LUND UNIVERSITY
Faculty of Science

Multiple-Emitter Super-resolution Imaging using the Alternating Descent Conditional Gradient Method

Dolev Illouz

Thesis submitted for the degree of Bachelor of Science
Project credits: 15 ECTS

Supervised by Tobias Ambjörnsson

Computational Biology and Biological Physics
Department of Astronomy and Theoretical Physics
June, 2020

Contents

Popular Summary

Through the Looking Glass: Beyond Physics

Plants, viruses, “the birds and the bees” all function due to sophisticated bio-molecular machinery, whose cumulative and simultaneous action results in what we call *life*.

Miraculously, an organism’s DNA, or RNA, contains the blueprints for all the little bits and pieces of bio-molecular machinery necessary to sustain life. If we could read such blueprints, we could determine an organism’s ancestry, create treatment schemes for hereditary diseases, or even identify an organism’s mode of infection and consequently deduce ways to stop its spread. These blueprints are written in a language that consists of four letters separated from one another by approximately a third of a nanometer. However, this presents a problem as reading such tightly spaced “letters” requires a level of precision that is unattainable, per the inescapable laws of physics, which state that in a microscope, the resolution is limited by the wavelength of light observed and the numerical aperture of the lens system; this is known as Abbe’s diffraction limit.

Scientists have worked around the aforementioned problem by using next-generation sequencing (NGS) to determine the genes present in a sample. Such a process is quite expensive and time-consuming. Furthermore, the generated DNA sequences are short snippets of the sequenced gene. Hence, a tremendous amount of processing power is needed to piece these snippets together. Generally, this results in imperfect genome reconstructions, which may stall and even prevent the development of life-saving treatments. Thus, complementary methods are crucial in further improving genome reconstructions. One such method is optical gene mapping, where pictures of the gene labeled using fluorescent proteins at commonly appearing DNA regions provide a map for the final genome assembly. However, the Abbe diffraction limit prevents a detailed skeleton, as the light sources in the optical maps blend, resulting in the individual sources being indistinguishable from one another.

Fortunately, novel solutions that aim to defy the Abbe diffraction limit have been proposed. Among these solutions, the alternating descent conjugate gradient method, ADCG for short, has proven to be a state-of-the-art tool for resolving single-molecule images containing fluorescent dots. Using ADCG, highly detailed maps can be created in which the light sources, or fluorophores, are localized with precision in the tens of nanometers. These maps can then be used to rapidly identify and characterize bacterial strains in an economical and automatable manner, allowing scientists to react to mutations, consequently saving lives.

A typical single-molecule image consists of a DNA segment that has been chemically labeled and imaged using a microscope. Unfortunately for such images, Professor Abbes’ law seemingly holds, as finding the individual segments——which are shaped like dots——seems impossible. Alas, ADCG proposes a relatively simple solution. First, it assumes that the image comprises the minimum number of dots possible; in the context of information theory, this makes its predictions "sharp." Consequently, ADCG sequentially constructs synthetic fluorescent dots and moves them within the image until it effectively reconstructs the original image. This is done by minimizing the difference between the real and ADCG-generated images.

Using ADCG, the molecular motion of life can be captured and viewed at "super-resolution" that surpasses the laws of physics, which lends itself to assisting and, at times, replacing NGS and other sequencing methods as a fast and cheap alternative. Hence, developing an updated software library featuring additional functionalities in a modern computing language will expedite scientific discovery in molecular biology and nano-biophysics.

Acknowledgements

First and foremost, I would like to thank my family, Michael & Mimi Illouz, Sienna De L'Orpaz, Chloe Carter, and Sean Carter. Your endless support, encouragement, and sacrifice have allowed the continued pursuit of my scientific aspirations. I would also like to thank Rahul and Brandon, who believed in me when I doubted myself and never failed to put a smile on my face. Musti, Yasser, and Anton, I cannot thank you enough for the hours spent studying together and unknowingly motivational talks, which have been a source of strength.

I sincerely thank Tobias Ambjörnsson for hours spent discussing every detail of this thesis. Without your invaluable guidance, patience, and expertise, I genuinely believe I would've been lost; thank you so much for enriching my academic experience. I am also grateful to Carl Troein for his constructive feedback, valuable insights and for serving as a reviewer for my thesis.

Lastly, I would like to thank Jonathan Jeffet of NanoBioPhotonix Lab at Tel-Aviv University for providing the experimental data used in my thesis and Westerlund's BioNanoFluidics Lab at Chalmers University for their support in explaining the experimental setups used to generate such datasets.

Part I

Research context

1 Introduction

Outside of the Planckian realm, space-time can be considered continuous. However, our analytical tools capture and quantize input signals, resulting in the loss of information. Consequently, the detector may obscure a simple question such as, *'How many sources are present in a single input signal?'*. Such questions are known as "inverse problems," and at times, the laws of physics obfuscate the answer to such questions.

One such field in which inverse problems commonly occur is that of single-molecule imaging (SMI), where molecules are chemically stained and subsequently imaged under a microscope—thereby allowing the creation of optical DNA maps that aim to serve as a barcode of sorts, as the chemicals bind to specific molecules, such as short DNA or RNA sequences. However, the resulting images are diffraction-limited for sufficiently close molecules, and the fluorescent stains are optically indistinguishable. To resolve these molecules, an assumption, based on the principle of parsimony, is made: the input signal is simple, with low information content.[7] Under ideal conditions, this imposes sufficiently many constraints to super-resolve (SR) such images. In a computational implementation, the alternating descent conditional gradient method (ADCG) has proven itself to be the state-of-the-art 2D SR-technique in an analysis done by Sage *et al.*[6]. Despite these results, ADCG has yet to be utilized to map DNA optically.

Additionally, Sage *et al.* did not utilize pre-processing and post-processing techniques in their analysis. These methods greatly aid in modeling complex point spread functions (PSF) or analyzing high-emitter-density data, generally at the expense of robustness.[6] Nevertheless, post-processing steps such as those proposed by Jeffet *et al*[4] that leverage the intrinsic photo-bleaching of chemical stains over time to enhance localization, do not impact the algorithm's robustness. Similarly, post-processing the data to link single molecules temporally in the form of kymographs, which trace the motion over time of the molecules, as proposed by Jaqaman *et al.*[3] would exclude unlikely candidates thereby improving the localization without impacting the robustness of the localization algorithm.

While the ADCG, particle tracking, and pre-processing algorithms exist individually, they are mostly outdated scripts and lack a cohesive and robust package in a standard computing language. Therefore, to benefit the optical DNA mapping community, a Python class was developed. This class uses ADCG to localize fluorescent dots and contains modules for de-noising, the production of kymographs, and further localization routines that leverage photo-bleaching. The class uses the `NLopt` and `scipy.Optimize` libraries, which allows for swapping out any of the ADCG sub-routines in favor of an appropriate optimization algorithm contained in said libraries without loss of functionality.

In this thesis, the mathematical formalism of super-resolution and its explicit formulation in ADCG using the optimal SMI-specific sub-routines are described and characterized. Initially, a brief theoretical framework of single-molecule fluorescence is introduced alongside the functions to be optimized using ADCG. Subsequently, ADCG is described in terms of a modeling approach, and the pre-processing and post-processing methods used to improve fluorophore localizations are explained. Thereafter, the performance of ADCG and its time scaling as a function of input signal parameters are presented on synthetic and experimental data sets used to create optical DNA maps. Finally, a quantitative description of the algorithm parameters is presented.

2 Theory

2.1 Single Molecule Fluorescence

To characterize the input signals, it is imperative first to describe the physical phenomenon that generates these signals. Commonly, organic dye molecules, or fluorophores, are used for SMI, where the dye is chemically attached to the molecule of interest. These dye molecules have a continuous energy structure due to the interactions between the dye and its solvent. When the electrons from the lowest sub-level of the ground state are photo-excited up to the excited singlet band, de-excitation primarily occurs using either one of two methods, shown in figure 1. The first is phosphorescence, where the electron crosses over from the excited singlet state to the adjacent triplet state that is present in all organic dye molecules. The second is the radiative decay, or fluorescence, generally from the lowest sub-level of the excited state. Decay to the lowest sub-level of the excited state occurs due to internal conversion or vibrational relaxation. Hence, organic dyes exhibit what is known as a 'Stokes shift,' where the emitted photon is at lower energy relative to the absorbed photon. However, transitions from higher sub-levels may occur, which results in shorter wavelengths of the fluorescent photons. Nonetheless, it is extremely likely that the excited electrons will undergo some relaxation process within the excited energy band before radiatively decaying.[9]

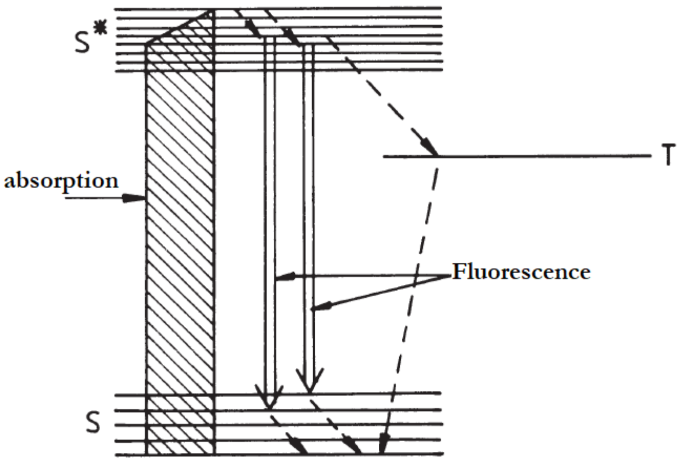

Figure 1: The vibrational energy levels of a dye where S is the ground state, S^* is the excited singlet state, T is the triplet state.[9]

The process of exciting the dye can result in the photobleaching of the fluorophore, which occurs due to the breaking of bonds over time, or covalent modification of the fluorophore that eventually results in the fluorophore being unable to fluoresce. This presents an added problem when it comes to creating super-resolution images and kymographs as the signals can 'die' out, in a stochastic manner.

