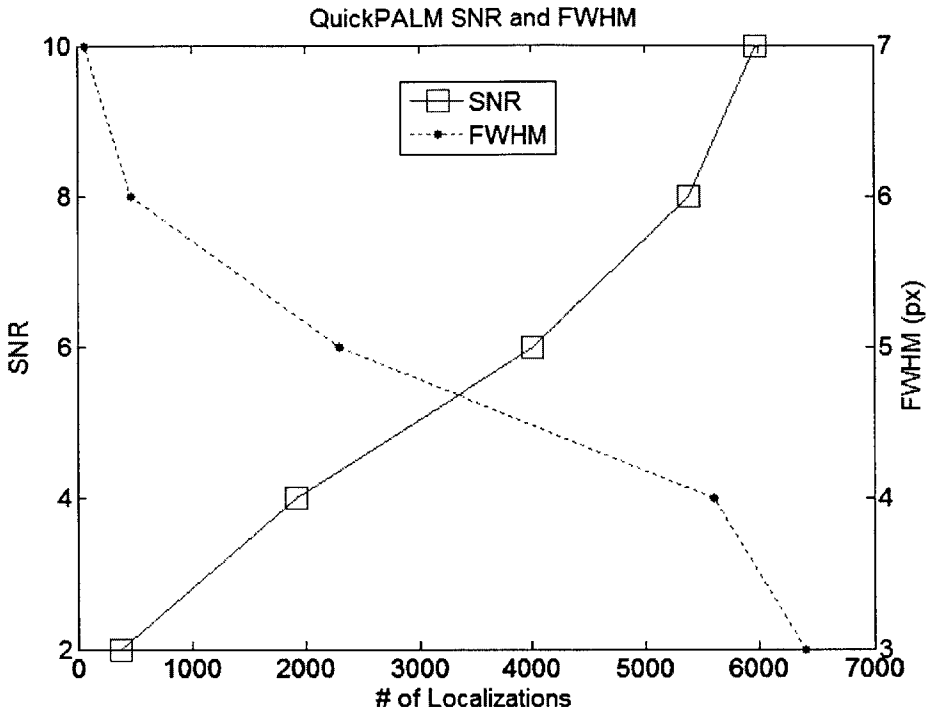
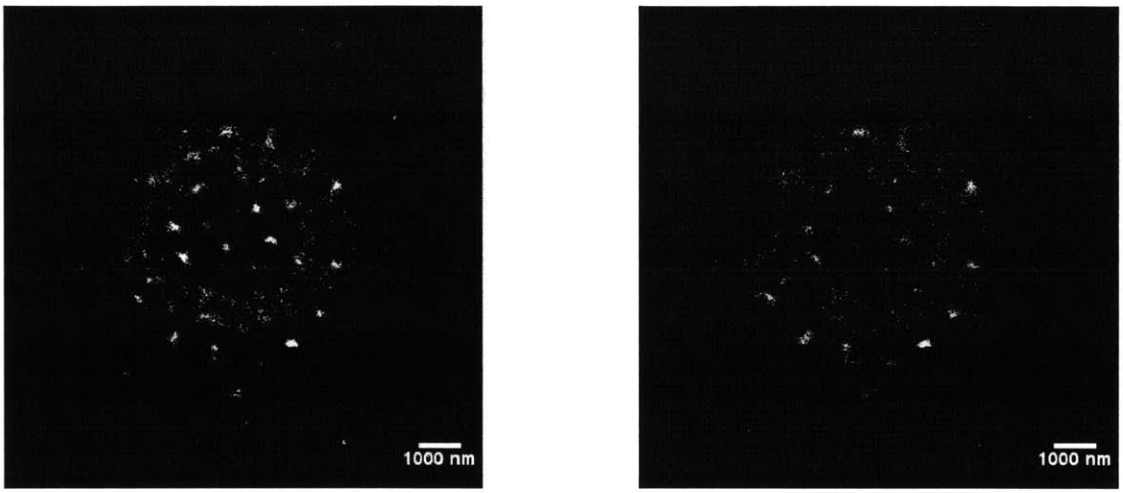


Figure 4-8: Effect of QuickPALM SNR and FWHM parameters on dye-labeled microsphere localization count. SNR was varied while FWHM=5, and FWHM was varied while SNR=5.

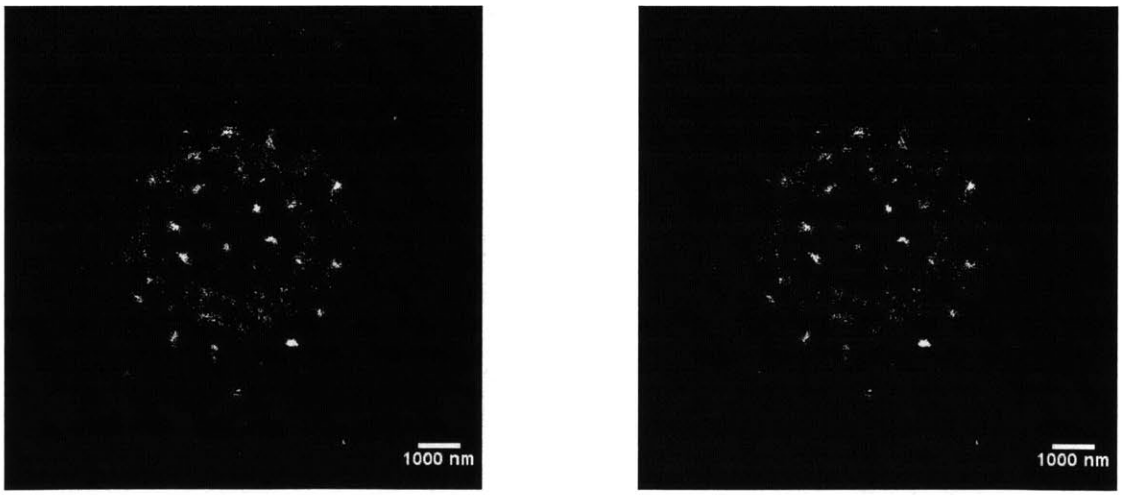
old adjustments make the most noticeable difference in the reconstructed eSTORM images.

The typical width of an emitter PSF, including fluorophore size and camera pixelation effects, is entered in the PSF FWHM field. The algorithm will fit spots in the images with a Gaussian with the specified FWHM. As expected, decreasing the PSF FWHM will increase the number of localizations (Figure 4-11). The Amplitude Discarding Threshold is the minimum amplitude parameter of the fit necessary for a spot to be considered a localization — if the fitted position has an amplitude lower than this value, it is discarded as an artifact. Therefore, increasing this threshold will decrease the number of noise spots. Wolter [52] recommends a minimum amplitude threshold between 1000-4000 A/D counts. Figure 4-12 shows an initial sharp decline in localizations which then starts to level off at high thresholds. The exponential drop can be associated with false positives which are expected to occur with exponentially



(a) (b)

Figure 4-9: Effect of Local Threshold on QuickPALM reconstruction: a.) Local Threshold=20%, 4004 localizations; b.) Local Threshold=80%, 1657 localizations



(a) (b)

Figure 4-10: Effect of Symmetry on QuickPALM reconstructions: a.) Minimum Symmetry=0%, 4004 localizations; b.) Minimum Symmetry=80%, 3050 localizations

decreasing probability. The less steep part can be attributed to particle detections because they exhibit a different probability distribution. A reasonable Amplitude Discarding Threshold would therefore be a value above the transition point, or around 1500-2000 ADC in this case. In practice, we observe that a higher value produces cleaner images without significant feature loss.

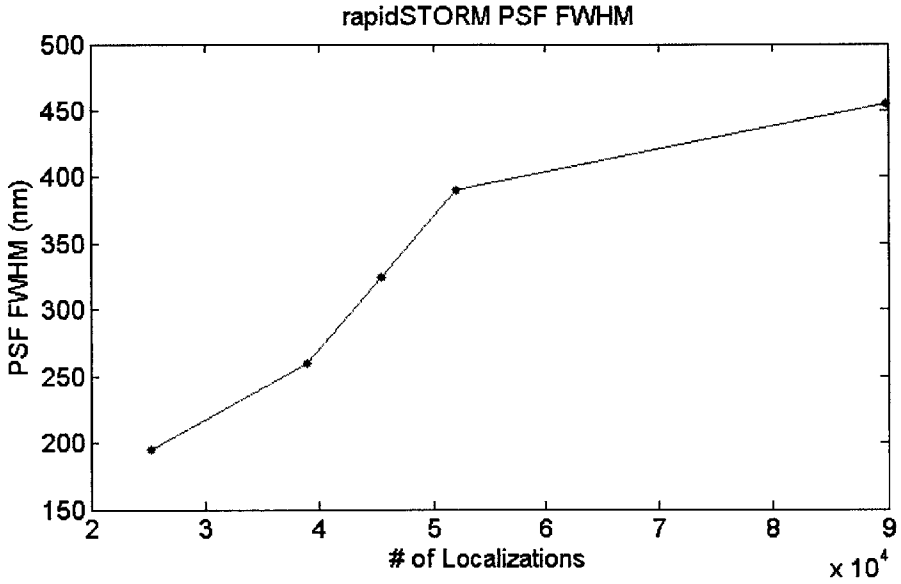


Figure 4-11: Effect of rapidSTORM PSF FWHM on localization count

Besides changing the Amplitude Discarding Threshold and the FWHM, one can adjust the type and size of the Smoothing Filter which is applied before finding spot candidates. By default, rapidSTORM filters out noise in the image with an average mask (Spalttiefpass filter). Median smoothing provides slower, but sometimes more accurate and less blurring smoothing. Erosion is slightly faster than the median filter and gives similar results for small spots. [52] The size of the smoothing mask should be on the order of the PSF width to avoid filtering out particle data.

Changing the Dark Intensity to the measured mean dark pixel value had no effect on the localization results for this data set, and increasing the Minimum Spot Distance only caused small decreases in the number of localizations.

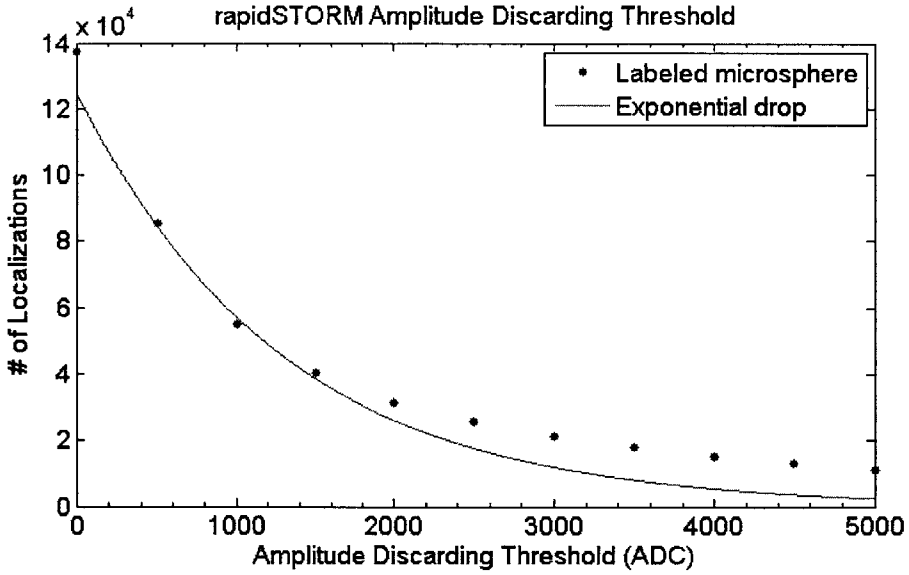


Figure 4-12: Effect of rapidSTORM Amplitude Discarding Threshold on localization count: line fits the exponential drop associated with false positives.

4.3.3 QuickPALM vs. rapidSTORM

QuickPALM and rapidSTORM use the same general idea for single-particle localization in that candidate spots are found, judged according to some critera, and fitted to a Gaussian if that criteria passes. However, they utilize different algorithms and have their respective advantages/disadvantages. We demonstrated with our reconstruction tests that the key difference lies in the way candidate spots are accepted. QuickPALM has an SNR parameter which is used as an absolute criteria for identifying candidates. If the brightest spot in the image does not meet this SNR threshold, the entire image is skipped over [50]. RapidSTORM, on the other hand, sets a low threshold and finds many candidates. These candidates are sorted and inserted into a list sequentially until some motivational parameter is satisfied [52]. In other words, rapidSTORM takes the top X number of candidates instead of taking only candidates above a certain threshold.

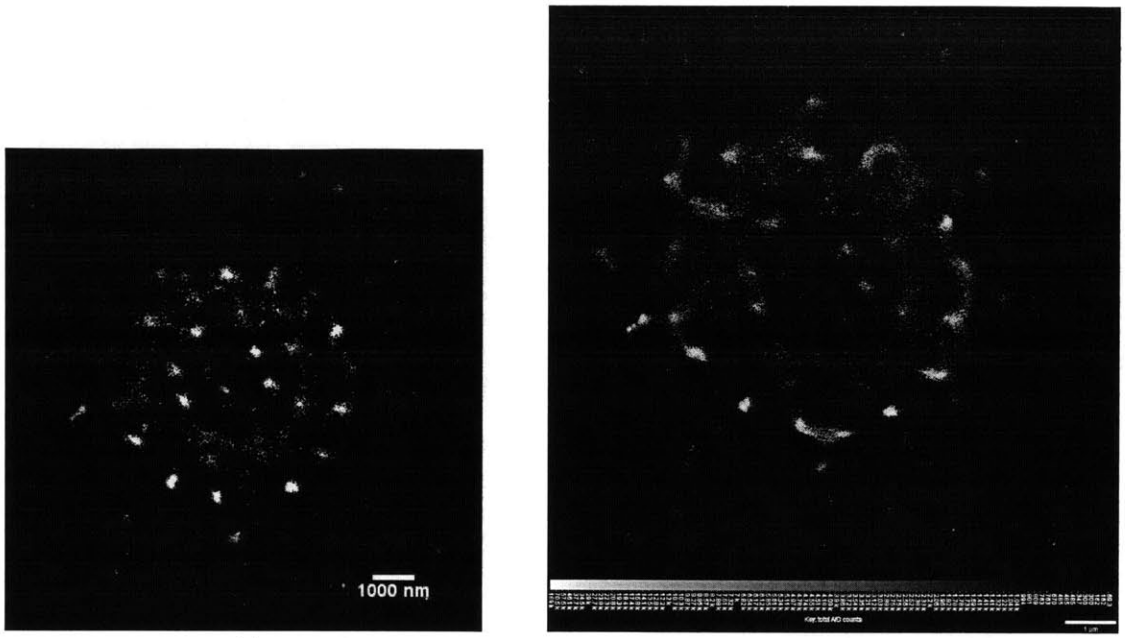
Figure 4-13 displays reconstructed images of the same dye-labeled bead dataset (taken with Orca-ER) processed with the two different programs. The PSF FWHM was fixed to 450 nm in rapidSTORM and accordingly 7 pixels in QuickPALM. Other

software settings were selected to produce the qualitatively best image for each. While several of the high density localization spots appear in both images, but many of the quickPALM spots (Figure 4-13) look brighter and less noisy than the ones in rapidSTORM (Figure 4-13), and QuickPALM also missed some bright spots visible in the rapidSTORM image.

We discovered that rapidSTORM generally produces several thousand more localizations than QuickPALM. This difference appears more noticeably in Figure 4-14, where reconstruction settings were adjusted to produce roughly the same number of localizations in each image. The Amplitude Discarding Threshold for rapidSTORM had to be set very high (7500 ADC) in order to sufficiently reduce the number of localizations. This threshold only leaves the highest intensity spots on the edge of the bead, in contrast to QuickPALM whose localizations comprise many more spots.

Based on the algorithm documentation and our own experience, QuickPALM's Threshold performs a similar function to rapidSTORM's Amplitude Discarding Threshold. QuickPALM and rapidSTORM also both pre-filter the image, but QuickPALM uses a Gaussian bandpass noise filter whereas rapidSTORM uses an averaging filter or a median filter.

As a general rule, we discovered that rapidSTORM is able to quickly find likely fluorophore positions in very noisy images, but it is also more likely to localize that noise. Despite noise detections, it produces images which visibly show the target object regardless of noise. QuickPALM produces more realistic localization counts, but it relies on adjusting the SNR to get the best image and doesn't seem to work as well for low SNR images. In the low SNR Orca-Flash image stack reconstructed in Figure 4-15, QuickPALM appears to localize particles outside of the microsphere which reside outside of the focal plane and do not visibly switch in the image stack. Increasing the SNR parameter will cause localizations in the microsphere perimeter to disappear. Another disadvantage is computation speed: rapidSTORM can perform on the order of 10^5 localizations/second, compared to QuickPALM's rate of 10^3 localizations/second [53]. Although eSTORM does not require real-time processing speed, processing an image stack typically takes on the order of tens of seconds for



(a) (b)

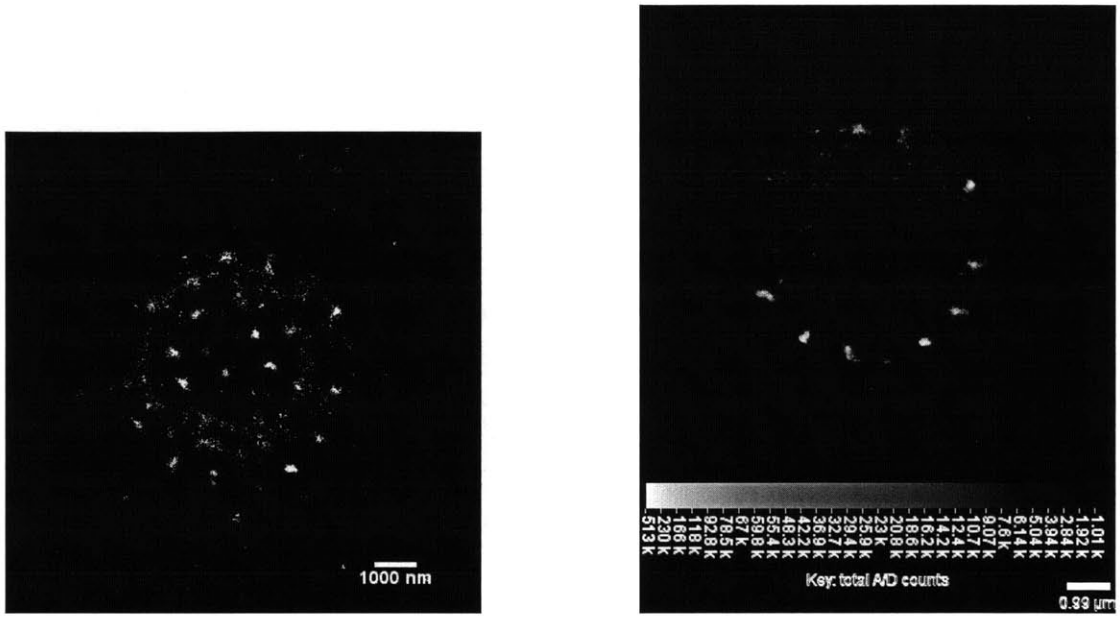
Figure 4-13: Orca-ER data reconstructed by: a.) QuickPALM and b.) rapidSTORM

rapidSTORM and hundreds of seconds for QuickPALM.

The advantage of QuickPALM is that localization counts appear to be more realistic in our images with high SNR. Finally, QuickPALM integrates with ImageJ and allows us to easily pre/post-process the image.

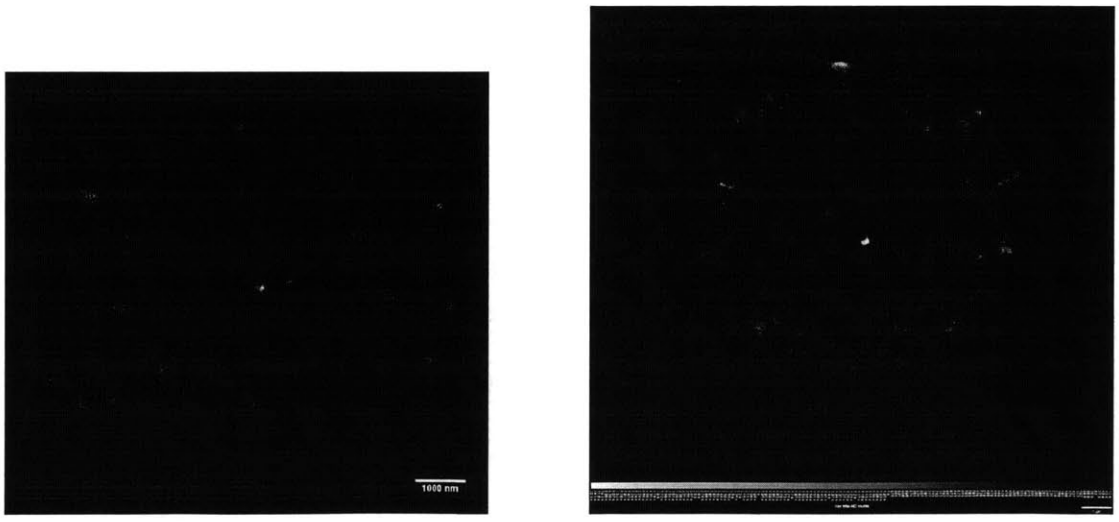
4.3.4 Camera comparison

The goal of eSTORM is not to achieve the absolutely best resolution but rather to find a compromise between performance and cost. The EMCCD camera used universally in commercial TSTORM instruments is designed for low noise acquisition and costs $20,000 alone. EMCCD cameras are not an option given our design constraints, but we could afford a cooled camera with lower background noise such as the Orca-ER. Therefore, we compared the SR image quality produced by the Orca-Flash and the Orca-ER, the latter costing about 3 times the former. Despite having pixels twice the size of the Orca-Flash, the Orca-ER has higher SNR and greater dynamic range which could be important for localization accuracy.