2.2 Diffraction Limited Images

According to the Abbe diffraction limit, the shortest distance at which two objects are distinguishable in the image generated by a microscope, also known as the maximal resolution, d, of optical images is [5]

$$d = \frac{\lambda}{2NA}, \tag{2.1}$$

where λ is the emission wavelength of the fluorophore, and NA is the numerical aperture of an imaging microscope, which describes the maximal angle of light that the imaging lens can capture. Hence, maximizing resolution involves using short wavelengths and lenses of high numerical apertures. However, since the fluorophores are attached to biological structures, reducing the wavelength to resolve nanometer-spaced fluorophores is not viable as it would rapidly destroy the samples. Consequently, biological images are diffraction-limited, and as such, super-resolution algorithms aim to remove the blur induced by diffraction in addition to the effects of pixelization and noise.[7]

Point Spread Function

Since most of the changes to the input signals arise from the imaging system, it is important to characterize the process through which these effects occur.

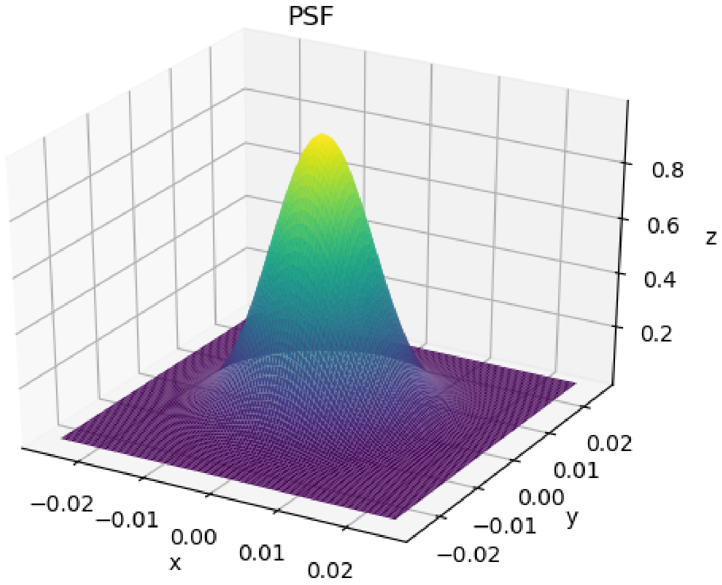

Figure 2: The intensity of a 2D Gaussian PSF as a function of space.

When imaging a single-point source, the resulting image is altered due to photon dispersion and the microscope's blurring of the source location. Mathematically, this blurring can be represented as the imaging system's point spread function (PSF), as shown in Figure 2.

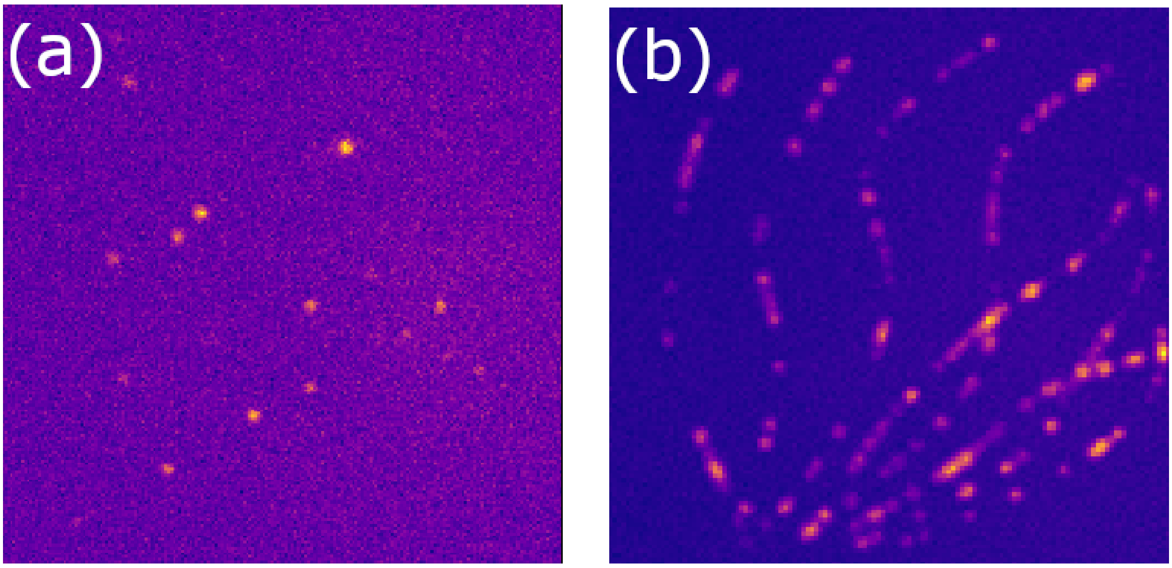

Figure 3: (*a*) A single frame from the 3 LS data set[2], which consists of simulated helicoidal tubes. (*b*) A single frame from the 1 HD data set[2], which consists of simulated tubulins of various diameters.

The convolution of the PSF and the sum of Dirac delta functions at the positions of the emitters results in an image, as shown in figure 3. If the point sources are separated by a distance less than d, the sources become indistinguishable in the resultant image, as shown in figure 3 (b). Mathematically, a convolution is a linear operator and an image is simply a function. Therefore, the image is the noisy measurement of a few weighted sources [1]

$$y(s) = \sum_{i=1}^{M} \omega_i \psi(p, r_i) + \nu, \tag{2.2}$$

where y is the image at pixel p, r_i is the spatial position of the point-source, ω_i is the intensity of the i^{th} source during the exposure time, M is the total number of sources, ψ is the PSF and ν is the additive background noise.

If the number of "empty" pixels is greater than twice the number of sources, and the PSF satisfies the conditions detailed in [7], then a unique total intensity exists.[1]

[1] These conditions are positivity, independence, and forming a T-system, but are of no practical consequence to ADCG as the PSF used, a Gaussian, intrinsically satisfies them.

Part II

Methodology

3 Method

From equation 2.2, it is possible to roughly infer that reconstructing a noise-free super-resolution image would amount to deconvolving the PSF of the imaging system used and subsequently estimating the intensities of the point sources.

3.1 Single Molecule Imaging as a Sparse Inverse Problem

By considering the fluorophores in a multi-emitter single-molecule image as point sources, the task of finding the number of sources, their position, and intensities present in an image can be formulated as a sparse inverse problem. The task of deducing the position and number of sources requires estimating the intensities of the fluorophores, which necessitates a mathematical formulation of their total intensity, μ, as [1]

$$\mu = \sum_{i=1}^{M} \omega_i \delta(r_i), \tag{3.3}$$

where ω_i is the intensity of the i^{th} fluorophore once the mean background noise has been subtracted, and $\delta(r_i)$ is a Dirac delta function centered at the position of the fluorophore, thereby satisfying the assumption that the fluorophore is a point source of light. However, labeling each pixel as an emitter can artificially decrease the loss. Therefore, the ADCG algorithm imposes a constraint on the maximal intensity, τ, which ensures sparsity in the number of fluorophores. Hence, τ controls the number of fluorophores that can exist in a single image and is, as such, greater than zero.

To evaluate the difference between the image and the super-resolution image produced by the ADCG algorithm, a convex loss function, ℓ, is used. In this thesis, ℓ is the Euclidean norm, as the images can be expressed as vectors. Thus, the sparse inverse problem can be formulated as the following optimization problem

$$\begin{aligned} \underset{p \,\in\, n}{\text{minimize}} \quad & \sqrt{\sum_{j=1}^{n} \left(\bar{y}_{sr}^{(j)} - (\bar{y}^{(j)} - \nu) \right)^2} = \ell, \\ \text{s.t.} \quad & |\mu| \leq \tau \end{aligned} \tag{3.4}$$

where j is the pixel index of the flattened image, n is the total number of pixels, ℓ is the loss function, y_{sr} is the super-resolution reconstruction of the image created by ADCG and $\bar{y}^{(j)}$ is the intensity of the experimental image at pixel j. This problem can be solved using ADCG.

3.2 Alternating Descent Conditional Gradient Method

Given equation 2.2, the noise is assumed to be statistically identical across all pixels. As such, the mean noise is determined by roughly segmenting areas containing the fluorophores and excluding them from the calculation; the specific methods used for each data set are detailed in appendix C sections 1 and 2. Following that, the mean noise was subtracted from each image within the dataset. Furthermore, the localization PSF used was created using equations 3.7 and 3.8, and the images were converted to a floating point format. A rough summary of the ADCG algorithm used in this thesis, a visual representation of which is shown in figure 5, is detailed in appendix B section 1, and its steps are referenced in the following explanation below.

To solve the problem presented in equation 3.4, ADCG initially grids the image into a $G \times G$ grid, where G is the total number of desired grid points. The resulting gridded image, Ω, contains more grid points relative to the total number of pixels, such that ADCG can localize the fluorophores at a sub-pixel accuracy. An arbitrary image is generated, and the loss is computed, as detailed in step 1.(a), according to equation 3.4. Subsequently, in step 1.(b), ADCG computes the gradient of ℓ, equation 3.4, across the gridded image,

$$\nabla \ell = \frac{\bar{y}^{(s)}_{sr} - (\bar{y}^{(s)} - \nu)}{\sqrt{\sum_{s=1}^{n} (\bar{y}^{(s)}_{sr} - (\bar{y}^{(s)} - \nu))^2}} \tag{3.5}$$

where s is the grid point index. In the grid representation, the image becomes a function of the grid point g and the spatial position of the fluorophore on the grid, r_g. Then the point in the gridded image, g_s^*, that minimizes the directional derivative of the unweighted projection of the pixelated PSF onto the grid, Ψ_G, along $\nabla \ell$ is computed,

$$\underset{g_s^* \in \Omega}{\arg\min} \, \langle \Psi_G, \nabla \ell \rangle = {}^{h}\Psi_G^T \cdot \nabla \ell \cdot {}^{w}\Psi_G \tag{3.6}$$

where $\langle ... \rangle$ represents the inner product, h denotes the PSF component along the height of the image, and w denotes the PSF component along the image's width. In effect, the point g_s^* that satisfies equation 3.6 corresponds to the brightest point in the image space Ω. In this thesis, Ψ_G is chosen to be a 1D Gaussian, as it intrinsically satisfies the conditions detailed in Schiebinger *et al.*, which ensures a unique total intensity exists, and the PSF can be resolved at arbitrary separations from $2M + 1$ observations. However, this assumes that there are at least $2M$ background pixels.[7]

The PSF is generated according to,

$${}^{a}\Psi_G(g) = \frac{1}{\sqrt{2\pi\sigma_a}} e^{-\frac{(g_a - s_{g,a})^2}{2\sigma_a^2}} \tag{3.7}$$

where a is an arbitrary component of the PSF, g_a is the grid coordinates along the component axis, $s_{g,a}$ is the current location, grid-wise, and σ_a is the standard deviation of the arbitrary Gaussian component, which in the case of diffraction-limited fluorophores can be approximated as,

$$\sigma_a = \frac{d}{\sqrt{2\ln(2)}} \tag{3.8}$$

where d is the maximal resolution shown in equation 2.1. Assuming the imaging system is free of astigmatism, the uneven blurring of the fluorophore in the image. Then, the image's standard deviation along the height, σ_h, and width, σ_w, are equal. Thus, the 2D PSF is generated by, ${}^{h}\Psi_G^T \cdot {}^{w}\Psi_G$ and denoted as Ψ_G.

Subsequently, the PSF, shown in figure 2, is moved around the local neighborhood of r_s to find the optimal position in the Ω, shown in figure 4. This is achieved by solving the left-hand side of equation 3.6 and corresponds to step 1.(c) in the rough summary. Once its position is set, the local descent routine is used, where the intensities of all localized PSFs are scaled to match the peaks, seen in figure 4. This process is repeated k_L times within each iteration of ADCG, with G, τ, ϵ, and k_L values chosen for the specific data set used, as described in section 3 of this thesis.

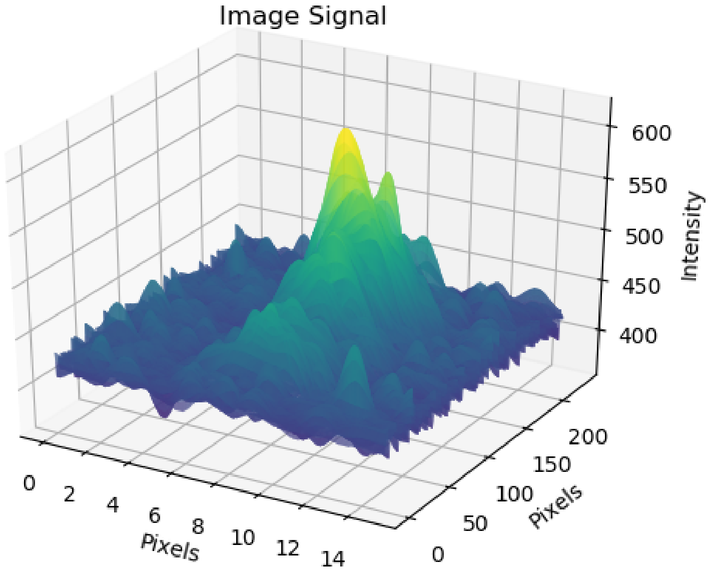

Figure 4: The intensity of an AF647 stained DNA segment as a function of space. By moving PSFs over this space, optimal locations that minimize a loss function can be found.

To calculate the weights of the current i^{th} proposed PSF location, step 1(d).ii, the weights of the current set of PSFs, $\{\Psi_G^{(i)}|i > 0\}$, are computed[1]

$$\begin{aligned} \text{arg min} \quad & \ell\left(\sum_{r_i \in S_n}^{M} \mu(\{r\})\Psi(r_i) - y\right), \\ \text{s.t.} \quad & |\mu| > 0,\ |\mu(S_n)| \leq \tau \\ & \mu(S_n^c) = 0 \end{aligned} \tag{3.9}$$

where $\mu(\{r\})\Psi(r_i)$ is the super-resolution image of a fluorophore localized at the set r of all the emitter positions, y is the unflattened experimental image, S_n is the set of all PSF localizations, and S_n^c is all other pixels.[2] This process is simply scaling all of the fluorophore intensities in the super-resolved image to better match that of the original image while ensuring that the total intensity remains below τ. If any fluorophore is found to emit below a threshold, γ, it is removed from the super-resolved image, corresponding to step 1.(d).iii.

Subsequently, in step 1.(d).iv, the positions of all PSFs are re-evaluated using equation 3.6. This is possible, as the image is locally differentiable, meaning that the PSF can be smoothly moved. If no PSFs were removed prior and the difference between the positions on the grid is below a threshold, β, the algorithm stops. However, if the PSFs were moved, then the weights are locally re-computed[1]

$$\begin{aligned} \text{minimize} \quad & \ell\left(\sum_{i=1}^{M} \omega_i \psi(r_i) - y\right). \\ \text{s.t.} \quad & \sum_{i=1}^{M} \omega_i \leq \tau \\ & r_i \in \Omega \end{aligned} \tag{3.10}$$

As before, the i^{th} fluorophore for which $\omega_i \leq \gamma$ is removed from the super-resolution image. Within each iteration of ADCG, the local descent optimization is run k times unless the threshold set by β is met.

In each iteration of ADCG, an additional localization point is added or removed from the super-resolution image y_{sr}. The process stops when the criterion below is met[1]

$$\tau\,|\langle \Psi_G(r_n), \nabla \ell_n \rangle| - \langle y_{sr_n}, \nabla \ell_n \rangle < \epsilon, \tag{3.11}$$

where r_n is the optimized localization of the nth iteration of the ADCG algorithm. This step corresponds to 1(d).i, but can only be used after the first optimal weight has been computed. Intuitively, equation 3.11 is satisfied when the nth PSF localization can contain the maximal intensity of the image and have a loss which is sub-ϵ, where ϵ is a user-defined threshold. The result would look something like step 7 of figure 5, where the PSFs are scaled to match the intensity of the dot at the location of the dot and

[2] For the method used to solve this equation, refer to appendix A.

have no noise.