(a) (b)

Figure 4-14: a.) QuickPALM: 4004 localizations; b.) rapidSTORM: 4144 localizations



(a) (b)

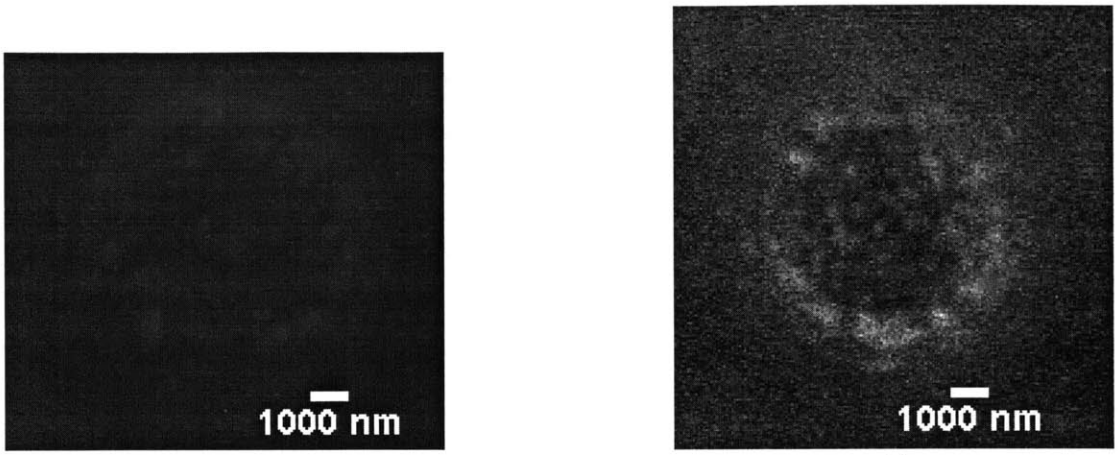
Figure 4-15: Orca-Flash data reconstructed by a.) QuickPALM and b.) rapidSTORM

We imaged different microsphere samples prepared with the same protocol because the sample would deteriorate in the time it takes to image them and switch out cameras. We took images at optimal gain and exposure settings such that switching fluorophores could be observed and were not overexposed. The first image of each stack is shown in Figure 4-16 (before significant photobleaching occurs). Like we expect, the fluorophores appear dimmer with the Orca-Flash and are also surrounded by more background noise. We processed both image stacks in rapidSTORM with a gradient of Amplitude Discarding Thresholds and totaled the number of localizations (Figure 4-12) and the reconstructed image intensity (Figure 4-18). Parameter settings were kept the same in both reconstructions. At low thresholds, the Orca-Flash has more localizations due to noise. Less localizations are found for the Orca-Flash at high thresholds due to the difference in SNR. Figure 4-19 shows images reconstructed with the same software parameters. The Orca-ER finds more spots, and the spots have higher intensity overall.

Although we could not ensure that the samples were exactly the same, it makes sense that the low SNR, low quantum yield Orca-Flash would produce images with lower intensity (Figure 4-18) and consequently more localization uncertainty. The Orca-ER also requires less laser power and shorter exposure times than the Orca-Flash, meaning that we could offset the price differential by using a less expensive laser and also image faster with less bleaching and drift effects. Further optimization is needed to determine the best cost/performance trade-off. However, in the meantime, we demonstrate in the following section that we can achieve sub-diffraction resolution even with the Orca-Flash camera.

4.3.5 Actin imaging

Individual F-actin microfilaments are approximately 7 nm in diameter and are organized into bundles or networks via cross-linking proteins [55]. We performed eSTORM imaging on 3T3 actin by capturing 1900 images with the Orca-Flash (gain=255, exposure=0.5 sec), cropping regions of interest, and reconstructing the cropped stack with both QuickPALM and rapidSTORM. We arbitrarily selected the number of im-



(a) (b)

Figure 4-16: a.) The first image of a 7.2 μm microsphere in a.) an Orca-Flash stack and b.) an Orca-ER stack

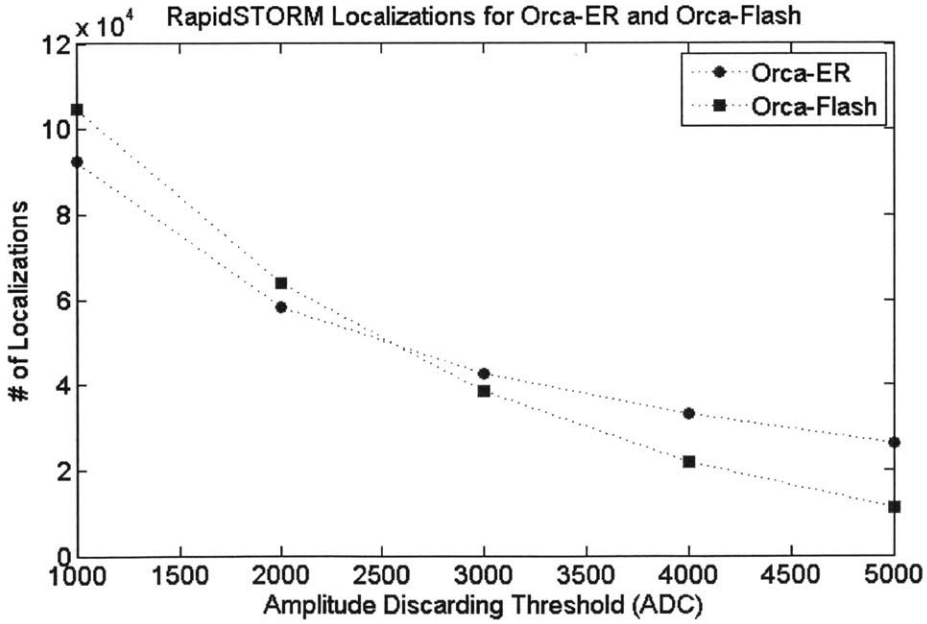


Figure 4-17: Localization counts for image stacks taken with Orca-Flash and Orca-ER cameras.

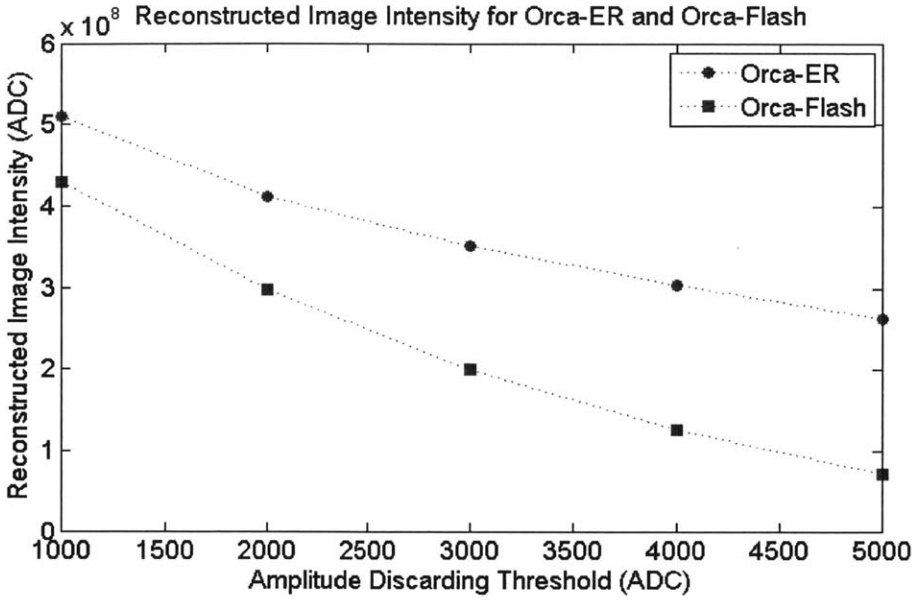
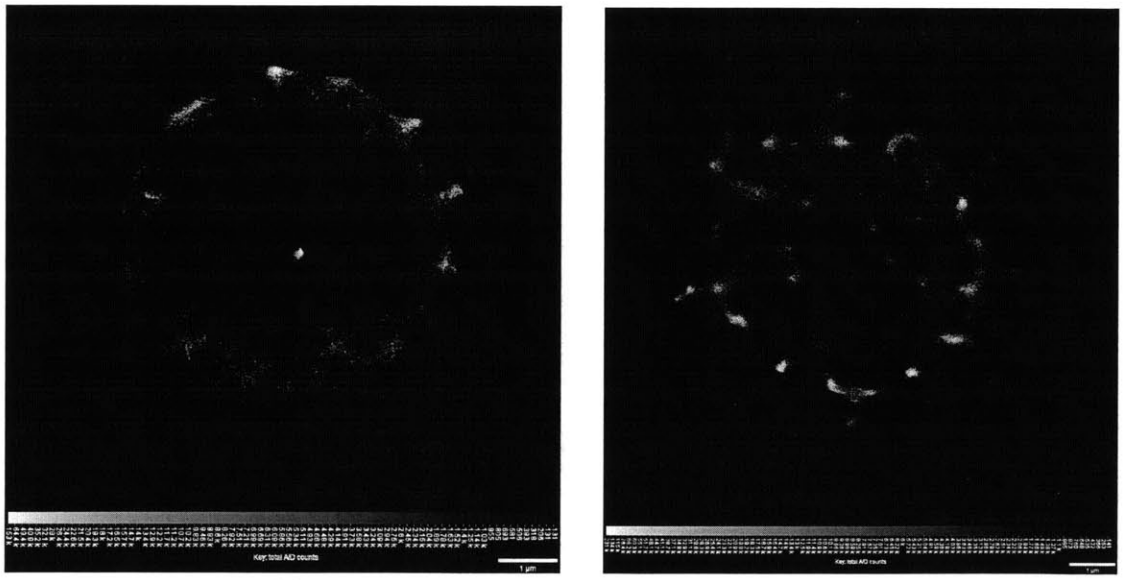


Figure 4-18: Total image intensity of reconstructed images for image stacks taken with Orca-Flash and Orca-ER cameras.