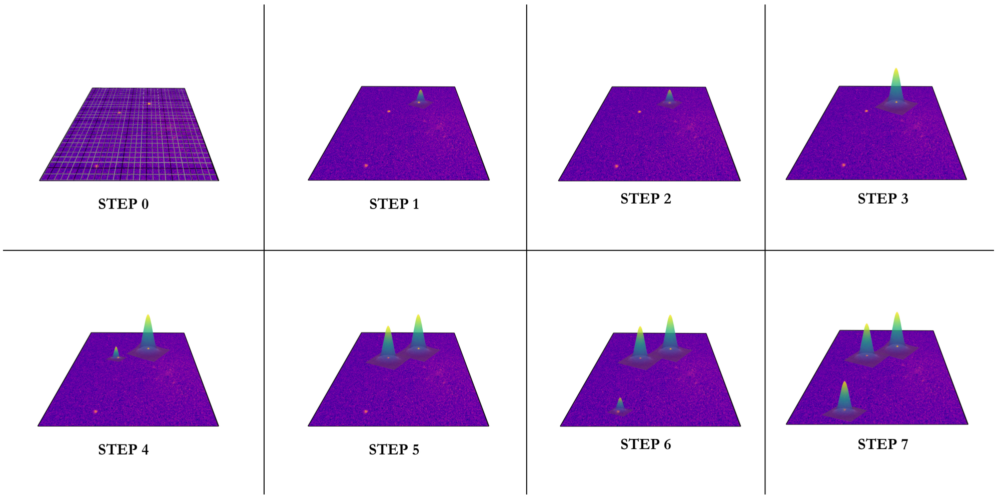

Figure 5: An artistic representation of the ADCG algorithm superimposed atop a frame from the 3 LS data set. Initially, the image is gridded, and the brightest pixel is used as the starting point for `NLOpt` , from which all subsequent weights are scaled. The process is then repeated for the remaining fluorophores.

3.3 Temporal Molecule Tracking

The final fluorophore localizations across each frame are static. However, for temporally linked data sets, tracking the PSFs through time can create optical bar codes or filter noise-induced localizations. The linked fluorophores generate a kymograph, an image that represents the transverse motion of the fluorophores as a function of time. However, identifying single molecules temporally, in the realm of SMI, necessitates the consideration of the fluorophores' inherent flickering. Furthermore, due to imperfect replication of the imaging systems' PSF, ADCG is not guaranteed to resolve sufficiently close fluorophores, resulting in an effect similar to flickering. Hence, a simple nearest-neighbor distance map is inaccurate. The temporal tracking algorithm used for the AF647 stained DNA sequences, adapted from the general algorithm detailed by Jaqaman *et al.*[3], is iterated thrice, front-to-back over the data, to optimize the bounded adaptive search radii, denoted by ζ. A rough summary of the algorithm used in this thesis, consisting of two parts, is presented in appendix B section 2, and its steps are referenced in the following explanation below.

Track Segmentation

Conceptually, the fluorophores were assigned "tracks" and temporally reassigned to these tracks in each subsequent frame unless they have photobleached. In the first step, for each frame, a matrix, shown in figure C.1 of appendix C section 2, of size $(M_t + M_{t+1}) \times (M_{t+1} + M_t)$ is constructed which is conceptually divided up into four segments. Thereafter, in steps 2 and 3, the center of mass in frames t and $t+1$ are computed, and a particle-to-track dictionary is created, which initially contains a one-to-one particle assignment.

Subsequently, in step 4, a Kalman filter is used to predict the future location of the particles from one frame to the next, and the squared distance between the prediction and the location in the next frame, $t+1$, is computed. Thereafter, in step 4.(a), the bounded adaptive particle search radius, ζ_p, is computed according to

$$\zeta_p = \max\left(\left\{ 3\sigma_i^{(t)} \right\}, \left\{ \frac{1}{2}\Delta\mathbf{x} \right\} \right), \tag{3.12}$$

where $\sigma_i^{(t)}$ is the standard deviation of each optimal particle assignment, $\Delta\mathbf{x}$ is the vector containing the distance between particle i and its nearest neighbor, and i is its index. Additionally, if the particle from frame t was assigned a track prior, its nearest neighbor distance was computed as the mean of past frames.

After that, tracks whose Kalman distance exceeds ζ^2 are pruned from the particle-to-track dictionary, and the remaining tracks' Kalman distances are inserted into the top left $M_t \times M_{t+1}$ section of the matrix, corresponding to step 4.(b). In each time frame, the bottom right $M_{t+1} \times M_{t+1}$ section is filled with the smallest squared distance, corresponding to step 5 in appendix B section 2.1.
The top right $M_t \times M_t$ and bottom left $M_{t+1} \times M_{t+1}$ sections are filled along their diagonal with a maximal link value, ε, which during the first step of the initial iteration, $t = 1$, is set as the 80^{th} percentile of all potential assignment costs. In subsequent time frames, ε becomes the maximal distance of the optimal solution set times an arbitrary scaling parameter slightly larger than 1.

Thereafter, in step 7, once the cost matrix has been fully constructed, it is inserted into a linear assignment problem (LAP) solver, where each row can be assigned to a single column at most. Each row in the top left section represents a fluorophore at time t, and each column represents a fluorophore at time $t+1$. Thus, the column index gets assigned the track number of their row assignment index. That is, fluorophores at $t+1$ get placed in the track of their assigned fluorophore from time t. For fluorophores at time t assigned to columns in the top left, their track is said to have ended, and for fluorophores at time $t+1$ that are assigned to rows in the bottom right, their tracks are said to have started. This amounts to adding the column index to the track list, $\{T_l | l > 0\}$, alongside a new track T_{l+1}, thereby updating the particle to track assignment matrix. Thus, in the end, each particle in every time frame is assigned to a track.

Track Optimization

Once the l tracks are generated, the flickering and thermal motion of the fluorophores may give rise to disjoint or globally suboptimal assignments. Hence, the tracks should be re-optimized using the intensity, temporal track length, and individual track drift over time; a rough summary is presented in appendix B section 2.2, and its steps are referenced in the following explanation below.

Initially, in step 1, a $3l \times 3l$ matrix that is conceptually segmented into nine different sections is created; see Jaqaman *et al.*[3] figure 1c for a schematic illustration. Thereafter, the squared distance of the start or end of track segment I and the midpoint of track segment J, δ_{IJ}^2, are computed and inserted into the upper left $l \times l$ section. The potential assignment costs corresponding to steps 3-4 are defined as

$$m_{IJ} = \begin{cases} \delta_{IJ}^2 \frac{A_J(t-1)}{A_I(t)+A_J(t)}, & \text{if } \frac{A_J(t-1)}{A_I(t)+A_J(t)} > 1 \\ \delta_{IJ}^2 \left(\frac{A_I(t)+A_J(t)}{A_J(t-1)}\right)^2, & \text{if } \frac{A_J(t-1)}{A_I(t)+A_J(t)} < 1 \end{cases} \tag{3.13}$$

where A is the intensity of a track segment at a given time.

$$s_{IJ} = \begin{cases} \delta_{IJ}^2 \frac{A_J(t)}{A_I(t-1)+A_J(t-1)}, & \text{if } \frac{A_J(t)}{A_I(t-1)+A_J(t-1)} > 1 \\ \delta_{IJ}^2 \left(\frac{A_I(t-1)+A_J(t-1)}{A_J(t)}\right)^2, & \text{if } \frac{A_J(t)}{A_I(t-1)+A_J(t-1)} < 1 \end{cases} \tag{3.14}$$

Equations 3.13 and 3.14 represent the merging and splitting costs, respectively. The rejection costs of merging and splitting, respectively, are

$$m'_J = \begin{cases} \bar{\delta_J}^2 \frac{A_J(t)}{A_J(t-1)}, & \text{if } \frac{A_J(t)}{A_J(t-1)} > 1 \\ \bar{\delta_J}^2 \left(\frac{A_J(t-1)}{A_J(t)}\right)^2, & \text{if } \frac{A_J(t-1)}{A_J(t)} < 1 \end{cases} \tag{3.15}$$

where $\bar{\delta_J}^2$ is the average frame-to-frame displacement in track segment J.

$$s'_J = \begin{cases} \bar{\delta_J}^2 \frac{A_J(t-1)}{A_J(t)}, & \text{if } \frac{A_J(t-1)}{A_J(t)} > 1 \\ \bar{\delta_J}^2 \left(\frac{A_J(t)}{A_J(t-1)}\right)^2, & \text{if } \frac{A_J(t-1)}{A_J(t)} < 1 \end{cases} \tag{3.16}$$

For steps 2-6, the top left $l \times l$ section is filled with δ_{IJ}^2, the top central $l \times l$ section is filled with m_{IJ} and the top right $l \times l$ section is filled with ε' along its diagonal, where ε' is the 90^{th} percentile of all potential assignment costs. The central left $l \times l$ section is filled with s_{IJ}, the central most $l \times l$ section is set to 0, and the central left $l \times l$ section is filled with s'_J. The bottom left $l \times l$ section is filled with ε' along its diagonal, and the bottom central $l \times l$ section is filled with m'_J.

Subsequently, in step 7, the costs for track linking associated with a separation distance that exceeds the bounded adaptive track search radius,$zeta_T = \max(\zeta_s, \zeta_e)$, are removed.