(a) (b)

Figure 4-19: a.) Reconstructed Orca-Flash data and b.) Reconstructed Orca-ER data

ages from a range of 1000-3000. The standard method of measuring SR microscopy resolution is to make a sample with a very sparse labeling density and calculate the variance in localization position for a single fluorophore. Since we encountered difficulty making a sparsely labeled sample that emitted enough light for our system, instead we used the reconstructed actin images to estimate the resolution limits of our instrument.

A diffraction-limited image of an actin region is shown in Figure 4-20. We can see that an actin bundle branches into two, but the blur does not allow us to resolve the size of these branches. After reconstruction, the resolution has improved significantly (Figure 4-21). The left branch has well-defined edges, and we measure its width as approximately 350 nm. Of even more interest is how the split between the two branches is now brought into view. Because the split is an area of high spatial frequency, we realized we could estimate the resolution by modeling the split and finding the Gaussian kernel which would make the feature unresolvable. We created a simple object model in ImageJ(Figure 4-22a) and applied Gaussian filters with increasing σ until the split became blurred and unresolvable (Figure 4-21b). Using Eq. 4.2, we find that the resolution is approximately 70 nm.

Another way we estimated the resolution was to look for the smallest well-defined and measurable feature. Figure 4-24 presents SR reconstructions of the actin in Figure 4-23. A thin, bright actin bundle is visible in the rapidSTORM reconstruction. This feature has a width between 2-3 pixels, or 72-108 nm. This estimate is on the same order of magnitude as the resolution found using the Gaussian blur method.

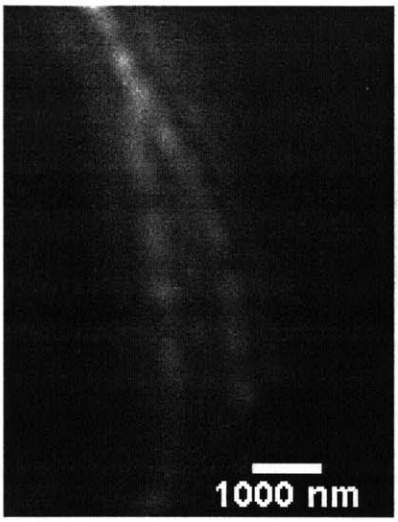
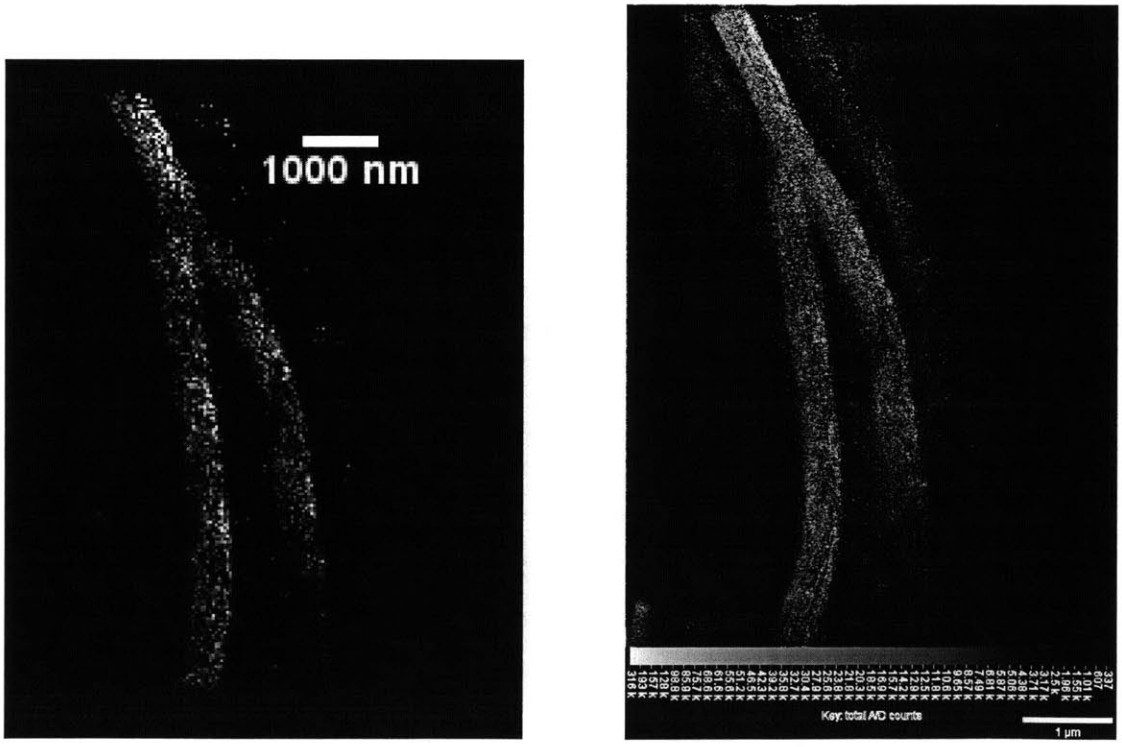
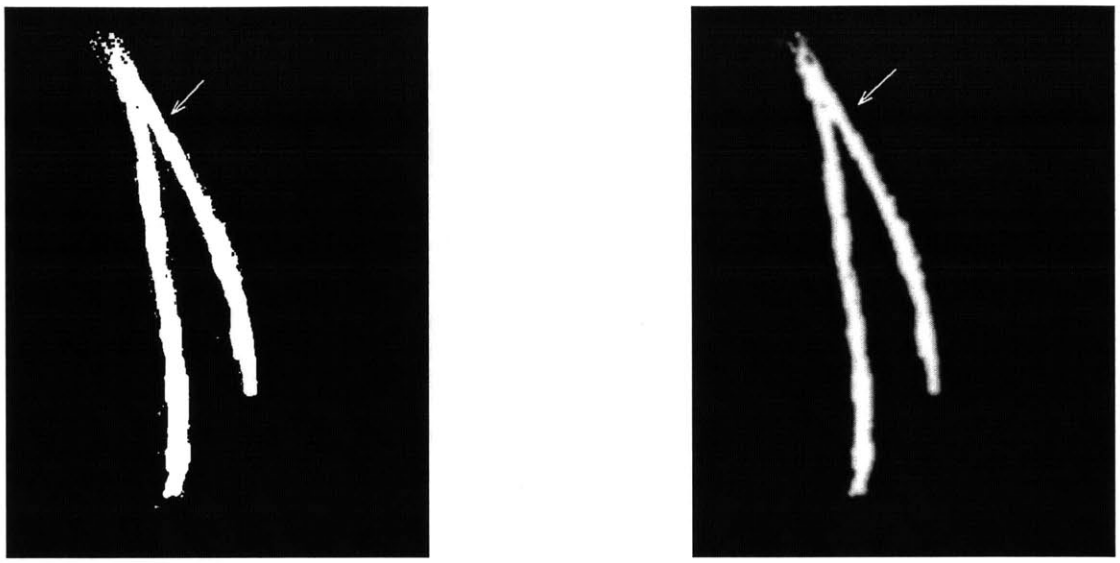


Figure 4-20: Diffraction-limited image of actin #1



(a) (b)

Figure 4-21: Actin #1 reconstructed with a.) QuickPALM and b.) rapidSTORM



(a) (b)

Figure 4-22: a.) Binary image modeling actin #1 branch - the branching point (indicated by the arrow) contains high frequency information; b.) Model filtered with Gaussian with $\sigma = 30$ nm;

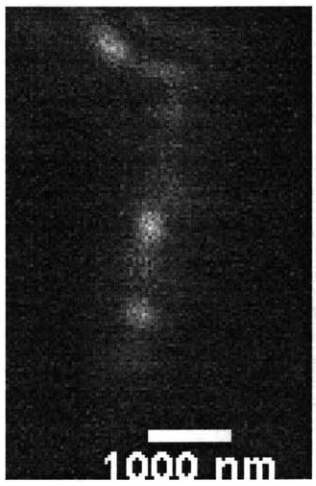
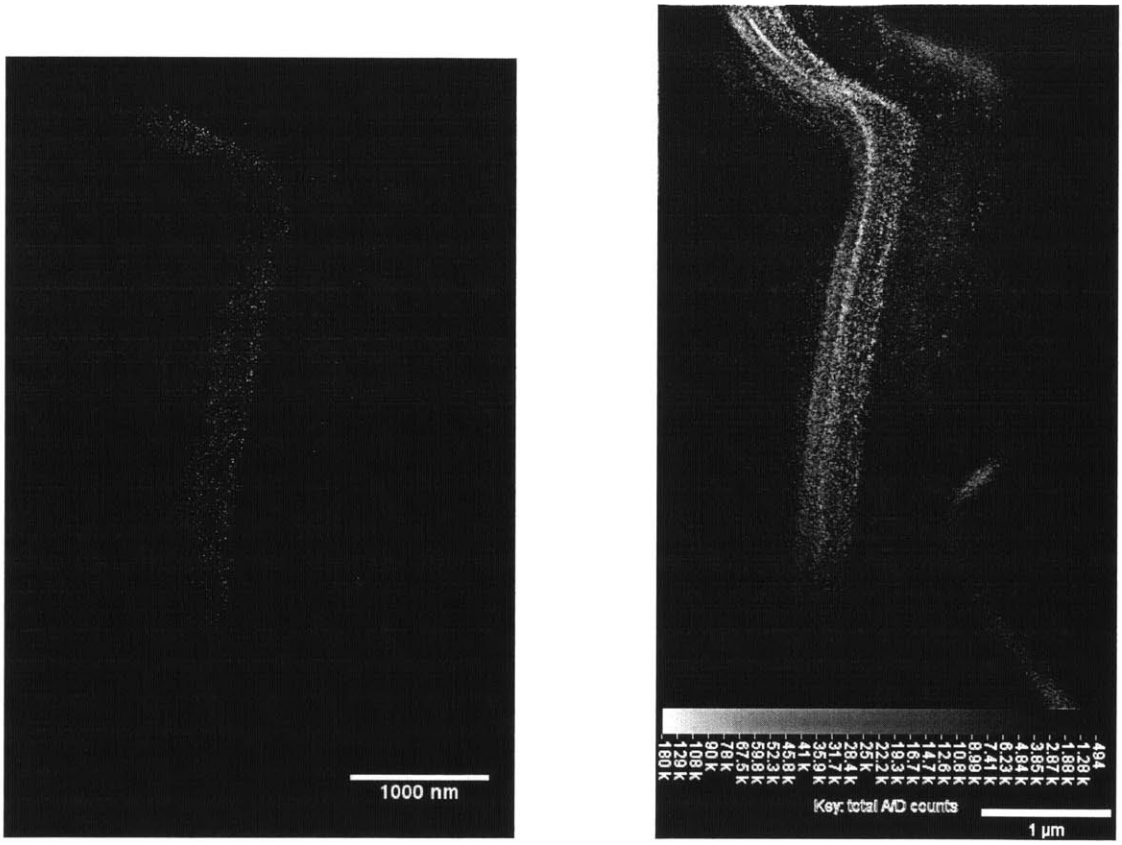


Figure 4-23: Diffraction-limited image of actin #2



(a) (b)

Figure 4-24: Actin #2 reconstructed with a.) QuickPALM and b.) rapidSTORM

Chapter 5

Conclusions

A custom stochastical optical reconstruction microscope was demonstrated to achieve a sub-diffraction resolution between 70 - 100 nm. The total system components cost well under the $20,000 goal, as detailed by the cost analysis table in the appendix. This lower cost STORM instrument makes it feasible for teaching laboratories to give students a hands-on introduction to an increasingly prominent biological research tool. Our photon budget estimate revealed that the most expensive parts of a commercial system, namely the camera, objectives, and laser, could be replaced by lower cost alternatives and still achieve super-resolution.