The track end and start search thresholds are computed by

$$\zeta_s = \begin{cases} \max\left(3\sigma_I^l\sqrt{\delta t}, \tfrac{1}{2}\Delta\mathbf{X_S}\right), & \text{if } \delta\mathbf{t} < t_p \\ \max\left(3\sigma_I^l\delta t)^{\frac{1}{10}}, \tfrac{1}{2}\Delta\mathbf{X_S}\right) & \text{if } \delta\mathbf{t} > t_p \end{cases} \tag{3.17}$$

where t_p is a user-defined time gap that scales the search radius, $\mathbf{X_S}$ is a vector containing the position of all the starts of the tracks, σ_I^l is the standard deviation of frame-to-frame displacements in track segment I, and l is its index.

$$\zeta_e = \begin{cases} \max\left(3\sigma_I^l\sqrt{\delta t}, \tfrac{1}{2}\Delta\mathbf{X_E}\right), & \text{if } \delta\mathbf{t} < t_p \\ \max\left((3\sigma_I^l\delta t)^{\frac{1}{10}}, \tfrac{1}{2}\Delta\mathbf{X_E}\right) & \text{if } \delta\mathbf{t} > t_p \end{cases} \tag{3.18}$$

where $\mathbf{X_E}$ is the vector containing the positions of all the ends of the tracks. Subsequently, tracks whose beginning and end have a temporal separation of less than Δt units of time are removed. Then, the bottom left $l \times l$ section is filled with the minimum δ_{IJ}^2 cost. Finally, the $3l \times 3l$ matrix is inserted into a LAP solver, and its assignments yield the optimal track configurations.

The track segmentation and subsequent optimization described above constitute a single iteration's "front" half. The bounded adaptive search radii are re-used and updated in the current iterations' subsequent "back" half, where $t_0 = t_f$.

4 Optimization routines, data sets, and parameter selection

Given the myriad of convex and non-convex optimization methods, each having its strengths and weaknesses, a simplistic testing routine was performed in which various non-linear and linear optimization methods were compared to one another. The non-linear optimization methods were implemented using the `NLopt` python library, and the linear methods using the `scipy.Optimize` python library.

4.1 Single Molecule Localization Microscopy Challenge Data Sets

The "Bundled Tubes Long Sequence" (BTLS) dataset and the "Bundled Tubes High Density" (BTHD) dataset from the 2013 Grand Challenge Localization Microscopy[2, 6] were downloaded alongside their ground-truth files. These data sets were generated using an emitter wavelength of 723 nm, an NA of 1.4, and a pixel size in the image plane of 100 nm. Subsequently, the relevant sub-routines were combined using threshold parameters that yielded similar localization accuracies. The testing was performed with a grid mesh size of 1000×1000, $\gamma = 0$, and $\tau = 3\times10^5$. For the BTLS dataset, ADCG was run for a maximum of 30 iterations, whereas for the BTHD set, ADCG was run for 1000 iterations, as the total number of fluorophores was much greater in the latter. The latter data set was used as a visual confirmation of the ability of ADCG to localize a high density of fluorophores accurately. However, this thesis will present no analysis, as the AF647 data is relatively dense.

4.2 Artificial Data sets

Once the best cost-to-performance routines were deduced, further testing was performed regarding the algorithm's performance as the input parameters scale; specifically, the time scaling as a function of the fluorophore density, the number of pixels, and grid points were evaluated. The former two were performed on the SMLM training datasets using a grid mesh of 5000×5000, a maximal ADCG iteration of 20, with an ϵ of 7×10^{-5} and a maximal IP iteration of 200. The latter was performed using an artificially generated square image containing a single Gaussian PSF centrally localized in the image, with a σ equal to 1% of the image height over a 5000×5000 grid. The computational time was the mean wall time spent optimizing the image over five runs, and the test was performed by restarting the kernel each time without any special care to the background processes running on the host machine.

4.3 AF647 Stained DNA Molecules

The data comes from an unpublished set of Optical DNA mapping experiments performed by Jonathan Jeffet at the NanoBioPhotonix Lab at Tel-Aviv University. In the experiment, DNA molecules of bacterial plasmid vectors were stained and inserted into an Irys chip containing 45 nanometer (nm) nanochannels. The data contains a YOYO-1 stained channel and an AF647 channel, which are hyperstacked in a .tiff file. Each field-of-view has 500 frames of 40 millisecond (ms) exposure per frame with a pixel size of 117 nm in the sample plane.

The parameters used in analyzing experimental data sets were determined by calculating the desired theoretical localization precision, the expected number of fluorophores in each frame, and the sum of the entire image array. Without such information, an initial optimization was performed on the first five images within a dataset to determine sensible parameter values roughly. The default values for the code is an $\epsilon = 1 \times 10^{-5}$, $\tau = 2 \times 10^{5}$, $\beta = 1 \times 10^{-7}$, $\gamma = 0$, and a maximal number of IP iterations of 200.

Using the default set of parameters, the AF647 stained DNA sequences were analyzed with a mesh grid of 5500, thereby allowing a theoretical localization precision of 5 nm. Additionally, since the vector contains 37 TCAG repeats, the maximal number of ADCG iterations was set to 50. Prior to each image being analyzed, the YOYO-1 stack was used to segment the DNA strand from the background, and the difference between the two stacks' backgrounds was used to calculate the mean noise. The PSF localizations were normalized relative to the center of mass of the DNA sequence, determined by the length of the YOYO-1 stack, and subsequently linked.

Part III

Numerical Results

5 Results

Based on the testing performed, the chosen routines were LD_MMA[8] for the non-linear optimization subroutines and interior-point (IP) for the linear optimization subroutines as they were the most cost-effective, in terms of their accuracy relative to the needed computational time.

5.1 Single Molecule Localization Microscopy Challenge (SMLM)

The SMLM data sets consist of fluorescent dots at various densities. From the entire set, two sets are presented here: the BTLS and BTHD data sets. The BTLS data set is an artificially generated data set representing bundles of 8 tubes that are 30 nm in diameter at low fluorophore density, and the BTHD data set is its high-density counterpart. A sample of the BTLS data set is shown in figure 6 (a). Both data sets are generated using a Gaussian PSF, with a pixel size of 100 nm in the sample plane.

Localization

Inserting the image in figure 6 (a) into the ADCG algorithm produces the image in figure 6 (b), which has an average localization error of 8.11 nm for a mesh grid of 500×500. When using finer meshes of 1000×1000 and 5000×5000, an average localization error of 6.80 nm and 6.82 nm are obtained, respectively. For reference, a single fluorophore represents a single α/β-tubulin heterodimer that is approximately 8 nm long. Thus indicating that ADCG is capable of accurately and precisely localizing the dots, even though the PSF used is not the same as the one used to generate the artificial dots. Furthermore, the variance in the localization between individual runs of the same images was on the order of 10^{-27}, pointing to the excellent robustness provided by the ADCG algorithm.

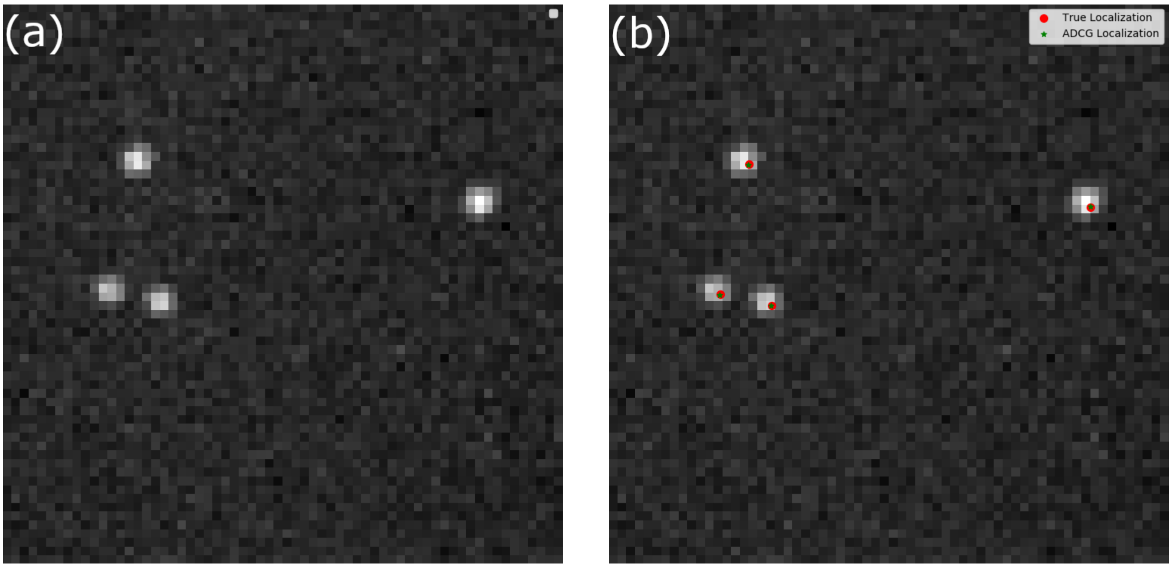

Figure 6: (*a*) A single frame from the Bundled Tubes Long Sequence data set consists of bundles of 8 tubes of 30 nm in diameter. (*b*) The fluorophore localizations produced using the ADCG algorithm (predicted localizations shown in green, and the true positions in red).