Making a robust biological sample was one of the most difficult parts of the project. We successfully developed a streptavidin-Alexa 647 coated microsphere sample which could be used for basic instrument testing. We also replicated actin staining protocols from the Zhuang lab and showed that they work with eSTORM. However, we observed some variation in the sample quality, possibly due to factors such as chemicals aging and differences in cell growth phase. Next time, it may be worth examining the effect of adding a blocking buffer before using detergent to prevent nonspecific binding and reduce background noise. In addition, future efforts should continue to develop a novel protocol for creating photoswitching plant samples. One could start out by doing more chemical concentration and incubation time adjustments.

Writing the simulator was useful for understanding some key properties in the STORM imaging system, namely fluorophore, noise, and camera properties. An

interesting next step would be to link experimental data to simulator parameters in order to help evaluate cost/performance trade-offs. For example, basically all other STORM instruments utilize a Total Internal Reflection Fluorescence (TIRF) objective, which is more expensive but effectively rejects background noise from the molecules that are out of focus by exciting only the molecules at the sample surface [56]. We discovered that the cooled Orca-ER camera produced higher SNR stacks than the Orca-Flash, but perhaps that price difference should go towards a TIRF objective.

We analyzed the differences between reconstructing real image stacks with two prevalent software packages, QuickPALM and rapidSTORM. In contrast to articles which compare them with feature checklists and focus on speed [53] , we compared their reconstruction quality using real samples. We found that QuickPALM produced comparatively better images at higher SNR, and vice versa for rapidSTORM.

The eSTORM system's noise properties were also characterized because higher noise is usually the price one pays for lower-cost components. We noted that the instrument is subject to mechanical drift but not much vibrational noise. There exist functions in QuickPALM and rapidSTORM which perform drift correction using fixed beads in the sample. The limitation of this tracking method is that it cannot do anything when samples drift out of the FOV. Motorized stage actuators could be programmed to turn based on the bead track, but this would certainly add cost and complexity.

A way to reduce the effect of drift while simultaneously reducing photobleaching effects is to shorten the acquisition time. A low-cost idea is to synchronize the laser's analog modulation inputs with the camera shutter — the camera shutter should be closed when the lasers are pulsing, and they should open immediately after each pulse to maximize photon collection.

Finally, one future avenue to explore is to write software that identifies whether multiple localizations likely come from the same particle. Not only would this information enhance reconstruction contrast, it could also be used to correct drift and provide a more definitive measure of the instrument resolution.

Chapter 6

Appendix

6.1 Tables

Component	Cost
Orca Flash 2.8	$3000
Laser, 150 mW	$3000
Stage	$1200
Optomechanics	$1000
Aspheric lenses	$500
Objective, 100x /1.25	$400
Emission Filter	$300
Dichroic Mirror	$300
Tube lens	$250
TOTAL	**$9950**