However, these results are potentially misleading, as the BTLS data set was artificially generated with Gaussian noise, in addition to the signal-to-noise ratio being quite high.

Performance & Scaling

Naturally, as the search space within the image increases with respect to either the number of points the algorithm needs to consider or the number of objects to be localized, the optimization time should increase.

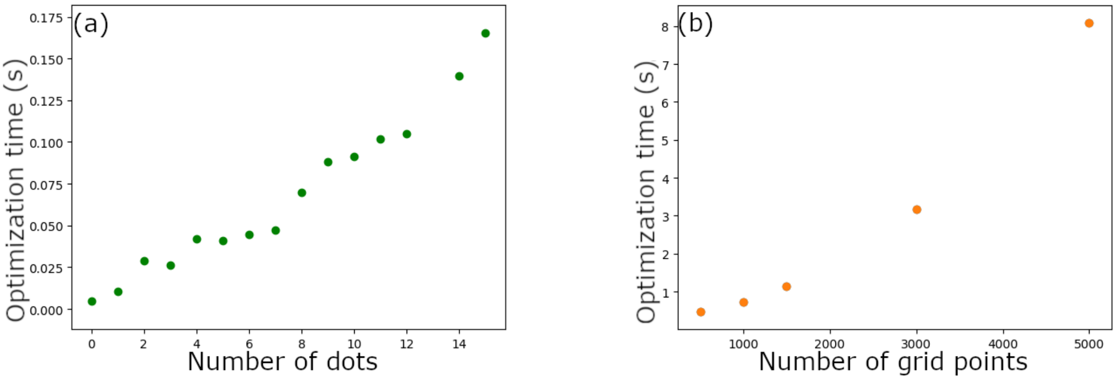

Figure 7: The average optimization time of an image from the BTLS data set: (*a*) as a function of fluorophore density. (*b*) as a function of the grid density.

In the absence of any fluorophores, the optimization time is non-zero, as one iteration of ADCG is necessary to deduce this matter. Furthermore, it is clear from figure 7 that increasing the number of grid points is a more expensive task than handling high fluorophore densities.

Intensity Analysis

To determine ADCG's robustness to noise, an analysis of its predicted localizations' signal-to-noise ratio (SNR) is necessary, which is done by segmenting the background and plasmid. In the absence of emitters, the background intensity is expected to be distinct from the normalized intensity of the emitters. Hence, a radius of 2σ, the average standard deviation of nearest neighbors across 50 frames, is used as a segmentation threshold. Additionally, the normalized intensity proposed by ADCG should match that of the normalized intensity of the image.

The background intensity, as shown in figure 8, is centered around zero, while the estimated intensity and predicted ADCG intensity line up fairly well, while remaining distinct from the noise.

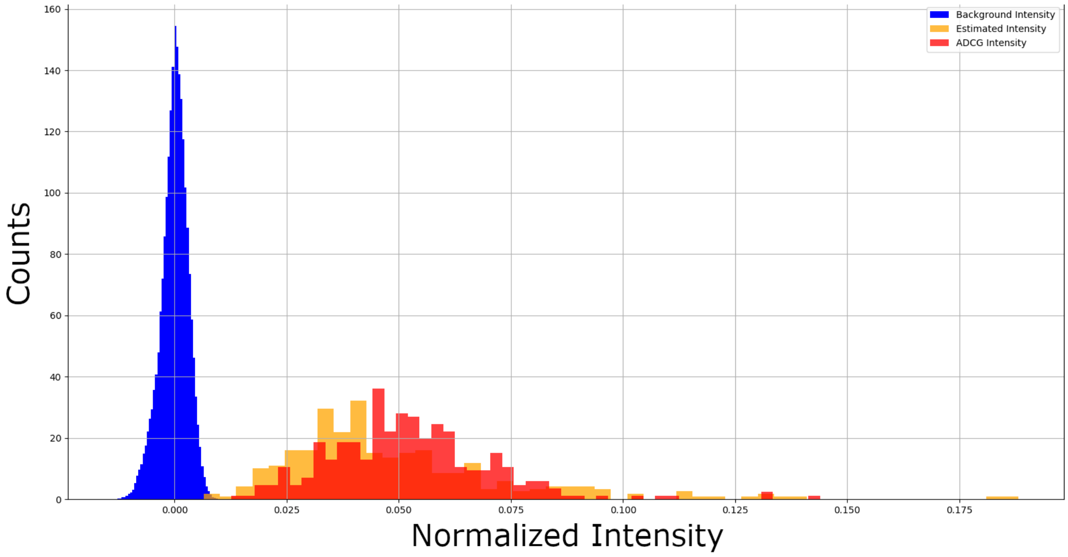

Figure 8: A normalized intensity histogram for a collection of 50 images from the BTLS dataset.

5.2 Artificial Data Sets

By increasing the total number of pixels present in a single image, the local optimization routine would have to be appropriately expanded as the number of candidate points increases due to the increased information content. Further gridding increases the parameter space to be searched, and therefore, the optimization time is expected to increase, which is clearly shown in figure 9.

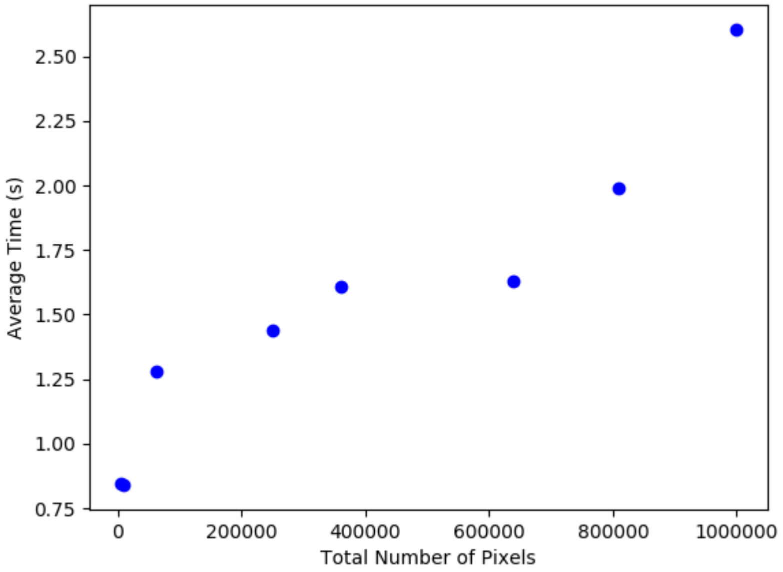

Figure 9: The average computational time as a function of the total number of pixels present in an image.

The computational time is the wall time spent optimizing the image, averaged over five runs. Even though the number of grid points used remains the same, it is clear that the computational time increases. The increase in time may be attributed to the increase in the standard deviation of the PSF used to generate the images, consequently increasing the number of candidate localization points.

5.3 AF647 Stained DNA Sequence

While ADCG performs tremendously well in regards to the artificial training data sets, the true test lies in the analysis of experimental data. The raw image of an AF647 stained DNA sequence is shown in figure 10, and the task of localizing all the dots' positions is difficult, as upon visual inspection, their positions are obfuscated by the noise present in the image.

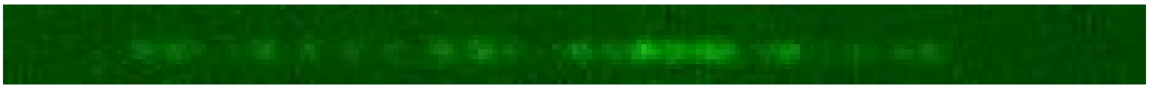

Figure 10: A single frame of an AF647 stained DNA sequence.

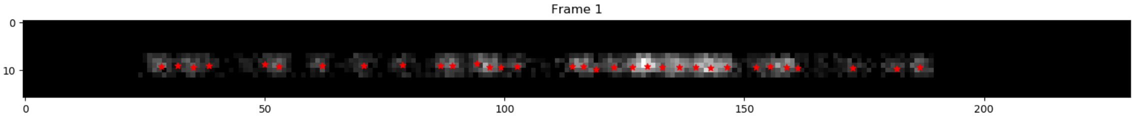

Figure 11: The fluorophore localization positions of an AF647 stained DNA sequence using ADCG.

When applying ADCG to every frame of an AF647 stained DNA sequence video using a 5000 × 5000 grid size, each frame takes roughly 7 seconds. As shown in figure 11, using the ADCG algorithm, the resulting localizations appear rather accurate. Better yet, the degree of accuracy becomes more apparent when an entire sequence of points is shown in the form of a kymograph in figure 12.