eSTORM Cost Estimate

6.2 Code

Simulator

```
function fluor_locations = make_actin (x1, y1, z1, varargin)
    %Example:  actin1=make_actin(20,50,1);
        %      actin2=make_actin(50,50,1);
        %      fluor_locations=[actin1;actin2];
    p=inputParser;
    p.addRequired('x1', @isnumeric);
    p.addRequired('y1');
    p.addRequired('z1');
    p.addParamValue('n', 1000, @isnumeric);  %number of fluorophores
        per actin
    p.addParamValue('x_length', 20, @isnumeric);
    p.addParamValue('y_length', 200, @isnumeric);
    p.addParamValue('z_length', 20, @isnumeric);
    p.parse(x1,y1,z1,varargin{:});
    parameters=p.Results;


%monomorsize=5e-9;  %5nm monomer width
%pixel=5e-9; %5nm per pixel (1 monomer per pixel)
%filamentlength=1e-6; %1 um per filament
%filamentwidth=100e-9; %100nm width

%Actin filament
a=parameters.x1; b=a+parameters.x_length-1;
rx1 = a + (b-a).*rand(parameters.n,1);
a=parameters.y1; b=a+ parameters.y_length-1;
ry1=a + (b-a).*rand(parameters.n,1);
a=1;b=20;
rz1=a + (b-a).*rand(parameters.n,1);

fluor_locations=[rx1,ry1,rz1];
%fluor_locations=[rx1,ry1,rz1;rx2,ry2,rz2];
```

```
%Object is 3D actin "bars" spaced by less than 200nm
```

```
function obj=aliasedbin3d(location,x,y,z,w)
%Map fluorophore locations to empty grid based on distance.
%linearly decreasing with distance  (essentially triangle filter)

%location: Nx3
%x,y,z: size of grid
%w: scalar determining extent of aliasing

obj=zeros(x,y,z);
if (location~=0)
for num=1:size(location,1)
x0=location(num,1);
y0=location(num,2);
z0=location(num,3);
for i=ceil(x0-w):floor(x0+w);
    for j=ceil(y0-w):floor(y0+w);
        for k=ceil(z0-w):floor(z0+w);
            if(i>0 && j>0 && k>0)
d=sqrt((i-x0).^2+(j-y0).^2+(k-z0).^2);
value=1-(d/w);
if(value>0)
obj(i,j,k)=value;
end
            end
        end
    end
end
end
end
```

```
/*
 *
 * File:   main.cpp
 * Author: ranbel
 *
```

```
 * Created on March 8, 2012, 3:55 PM
 */

//Algorithm from "Numerical Recipes in C: The Art of Scientific
   Computing" by Press et al.
\\Requires nrutil.c from <http://www.nr.com/pubdom/nrutil.c.txt>

#include <stdlib.h>
#include <math.h>
extern "C" {
    #include "nrutil.h"
}
#include "mex.h"

#define SWAP(a,b) tempr=(a);(a)=(b);(b)=tempr
//#define N 4


void fourn(float data[], unsigned long nn[], int ndim, int isign) {
    int idim;
    unsigned long i1, i2, i3, i2rev, i3rev, ip1, ip2, ip3, ifp1,
       ifp2;
    unsigned long ibit, k1, k2, n, nprev, nrem, ntot;
    float tempi, tempr;
    double theta, wi, wpi, wpr, wr, wtemp;
    for (ntot = 1, idim = 1; idim <= ndim; idim++)
        ntot *= nn[idim];
    nprev = 1;
    for (idim = ndim; idim >= 1; idim--) {
        n = nn[idim];
        nrem = ntot / (n * nprev);
        ip1 = nprev << 1;
        ip2 = ip1*n;
        ip3 = ip2*nrem;
        i2rev = 1;
        for (i2 = 1; i2 <= ip2; i2 += ip1) {
```

```
40              if (i2 < i2rev) {
41                  for (i1 = i2; i1 <= i2 + ip1 - 2; i1 += 2) {
42                      for (i3 = i1; i3 <= ip3; i3 += ip2) {
43                          i3rev = i2rev + i3 - i2;
44                          SWAP(data[i3], data[i3rev]);
45                          SWAP(data[i3 + 1], data[i3rev + 1]);
46                      }
47                  }
48              }
49              ibit = ip2 >> 1;
50              while (ibit >= ip1 && i2rev > ibit) {
51                  i2rev -= ibit;
52                  ibit >>= 1;
53              }
54              i2rev += ibit;
55          }
56          ifp1 = ip1;
57          while (ifp1 < ip2) {
58              ifp2 = ifp1 << 1;
59              theta = isign * 6.28318530717959 / (ifp2 / ip1);
60              wtemp = sin(0.5 * theta);
61              wpr = -2.0 * wtemp*wtemp;
62              wpi = sin(theta);
63              wr = 1.0;
64              wi = 0.0;
65              for (i3 = 1; i3 <= ifp1; i3 += ip1) {
66                  for (i1 = i3; i1 <= i3 + ip1 - 2; i1 += 2) {
67                      for (i2 = i1; i2 <= ip3; i2 += ifp2) {
68                          k1 = i2;
69                          k2 = k1 + ifp1;
70                          tempr = (float) wr * data[k2]-(float) wi *
                                data[k2 + 1];
71                          tempi = (float) wr * data[k2 + 1]+(float) wi
                                * data[k2];
72                          data[k2] = data[k1] - tempr;
73                          data[k2 + 1] = data[k1 + 1] - tempi;
```

```
74                          data[k1] += tempr;
75                          data[k1 + 1] += tempi;
76                      }
77                  }
78                  wr = (wtemp = wr) * wpr - wi * wpi + wr;
79                  wi = wi * wpr + wtemp * wpi + wi;
80              }
81              ifp1 = ifp2;
82          }
83          nprev *= n;
84
85      }
86  }
87
88  void rlft3(float ***data, float **speq, unsigned long nn1, unsigned
        long nn2,
89          unsigned long nn3, int isign) {
90      void fourn(float data[], unsigned long nn[], int ndim, int isign
            );
91      void nrerror(char error_text[]);
92      unsigned long i1, i2, i3, j1, j2, j3, nn[4], ii3;
93      double theta, wi, wpi, wpr, wr, wtemp;
94      float c1, c2, h1r, h1i, h2r, h2i;
95      if (1 + &data[nn1][nn2][nn3]-&data[1][1][1] != nn1 * nn2 * nn3)
96          nrerror("rlft3: problem with dimensions or contiguity of
                data array\n");
97      c1 = 0.5;
98      c2 = -0.5 * isign;
99      theta = isign * (6.28318530717959 / nn3);
100     wtemp = sin(0.5 * theta);
101     wpr = -2.0 * wtemp*wtemp;
102     wpi = sin(theta);
103     nn[1] = nn1;
104     nn[2] = nn2;
105     nn[3] = nn3 >> 1;
106     if (isign == 1) {
```

```
107         fourn(&data[1][1][1] - 1, nn, 3, isign);
108         for (i1 = 1; i1 <= nn1; i1++)
109             for (i2 = 1, j2 = 0; i2 <= nn2; i2++) {
110                 speq[i1][++j2] = data[i1][i2][1];
111                 speq[i1][++j2] = data[i1][i2][2];
112             }
113     }
114     for (i1 = 1; i1 <= nn1; i1++) {
115         j1 = (i1 != 1 ? nn1 - i1 + 2 : 1);
116         wr = 1.0;
117         wi = 0.0;
118         for (ii3 = 1, i3 = 1; i3 <= (nn3 >> 2) + 1; i3++, ii3 += 2)
                {
119             for (i2 = 1; i2 <= nn2; i2++) {
120                 if (i3 == 1) {
121                     j2 = (i2 != 1 ? ((nn2 - i2) << 1) + 3 : 1);
122                     h1r = c1 * (data[i1][i2][1] + speq[j1][j2]);
123                     h1i = c1 * (data[i1][i2][2] - speq[j1][j2 + 1]);
124                     h2i = c2 * (data[i1][i2][1] - speq[j1][j2]);
125                     h2r = -c2 * (data[i1][i2][2] + speq[j1][j2 + 1])
                            ;
126                     data[i1][i2][1] = h1r + h2r;
127                     data[i1][i2][2] = h1i + h2i;
128                     speq[j1][j2] = h1r - h2r;
129                     speq[j1][j2 + 1] = h2i - h1i;
130                 } else {
131                     j2 = (i2 != 1 ? nn2 - i2 + 2 : 1);
132                     j3 = nn3 + 3 - (i3 << 1);
133                     h1r = c1 * (data[i1][i2][ii3] + data[j1][j2][j3
                            ]);
134                     h1i = c1 * (data[i1][i2][ii3 + 1] - data[j1][j2
                            ][j3 + 1]);
135                     h2i = c2 * (data[i1][i2][ii3] - data[j1][j2][j3
                            ]);
136                     h2r = -c2 * (data[i1][i2][ii3 + 1] + data[j1][j2
                            ][j3 + 1]);
```

```
137                     data[i1][i2][ii3] = h1r + wr * h2r - wi*h2i;
138                     data[i1][i2][ii3 + 1] = h1i + wr * h2i + wi*h2r;
139                     data[j1][j2][j3] = h1r - wr * h2r + wi*h2i;
140                     data[j1][j2][j3 + 1] = -h1i + wr * h2i + wi*h2r;
141                 }
142             }
143             wr = (wtemp = wr) * wpr - wi * wpi + wr;
144             wi = wi * wpr + wtemp * wpi + wi;
145         }
146     }
147     if (isign == -1)
148         fourn(&data[1][1][1] - 1, nn, 3, isign);
149 }
150
151 void mexFunction(int nlhs, mxArray *plhs[ ], int nrhs, const mxArray
        *prhs[ ]) {
152     //definitions
153    const mxArray *dataget;
154     const mxArray *psfget;
155     //values from mxarrays
156     double * dataValues_get;
157     double * psfValues_get;
158     double *dataOut;
159     double *speqOut;
160    // float ***dataOut;
161    // float **speqOut;
162     int dim;
163     const mwSize *dims=0;
164   int index1,index2,index3;
165   int i1, i2, i3;
166   int fakeout;
167     int j=0;
168     float fac, r, i, ***data1, ***data2, **speq1, **speq2, *sp1, *
          sp2;
169     void rlft3(float ***data, float **speq, unsigned long nn1,
170             unsigned long nn2, unsigned long nn3, int isign);
```

```
171
172
173     //get values from input mxarrays
174     dataget = prhs[0];
175     psfget = prhs[1];
176     psfValues_get = mxGetPr(psfget);
177     dataValues_get = mxGetPr(dataget);
178    dims = mxGetDimensions(dataget);
179    i1=dims[2];
180   i2=dims[0];
181   i3=dims[1];
182
183     //Allocate memory and assign output pointer
184     plhs[0] = mxCreateNumericArray(3, dims, mxDOUBLE_CLASS, mxREAL);
          //mxReal is our data-type
185     plhs[1] = mxCreateNumericMatrix(i1, 2 * i2, mxDOUBLE_CLASS,
        mxREAL);
186     //Get a pointer to the data space in our newly allocated memory
187     dataOut = mxGetPr(plhs[0]);
188     speqOut = mxGetPr(plhs[1]);
189   //int el =mxGetNumberOfElements(dataget);
190
191     data1 = f3tensor(1, i1, 1, i2, 1, i3);
192     data2 = f3tensor(1, i1, 1, i2, 1, i3);
193     speq1 = matrix(1, i1, 1, 2 * i2);
194     speq2 = matrix(1, i1, 1, 2 * i2);
195     /* load data1 and data2 */
196
197
198       for (index1=1; index1<=i1; index1++){
199         for (index3=1;index3<=i3;index3++){
200         //  for (index2=i2;index2>0;index2--){
201           for (index2=1;index2<=i2;index2++){
202         data1[index1][index2][index3]=(float) dataValues_get[j];
203         data2[index1][index2][index3]=(float) psfValues_get[j];
204         j++;
```

```
205       }
206     }
207   }
208
209
210
211    rlft3(data1, speq1, i1, i2, i3, 1);
212    rlft3(data2, speq2, i1, i2, i3, 1);
213
214
215
216     fac = 2.0 / (i1 * i2 * i3); //normalization factor
217
218     sp1 = &data1[1][1][1];
219     sp2 = &data2[1][1][1];
220     for (j = 1; j <= i1 * i2 * i3 / 2; j++) {
221         r = sp1[0] * sp2[0] - sp1[1] * sp2[1];  //Real part of
              multiplication
222         i = sp1[0] * sp2[1] + sp1[1] * sp2[0];  //Imaginary part
223         sp1[0] = fac*r;
224         sp1[1] = fac*i;
225         sp1 += 2;
226         sp2 += 2;
227     }
228     sp1 = &speq1[1][1];
229     sp2 = &speq2[1][1];
230     for (j = 1; j <= i1 * i2; j++) {
231         r = sp1[0] * sp2[0] - sp1[1] * sp2[1];
232         i = sp1[0] * sp2[1] + sp1[1] * sp2[0];
233         sp1[0] = fac*r;
234         sp1[1] = fac*i;
235         sp1 += 2;
236         sp2 += 2;
237     }
238
239
```

```
240
241
242     rlft3(data1, speq1, i1, i2, i3, -1); //inverse FFT of filtered
            transform
243     /*unload data1 and speq1 here*/
244
245
246   j=0;
247   for (index1=1; index1<=i1; index1++){
248       for (index3=1;index3<=i3;index3++){
249         //  for (index2=i2;index2>0;index2--){
250         for (index2=1;index2<=i2;index2++){
251         dataOut[j]=(double) data1[index1][index2][index3];
252         j++;
253       }
254     }
255   }
256   j=0;
257
258       for (index1=1; index1<=i1;index1++){
259         //for (index2=i2*2;index2>0;index2--){
260           for (index2=1;index2<=i2*2;index2++){
261       speqOut[j] = speq1[index1][index2];
262       j++;
263     }
264   }
265
266     /**/
267
268     free_matrix(speq2, 1, i1, 1, 2 * i2);
269     free_matrix(speq1, 1, i1, 1, 2 * i2);
270     free_f3tensor(data2, 1, i1, 1, i2, 1, i3);
271     free_f3tensor(data1, 1, i1, 1, i2, 1, i3);
272
273     return;
274 }
```

```
function psfstack= loadtifs(myFolder)
%myFolder = 'C:\Users\ranbel\Documents\MATLAB\eSTORM\Simulator\
    psf_test';
if ~isdir(myFolder)
  errorMessage = sprintf('Error: The following folder does not exist
     :\n%s', myFolder);
  uiwait(warndlg(errorMessage));
  return;
end
%figure;
filePattern = fullfile(myFolder, '*.tif');
psfFiles = dir(filePattern);
for k = 1:length(psfFiles)
  baseFileName = psfFiles(k).name;
  fullFileName = fullfile(myFolder, baseFileName);
  %fprintf(1, 'Now reading %s\n', fullFileName);
  psfstack(:,:,k) = imread(fullFileName);
  %imshow(psfstack(:,:,k));
  %drawnow; %  update display immediately
end

end
```

Stability

```
function []=msd_storm(data)

%Tracks.m found here:  http://www.mathworks.com/matlabcentral/
    fileexchange/42573-particle-point-analysis/content/track.m

f=1800;
centroidslist=[];
for k=1:f
im=data(:,:,k);
%figure (1);
```

```
%imshow(im);

bw_im=im2bw(im, 0.02);
%imshow(bw_im);

eroded_bw_im=bwmorph(bw_im, 'erode');
%figure;
%imshow(eroded_bw_im);

bw_particles=bwlabel(eroded_bw_im);
s  = regionprops(bw_particles, im,'WeightedCentroid');
centroids = cat(1, s.WeightedCentroid);
% imshow(bw_particles)
% hold on
% plot(centroids(:,1), centroids(:,2), 'b*')
% drawnow;
% hold off
centroidslist=[centroidslist; [centroids, ones(size(centroids,2),1)
    .*k]];
end

tracks=track(centroidslist, 10);

particlenames=tracks(1:end, 4);

for i=1:size(unique(particlenames));
    temp=find(particlenames==i);
particle{i}.x=tracks(temp,1);
particle{i}.y=tracks(temp,2);
particle{i}.rsquared = (particle{i}.x-particle{i}.x(1)) .^ 2 + (
   particle{i}.y-particle{i}.y(1)) .^ 2;
end

%find difference in the two particles
initialx=particle{2}.x(1)-particle{1}.x(1);
initialy=particle{2}.y(1)-particle{1}.y(1);
```

```
ddisplacement=(particle{2}.x-particle{1}.x-initialx).^2+(particle
    {2}.y-particle{1}.y-initialy).^2;

difftraj.x=particle{2}.x-particle{1}.x;
difftraj.y=particle{2}.y-particle{1}.y;
sumtraj.x=particle{2}.x+particle{1}.x;
sumtraj.y=particle{2}.y+particle{1}.y;

pixelsizesquared=(36)^2;  %nanometers

 dsquared_diff=[];
    dsquared_sum=[];
for j=1:180
   dsquared_diff=diff(difftraj.x(1:j:1800)).^2+diff(difftraj.y(1:j
       :1800)).^2;
   dsquared_sum=diff(sumtraj.x(1:j:1800)).^2+diff(sumtraj.y(1:j
       :1800)).^2;
msd_diff(j)=mean(dsquared_diff);
msd_sum(j)=mean(dsquared_sum);
end

t=linspace(0,18,180);
figure;
plot(t,msd_sum.*pixelsizesquared);
xlabel('\tau (s)');
ylabel('MSD (nm^2)');
hold on;
plot(t,msd_diff.*pixelsizesquared,'r');

% t=linspace(0,180,1800);
% plot(t, particle{1}.rsquared.*pixelsizesquared);
% hold on;
% plot(t, particle{2}.rsquared.*pixelsizesquared,'r');
% xlabel('Time(s)');
% ylabel('Meters^2');
% title('Particle displacement squared');
```

```
% figure;
% plot(t, ddisplacement(1:180).*pixelsizesquared);
% xlabel('Time(s)');
% ylabel('Meters^2');
% title('Square of the Difference in Displacement')
```