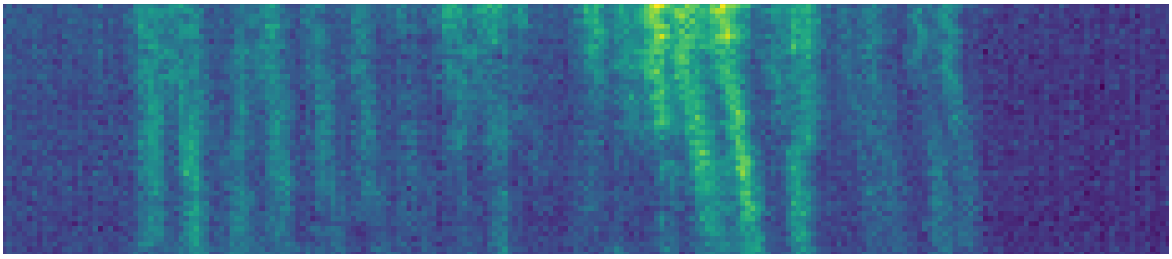

Figure 12: A 50 frame Kymograph of an AF647 stained DNA sequence movie.

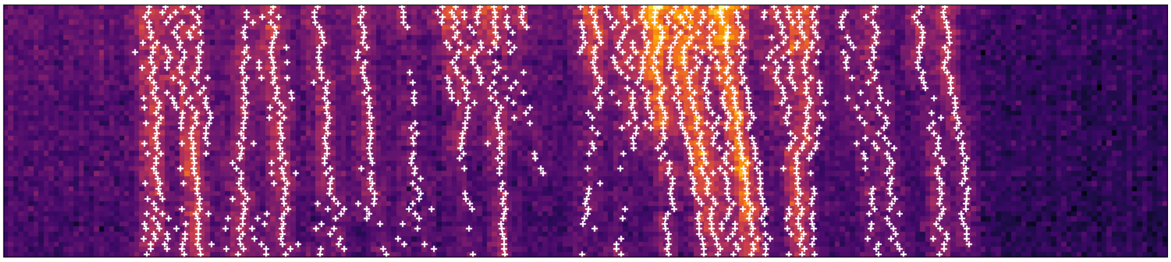

Figure 13: The proposed fluorophore localization across 50 frames.

Upon careful examination of the kymograph depicted in figure 13, it is apparent that the localizations, depicted as white crosses, could accurately generate tracks over time. Nonetheless, non-additive noise, which is not considered in equation 2.2, results in some noise being mislabeled as fluorescence, as illustrated in figure 11. The temporal molecule tracking routine addresses this issue by establishing links over time between fluorophores, effectively filtering out noise and retaining only the localizations assigned to tracks. Appendix 2 displays this in figure C.2.

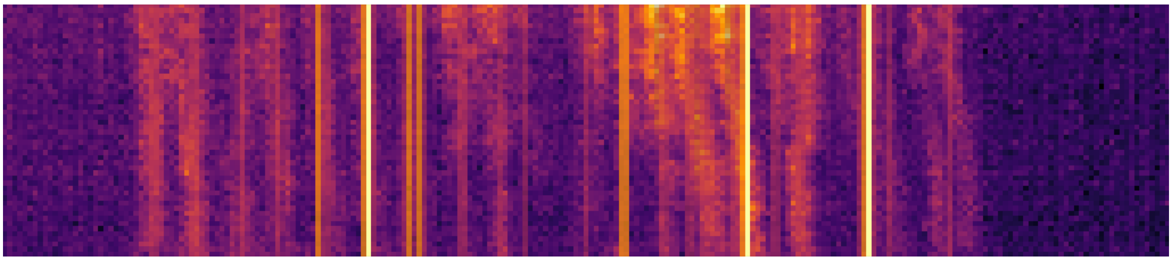

Figure 14: A DNA barcode overlayed on the AF647 labeled DNA sequence's raw kymograph that generated it. The DNA barcode was generated using the temporal tracking algorithm, which filtered out short tracks and provided the track's mean positions.

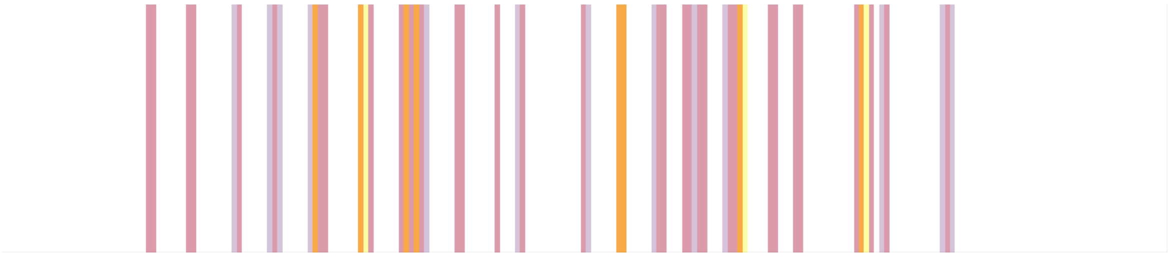

Figure 15: The optical DNA barcode generated as described in figure 14, by itself.

The generated tracks suffer from an average localization error of 300 nm due to thermal fluctuations. However, averaging the fluorophore's position over time, as detailed in Jeffet *et al.*, reduces the average thermal fluctuation localization error to approximately 150 nm, resulting in the optical barcode shown in figure 15. The barcode generated roughly matches the experimental data, as shown in figure 14, with the barcode positions aligning with the fluorescent tracks of the kymograph. However, it is entirely possible that some of the tracks seen in figure 12 are, in fact, noise, as quantitative analysis is not possible using the AF647 stained DNA sequence dataset beyond the following intensity analysis as the actual positions are unknown.

5.4 Intensity Analysis

Clearly, the localization performance on experimental data sets is worse as the SNR is lower when compared to the synthetic data sets. However, using the same method detailed in section 5.1, visualizing the efficiency at which ADCG avoids labeling noise is possible. As shown in figure 16, while the distance between the noise intensity and the PSF localizations is not as considerable as on the synthetic data set, there is still a distinct separation between the two, in addition to the predicted localizations intensity roughly matching the expected intensity obtained from segmenting the image.

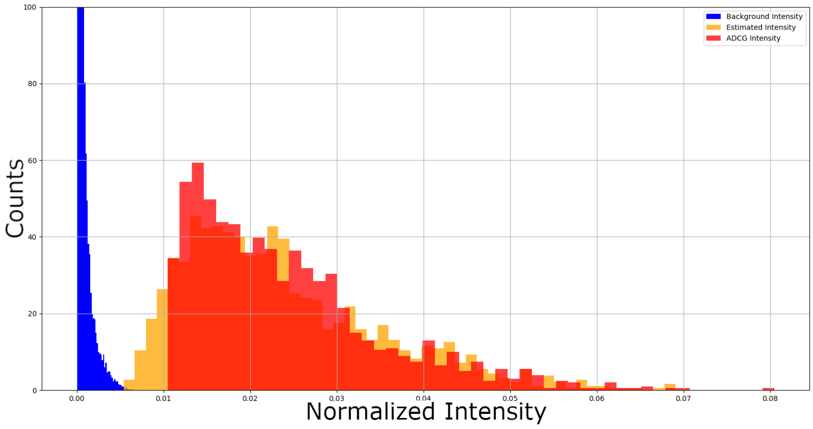

Figure 16: A normalized intensity histogram for a collection of 50 images from the NBP dataset.

Part IV

Conclusions

6 Conclusions

In conclusion, ADCG's localization on synthetic datasets is on the order of nanometers due to its high SNR. However, when working with experimental datasets that are typically noisier, the localization accuracy is on the order of tens of nanometers. Additionally, thermal fluctuations introduce uncertainties on the order of hundreds of nanometers for temporally linked data sets.

The reduction in accuracy on experimental datasets is likely a consequence of equation 2.2 being an inappropriate representation of the noise within the imaging system. Thus, the exact ν values required to localize all fluorophores present in an image while suppressing false labeling are highly sensitive in the model proposed in equation 2.2. Most likely, the CCD used introduces noise in a manner related to the intensity within each pixel. Furthermore, the confinement of the DNA sequences within nano-channels introduces surface effects to the emissions of the fluorophores; while unlikely, it is possible that the solvents in which the sequences are suspended lead to anisotropic emission.

Consequently, it is thought that the most significant barrier to accurate localization is due to the mischaracterization of the noise present in the image as purely additive. As such, further research into the effects of pre-processing methods that increase the signal-to-noise ratio and effectively eliminate non-additive noise sources would likely yield improvements in the localization accuracy and precision of ADCG. Alternatively, experimental research and parametrization of the imaging systems will likely provide the relevant modifications to equation 2.2, which could vastly improve the localization. Furthermore, using a synthetic temporally linked dataset would allow the parametrization of the temporal linking routines' accuracy in eliminating the non-additive noise, which is unquantified beyond an intensity analysis in this thesis.