PSF

```

function []=PSF_3D(slices_num)
%%For acquiring 3D PSF

clear all;
close all;

% vid = videoinput('avtmatlabadaptor_r2010a',1,'
    Mono8_656x492_Binning_1x1');
vid = videoinput('hamamatsu', 1, 'MONO16_1920x1440');
%preview(vid)
%input('Press a key when in focus')
%closepreview

%slices_num = 500;
%thickness = 0.5; %in micrometers

src=getselectedsource(vid);
src.ExposureTimeMode = 'Manual';
src.ExposureTime = 1;  %psf
src.ContrastGain=0;
src.Gain = 0;
src.TriggerTimes=1;
vid.ReturnedColorspace='grayscale';
vid.FramesPerTrigger=1;
vid.LoggingMode='memory';
triggerconfig(vid, 'manual');  %'immediate', 'manual', 'hardware'
```

```
vid.TriggerRepeat=slices_num;
vid.ROIPosition = [1230 950 300 300];


frames = zeros(304,304,1,slices_num,'uint16');%if changing
    resolution, gotta modify the cam script too

%After image is in focus, move stage up by half the total # of
    slices.

%z_stage('up',32,slices_num/2);


start(vid)
pause(2);

for n = 1:slices_num
    n
    trigger(vid)
    pause(src.ExposureTime);
    frames(:,:,:,n) = getdata(vid,1,'uint16');
    %frames(:,:,:,n)=z_cam(vid);
    z_stage('down',32,1)
    pause(0.8)

end
     stop(vid)
delete(vid)
clear vid

frames2=squeeze(frames);
%
imwrite(frames2(:,:,1),'frames130425.tif');
for k = 2:slices_num
     imwrite(frames2(:,:,k),'frames130425.tif','writemode','append')
         ;
```

```
end

%Set step size to be at least the Nyquist limit for z-direction:
    http://www.svi.nl/NyquistCalculator
%Make sure not to overexpose image: check intensity histogram and
    that very few
%values are at max intensity.
%thingie = double(flump) ./ 4095.0;   %12-bit camera, make range
    from [0 1]
%[counts, intensities] = imhist(thingie);
%semilogy(intensities, counts)
%DeconvolutionLab in ImageJ to fit PSF
```

```
function z_stage(dir,step, numsteps)
%Stepper motor/NI-DAQ controller
% direction should be 'up' or 'down', step can be 1,2,4,8 or 16,
    distance is
% in micrometers and duration is in seconds

if strcmp('up',dir) == 1
    dir_val = 0;
elseif strcmp('down',dir) == 1
    dir_val = 1;
else error('invalid direction input')
end

dio = digitalio('nidaq','Dev2');

chanD = addline(dio,0:3,'out');

switch step
    case 1
        pval = [0 0 dir_val];
    case 2
        pval = [1 0 dir_val];
    case 4
```

```
23          pval = [0 0 dir_val];
24      case 8
25          pval = [1 1 dir_val];
26      case 16
27          pval = [1 1 dir_val];
28      case 32
29          pval = [0 1 dir_val];
30      otherwise
31          error('invalid step input')
32  end
33
34  for i=1:numsteps
35  putvalue(dio.Line(1:4),[pval 0]);
36  pause(0.05);
37  putvalue(dio.Line(1:4),[pval 1]);
38  pause(0.1);
39  putvalue(dio.Line(1:4),[pval 0]);
40  pause(0.05);
41  end
42  %putvalue(dio.Line(4),0);
43  %pause(0.05);
44  delete(dio)
45  clear dio
46
47  % AO = analogoutput('nidaq','Dev2');
48  % chanA = addchannel(AO,0);
49  %
50  % %SampleRate = 10000; %in samples per second
51  %
52  % duty = 50; % in %
53  % DpS = 2.5/step; %in micrometers
54  %
55  % step_num = round(distance/DpS);
56  %
57  % period = duration/step_num; %in seconds
58  %
```

```
% %set(AO,'SampleRate',SampleRate)
% set(AO,'TriggerType','Manual')
% ActualRate = get(AO,'SampleRate');
%
% data = spulse(period,duty,ActualRate,duration);
% numsamples=length(data);
% tic;
% for i=1:numsamples
% %putdata(AO,data)
% putsample(AO,data(i));
% %start(AO)
% %trigger(AO)
% end
% putsample(AO,0);
% samprate = numsamples/toc
% %wait(AO,(duration + 1))
% %delete(AO)
% clear AO
```

```
%psf=loadtifs('C:\Users\ranbel\Documents\Classes\STORM\psf\psf
    images\frames130425_sequence');

function []= registration(psf)
%subtract out DC value
B=reshape(psf,1,304*304*500);
B=B-min(B);
newpsf=reshape(B,304,304,500);
clear psf;

%%
nmin=1;
nmax=500;
%registeredstack=zeros(607,607, nmax-nmin+1);
%dx=zeros(1,nmax-nmin+1);
%dy=zeros(1,nmax-nmin+1);

```

```
for n=nmin:nmax
n
image1=newpsf(:,:,nmin);
image2=newpsf(:,:,n);

%lpf the image and find the maximum value in the image
h=fspecial('average', [6 6]);
image1_temp=conv2(double(image1),double(h));
image2_temp=conv2(double(image2),double(h));
image1filt=image1_temp(2:305,2:305);
image2filt=image2_temp(2:305,2:305);

[m,i]=max(image1filt(:));
[ypeak1,xpeak1]=ind2sub(size(image1filt),i(1));
image_cropped1=zeros(40,40);
image_cropped1=image1filt(ypeak1-20:ypeak1+19, xpeak1-20:xpeak1+19);
%imagesc(image1filt);


[m,i]=max(image2filt(:));
[ypeak2,xpeak2]=ind2sub(size(image2filt),i(1));
image_cropped2=zeros(40,40);
image_cropped2=image2filt(ypeak2-20:ypeak2+19, xpeak2-20:xpeak2+19);
%figure;
imagesc(image2filt);

%C=xcorr2(image_cropped1,image_cropped2);
%imagesc(C);
%[m,i]=max(C(:));
%[ypeakcorr,xpeakcorr]=ind2sub(size(C),i(1));
%if (ypeakcorr==40 && xpeakcorr==40)
    dy=ypeak2-ypeak1;
    dx=xpeak2-xpeak1;
%else
%    disp('here')
%end
```

```
53
54
55 %%align using calculate dx and dy.
56
57 registered1=zeros(607,607);
58 registered1(152:455,152:455)=image1filt;
59 %if n==1
60 %    registeredstack(:,:,1)=registered1;
61 %end
62
63
64 registered2=zeros(607,607);
65 registered2(152-dy+1:455-dy+1, 152-dx+1:455-dx+1)=image2filt;
66 %registeredstack(:,:,n-nmin+1)=registered2;
67
68
69
70 low=min(min(registered2));
71 high=max(max(registered2));
72 if n==nmin
73     imwrite(registered2/(high-low),'registeredstack.tiff');
74 else
75     imwrite(registered2/(high-low), 'registeredstack.tiff','
        writemode','append');
76 end
77
78 end
79
80 %%
81
82 fname='registeredstack.tiff';
83 info=imfinfo(fname);
84 num_images = numel(info);
85 sliced=zeros(num_images,607);
86 for k=1:num_images
87     A=imread(fname,k);
```

```
88      sliced(k,:)=A(367,:,1);
89      %imagesc(A);
90      %drawnow;
91  end
92
93  low=min(min(sliced));
94  high=max(max(sliced));
95  imwrite(sliced/(high-low),'sliced.tiff')
96
97  for k=1:nmax-nmin+1
98      imagesc(registeredstack(:,:,k));
99       drawnow; %  update display immediately
100 end
```

```
1   function out = Gaussian2D(Parameters, Xdata)
2
3   out = Parameters(1) * exp(-((Xdata(:,1) - Parameters(4)) ./ (2 *
        Parameters(2))) .^ 2) .* ...
4                         exp(-((Xdata(:,2) - Parameters(5)) ./ (2 *
                              Parameters(3))) .^ 2) + Parameters(6);
5
6   plot(Xdata(:,1), out);
7   hold on
8   plot(Xdata(:,2), out, 'r');
9   if(size(Xdata,2) >= 3)
10      plot(Xdata(:,1), Xdata(:,3), 'y');
11      plot(Xdata(:,2), Xdata(:,3), 'c');
12  end
13  hold off
14  drawnow
```

```
1   function bestFit = PSFin2D(DisplayImage)
2   close all
3   %ActualImage = imread('blinking6.tif');
4   %DisplayImage = imread('psf_original_cropped.tif');
5   figure(1);
6   %Point1Display = DisplayImage(833:845, 797:809);  %y, x
```

```
Point1Display = DisplayImage;
imagesc(Point1Display);
% Point1DisplayGray = rgb2gray(Point1Display);
Point1Float = im2double(Point1Display);
intensitydata=Point1Float(:);  %reshape matrices into vector for
    lsqcurvefit


xsize=size(Point1Float,1);
ysize=size(Point1Float,2);
[xposition, yposition]=meshgrid(1:ysize,1:xsize);
X(:,1)=xposition(:);
X(:,2)=yposition(:);
X(:,3)=intensitydata(:);
x0 = [1, xsize/2 ,1, ysize/2];

options=optimset('TolX',1e-6);

%gaussianfunc = @(parameter, xdata) parameter(1) * exp(-(xdata(:,1)
    - parameter(2)) .^ 2/(2*a(3)^2)).* exp(-(A(:,2)-a(4)).^2/(2*a(3)
    ^2));


%out = lsqcurvefit(gaussianfunc,x0,X, intensitydata,[],[],options)

bestFit = nlinfit(X, intensitydata, @Gaussian2D, [ max(intensitydata
    (:)), 1, 1, xsize/2, ysize/2, 0]);

step=0.2;
[xfine,yfine]=meshgrid(1:step:ysize,1:step:xsize);
clear Xfine;
Xfine(:,1)=xfine(:);
Xfine(:,2)=yfine(:);
Ifit=Gaussian2D(bestFit,Xfine); %fitted gaussian applied to a finer
    grid
Ifit=reshape(Ifit,[size(xfine,1),size(xfine,2)]);%gaussian reshaped
```

```
    as matrix
% plot
figure(3);
mesh(xfine, yfine, Ifit)
title('Fitted')
figure(4)
mesh(xposition,yposition,Point1Float);
title('Actual')
colormap('default');
```

6.3 Protocols

Microtubule Polyermization (adapted from Mitchison Lab)

Materials

BRB80, 5x
Taxol in DMSO, 10mM
Unlabeled tubulin, 5 mg/mL
Labeled tubulin, 1 mg/mL
GTP, 100mM
Imaging buffer (TN buffer, GluOx, BME), or just 1xPBS
DMSO

Protocol:

1a. Split unlabeled and labeled tubulin into 20 ul tubes.

1b. On ice, mix 2 mg/ml tubulin, 2mM GTP, and 1xBRB80 to get total volume 50 ul.

Unlabeled:labeled	*Labeled (mg)*	*Labeled (ul)*	*Unlabeled*	*Unlabeled (ul)*
4:1	0.02 mg	---	0.08 mg	16 ul
9:1	0.01 mg	10 ul	0.09 mg	18 ul
19:1	0.005 mg	5 ul	0.095 mg	19 ul

1c. Add 1 ul of 100mM GTP to get 2mM in 50ul.

1d. Add 2 ml water to 0.5 ml of 5xBRB80 to get 2.5ml of 1xBRB80. Store any extra.

Unlabeled:labeled	*GTP + Tubulin (ul)*	*1xBRB80 (ul)*
4:1	17 ul	33 ul
9:1	29 ul	21 ul
19:1	25 ul	25 ul

2a. Dilute 1.5 ul of 10 mM taxol with DMSO. Store any extra.

Concentration (mM)	*DMSO volume (ul)*	*Total volume (ul)*
0.015 mM	998.5 ul	1000 ul
0.15 mM	98.5 ul	100 ul
2 mM	6 ul	7.5 ul

2b. Warm to 37° C for 2 minutes. Add 0.5 ul of 0.015 mM taxol. Mix by pipeting up/down and incubate at 37° C for 5 minutes.

2c. Add 0.5 ul of 0.15 mM taxol. Mix by pipeting up/down with *cutoff tip* and incubate at 37° C for 5 minutes.

2d. Add 0.5 ul of 2 mM taxol. Mix by pipeting up/down with cutoff tip and incubate at 37° C for 15 minutes.

3. Dilute 1 ul of microtubule solution to 1:20 by adding 1 ml of imaging buffer (980 ul of TN Buffer, 10 ul of GluOx and 10 ul of BME)

4. Pipet 10 ul into coverslip flowcell. Store at 37° C in the dark.

Bibliography

[1] J.W. Lichtman and J.A. Conchello, Nat. Methods. 2: 910 (2005).

[2] E. Abbe, Mikroskop Anat. 9: 413-420 (1873).

[3] R. Erni, M.D. Rossell, C. Kisielowski, and U. Dahmen, Phys. Rev. Lett. 102: 4 (2009).

[4] W.P. Putnam and M.F. Yanik. Phys. Rev. A. 80(4): 040902 (2009).

[5] E.H. Synge, Philos, Mag. 6: 356 (1928).

[6] D.W. Pohl, W. Denk, and M. Lanz, Appl. Phys. Lett. 44: 651 (1984).

[7] A. Lewis, M. Isaacson, A. Harootunian, and A. Muray, Ultramicroscopy. 13: 227 (1984).

[8] S.W. Hell and J. Wichmann, Opt. Lett. 19: 780-782 (1994).

[9] M.G.L. Gustafsson, J. Microsc. 198: 82-87 (2000).

[10] M.J. Rust, M. Bates, X.W. Zhuang, Nat. Methods. 3: 793-795 (2006).

[11] E. Betzig, G.H. Patterson, R. Sougrat, O.W. Lindwasser, S. Olenych, J.S. Bonifacino, M.W. Davidson, J. Lippincott-Schwartz, and H.F. Hess, Science. 313: 1642-1645 (2006).

[12] S.T. Hess, T.P.K. Girirajan, and M.D. Mason, Biophy. J. 91: 4258-4272 (2006).

[13] B. Huang, H. Babcock, and X.W. Zhuang, Cell. 143: 1047-1058 (2010).

[14] Unknown Editor. Method of the Year 2008. Nat. Methods. 6(1): 1-1 (2009).

[15] M. Heilemann, S. Linde, M. Schuttpelz, R. Kasper, B. Seefeldt, A. Mukherjee, P. Tinnefeld, and M. Sauer. Angew. Chem. Int. Ed. 47: 6172-6176 (2008).

[16] M. Heilemann. Design of Single-Molecule Optical Devices: Unidirectional Photonic Wires and Digital Photoswitches. PhD Thesis, Bielefeld University (2005).

[17] C.A. Karatsuba. Integral Transforms and Special Functions. 1: 269-276 (1993).

[18] M. Hagen. http://commons.wikimedia.org/wiki/File:Airy_vs_gaus.svg (2008).

[19] N. Bobroff. Review of Scientific Instruments. 57: 1152-1157 (1986).

[20] R.E. Thompson, D.R. Larson, and W.W. Webb. Biophys. J. 82: 2775-2783 (2002).

[21] M. Saxton. Fundamental Concepts in Biophysics vol 1. 156 (2009).

[22] S. van de Linde, R. Kasper, M. Heilemann, and M. Sauer. Appl. Phys. B. 93: 725-731 (2008).

[23] D. Evanko. Nature Methods 6: 19-20 (2009).

[24] N. Panchuk-Voloshina, R.P. Haugland, J. Bishop-Stewardt, M.K.Bhalgat, P.J. Millard, F. Mao, W. Leung, and R.P. Haugland. J. Histochem. Cytochem. 47 (9): 1179-1188 (1999).

[25] A. Jablonski. Nature. 131: 839-840 (1933).

[26] G.T. Dempsey, J.C. Vaughan, K.H. Chen, M. Bates, and X. Zhuang. Nat. Methods. 8(12): 1027-1036 (2011).

[27] R. Henriques, C. Griffiths, E.H. Rego, and M.M. Mhlanga. Biopolymers. 95(5): 322-331 (2011).

[28] B.O. Leung and K.C. Chou. Appl. Spectrosc. 65(9): 967-980 (2011).

[29] A. Yildiz, J.N. Forkey, S.A. McKinney, T. Ha, Y.E. Goldman, and P.R. Selvin. Science. 300(5628): 2061-2065 (2003).

[30] H. Shroff, C.G. Galbraith, J.A. Galbraith, and E. Betzig. Nat. Methods. 5: 417-423 (2008).

[31] K.I. Mortenson, L.S. Churchman, J.A. Spudich, and H. Flyvbjerg. Nat. Methods. 7, 377 (2010).

[32] B. Huang, M. Bates, and X. Zhuang. Annu. Rev. Biochem. 78, 993 (2009).

[33] K. Xu, H.P. Babcock, and X. Zhuang. 9(2): 185-188(2012).

[34] R. Widenhorn, M.M. Blouke, A. Weber, A. Rest, and E. Bodegom. Proc. SPIE 4669, Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications III, 193 (2002).

[35] C.E. Aitken, R.A. Marshall, and J.D. Puglisi. Biophys. J. 94, 1826-1835 (2008).

[36] Adsorption to Microspheres. TechNote 204. Bangs Laboratories, Inc. (1999).

[37] G. Olyslaegers and J.P. Verbelen. Biology of the Cell. 90 (3), 282 (1998).

[38] I. Semenova and V. Rodionov. Methods in Molecular Medicine. 137: 93-102 (2007).

[39] S.A. Sanchez, J.E. Brunet, D.M. Jameson, R. Lagos, and O. Monasterio. Protein Science. 13: 81-88 (2004).

[40] C. Gell, V. Bormuth, G.J. Brouhard, D.N. Cohen, S. Diez, C.T. Friel, J. Helenius, B. Nitzsche, H. Petzold, J. Ribbe, E. Schaffer, J.H. Stear, A. Trushko, V. Varga, P.O. Widlund, M. Zanic, and J. Howard. Methods in Cell Biology. 95: 221-245 (2010).

[41] C. Gell, M. Berndt, J. Enderlein, S. Diez. Journal of Microscopy. 234: 38-46 (2009).

[42] W.M. Bates, B. Huang, G.T. Dempsey, and X. Zhuang. Science. 317 (5845): 1749-1753 (2007).

[43] "Laboratory Technique: Spatial Filter." `http://cem01.ucsd.edu/~vitaliy/courses/ece182/182-06_files/SpatialFiltering.pdf`.

[44] H. Kirshner, D. Sage, and M. Unser. Journal of Microscopy. 249(1): 13-25 (2013).

[45] M. Born and E. Wolf. Principles of Optics, 7th ed. Cambridge University Press, Chapter 8.5: 436-445 (2003).

[46] S. Frisken-Gibson and F. Lanni. J Opt Soc Am A. 6(9):1357-67 (1989).

[47] B. Richards and E. Wolf. Proc. R. Soc. Lond. A. 253(1274):358-379 (1959).

[48] W. Press, B.P. Flannery, S.A. Reukolsky, and W.T. Vetterling. Numerical Recipes in C: The Art of Scientific Computing, 2nd ed. (1992).

[49] R.W. Cole, T. Jinadasa, and C.M. Brown. Nature Protocols. 6(12): 1929-1941 (2011).

[50] R. Henriques, M. Leek, E.F. Fornasiero, F. Vatorta, C. Zimmer, and M.M. Mhlanga. Nature Methods. 7(5): 339-340 (2010).

[51] S. Wolter, M. Schuttpelz, M. Tscherepanow, S. Van de Linde, M. Heilemann, and M. Sauer. Journal of Microscopy. 237(1): 12-22 (2009).

[52] S. Wolter. An Accurate and Efficient Algorithm for Real-Time Localisation of Photoswitchable Fluorophores. Diploma Thesis, Universitat Bielefeld (2009).

[53] S. Wolter, A. Loscherberger, T. Holm, S. Aufmkolk,M. Dabauvalle, S. van de Linde, and M. Sauer. Nat. Methods. 9: 1040-1041 (2012).

[54] C. W. Quanmen, A.C. Richardson, J. Haase, B.D. Harrison, R.M. Taylor, K.S. Bloom. "FluoroSim: A Visual Problem-Solving Environment for Fluorescence Microscopy. Proceedings of the Eurographics Workshop in Visual Computing for Biomedicine. Delft, the Netherlands. Oct. 67 (2008).

[55] A. Kishino and T. Yanagida. Nature. 344: 73-76 (1988).

[56] S. Reck-Peterson. "Imaging Single Molecules Using Total Internal Reflection Fluorescence Microscopy (TIRFM)." CSH Protocols (2010).