Bibliography

[1] N. Boyd, G. Schiebinger, and B. Recht, *The alternating descent conditional gradient method for sparse inverse problems*, 2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), (2015).

[2] S. Holden and D. Sage, *Super-resolution fight club*, Nature Photonics, 10 (2016), p. 152–153.

[3] K. Jaqaman, D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S. L. Schmid, and G. Danuser, *Robust single-particle tracking in live-cell time-lapse sequences*, Nature Methods, 5 (2008), p. 695–702.

[4] J. Jeffet, A. Kobo, T. Su, A. Grunwald, O. Green, A. N. Nilsson, E. Eisenberg, T. Ambjörnsson, F. Westerlund, and E., Weinhold *et al.*, *Super-resolution genome mapping in silicon nanochannels*, ACS Nano, 10 (2016), p. 9823–9830.

[5] S. G. Lipson, H. Lipson, and D. S. Tannhauser, *Optical physics*, Cambridge Univ. Press, 2004.

[6] D. Sage, P. Thanh-An, H. Babcock, T. Lukes, T. Pengo, J. Chao, R. Velmurugan, A. Herbert, A. Agrawal, and S., Colabrese *et al.*, *Super-resolution fight club: A broad assessment of 2d & 3d single-molecule localization microscopy software*, Nature Methods, (2018).

[7] G. R. Schiebinger, *https://personal.broadinstitute.org/*, PhD thesis, University of California, Berkeley, 2016.

[8] K. Svanberg, *A class of globally convergent optimization methods based on conservative convex separable approximations*, SIAM Journal on Optimization, 12 (2002), p. 555–573.

[9] A. P. Thorne, U. Litzén, and S. Johansson, *Spectrophysics*, Springer, 1999.

Part V

Appendices

Appendix A

Optimizing Weights Using the Primal Dual Interior Point Method

The PSFs are computed separately in h and w and subsequently multiplied with one another to obtain the full PSF, with the fluorophore localization corresponding to average of the Gaussian. The PSF vectors are collected into an array and subsequently, their current weight is computed

$$K_h = \Psi(\{t_h\})^T \times \Psi(\{t_h\}), \tag{0.1}$$

$$K_w = \Psi(\{t_w\})^T \times \Psi(\{t_w\}). \tag{0.2}$$

The diagonal of these arrays corresponds to the intensity squared of the current PSF localizations and the total intensity of the PSFs,

$$K = K_x \odot K_y \tag{0.3}$$

where $\odot$ represents an element-wise multiplication operator. Similarly, the total intensity of the image at the location of the PSFs, $\mu(y)$, is

$$\mu(y) = \sum_{j \in J} (\psi(\{t_w\})^T_{i,n} \times y_{ij}) \odot \psi(\{t_h\})^T_{j,n}, \tag{0.4}$$

where n is the number of PSFs, j is the vertical pixel index within the image, and i is the horizontal pixel index in the image. This is formulated as an optimization problem, where the function to be minimized is,

$$f_{min} = \frac{1}{4n}\tau_{prop} \cdot K \times \frac{1}{2n}\tau_{prop} - \frac{1}{2n}\tau_{prop} \cdot \mu(y) \tag{0.5}$$

Where τ_{prop} is a vector containing newly proposed τ values that would better match the total intensity in the system.

Appendix B

Algorithm Outlines

1 ADCG Algorithm

Iterate k times, where $k > M$.

1. Grid the image such that the gridded image, Ω, contains a larger number of grid points relative to the total number of pixels.
    (a) Calculate the loss at the current iteration of ADCG,
    $\ell = \sqrt{\sum_{j=1}^{n} \left(\bar{y}_{sr}^{(j)} - (\bar{y}^{(j)} - \nu)\right)^2}$.
    (b) Determine the starting point on which to build the PSF according to equation 3.6.
    (c) Optimize the position of the PSF using the non-linear optimization python software library, `NLOpt`.
    (d) Perform local descent, iterating k_L times, as detailed below,
        i. Check if the stopping condition in equation 3.11 is satisfied.
        ii. Compute the optimal weights for the current set of PSFs by solving equation 3.9 using the interior point solver contained in `scipy.Optimize`.
        iii. Prune any PSF that has an intensity equal to or less than zero.
        iv. Perform coordinate descent using `NLOpt` to solve equation 3.6.
2. Generate the image according to equation 2.2, where $\nu = 0$, at the locations, $\{s\}$, and intensities, $\{\omega\}$, given by ADCG.

2 Temporal Tracking Algorithm

2.1 Track Segmentation

Define an upper and lower bound for the adaptive particle search radius ζ_p. Iterate over the data three times, front to back, to ensure the adaptive pruning is representative of the data points. For each iteration of the three iterations, loop $t_f - 1$ times, where t_f is the last time point. Each loop iterates over the current and subsequent frame.

1. Create an $M_t + M_{t+1} \times M_t + M_{t+1}$ cost matrix.
2. Compute the center of mass of the DNA sequence using the YOYO-1[a] stack in the t and $t+1$ frame and subtract all the positions in the corresponding frame.
3. Create a particle-to-track look-up dictionary.
4. Apply a Kalman filter to frame t and compute the element-wise squared distance between frames t and $t+1$.
   (a) Compute the maximal search radius according to equation 3.12 and prune values that exceed it.
   (b) Insert this $M_t \times M_{t+1}$ matrix into the cost matrix.
5. Insert the minimal proposed link distance into the lower right $M_{t+1} \times M_{t+1}$ matrix to create a square matrix.
6. Fill the diagonal of the upper right $M_t \times M_{t+1}$ and lower left $M_{t+1} \times M_t$ sections of the cost matrix with $1.x \cdot \varepsilon$ where ε is the maximal accepted link cost and $1.x$ is an arbitrary scaling parameter slightly larger than 1.
7. Solve the cost matrix using a sparse linear-assignment-problem (LAP) solver, such as the `scipy.Optimize.linear_sum_assignment` function.
8. Depending on the cost matrix output, assign particles from $t+1$ to tracks,
   (a) For optimal links belonging to the upper left section, assign the particles from frame $t+1$ to the track associated with the linked particle from frame t.
   (b) For optimal links belonging to the upper right section, the particle from the $t+1$ frame is linked to nothing and the track is removed from the look-up dictionary.
   (c) For optimal links belonging to the lower left section, the particle from the $t+1$ frame is assigned to a new track, which is updated or added to the look-up dictionary.

[a] One of the dyes used to label the DNA. Refer to section 4.3 for more information.

2.2 Optimization

As a part of the three front-to-back iterations,

1. Create an $3l \times 3l$ cost matrix, conceptually segmented into 9 different $l \times l$ sections.
2. Compute the tip-to-tip squared distances of the tracks and insert them into the upper left $l \times l$ section.
3. Compute the merging and splitting costs according to equations 3.13 and 3.14, and insert them into the upper central $l \times l$ section and the middle left $l \times l$ section, respectively.
4. Compute the rejection costs for merging and splitting according to equations 3.15 and 3.16, and insert them into the bottom central $l \times l$ section and the middle right $l \times l$ section.
5. Compute the 90^{th} percentile of all potential assignment costs, ε', and fill the diagonal of the upper right and lower left $l \times l$ sections.
6. Set the central most section to 0, and the lower right section is set to the minimal potential cost, as before.
7. Compute the search radius at the start and end of the tracks, according to equations 3.17 and 3.18 respectively.
    (a) Exclude costs that exceed the search radius.
    (b) Exclude values which exceed a user defined temporal search radius Δt.
8. Depending on the cost matrix output, merge and split tracks,
    (a) For optimal links belonging to the upper left section, assign the track ends to the corresponding track start thereby creating on track..
    (b) For optimal links belonging to the upper central section, the end of a track I merges with a midpoint of track J.
    (c) For optimal links belonging to the middle left section, the start of a track J splits from a midpoint of track I.

Appendix C

Datasets

1 2013 Grand Challenge Localization Microscopy

1.1 Noise Filtering

The segmentation of the images to deduce the mean noise in the image involved segmenting a single image manually and taking its mean, after which the ADCG routine was run over the entire data set and using the emitter localizations, the image was reconstructed according to equation 2.2 with ν being 0. Thereafter, the super-resolution images were subtracted from the experimental images, and the resulting images were used to calculate the mean noise across the entire data set.

2 AF647 Stained DNA Sequences

For more information regarding the process in which the data was created, read DOI:10.1021/acsnano.6b05398

2.1 Noise Filtering

For the noise parametrization, the YOYO-1 hyper stack was used to deduce the location of the entire DNA molecule by use of a canny filter. Subsequently, the edges were filled and set to zero in the AF647 stack. Finally, the mean noise was calculated for the single frame.

2.2 Temporal Localization Sketch

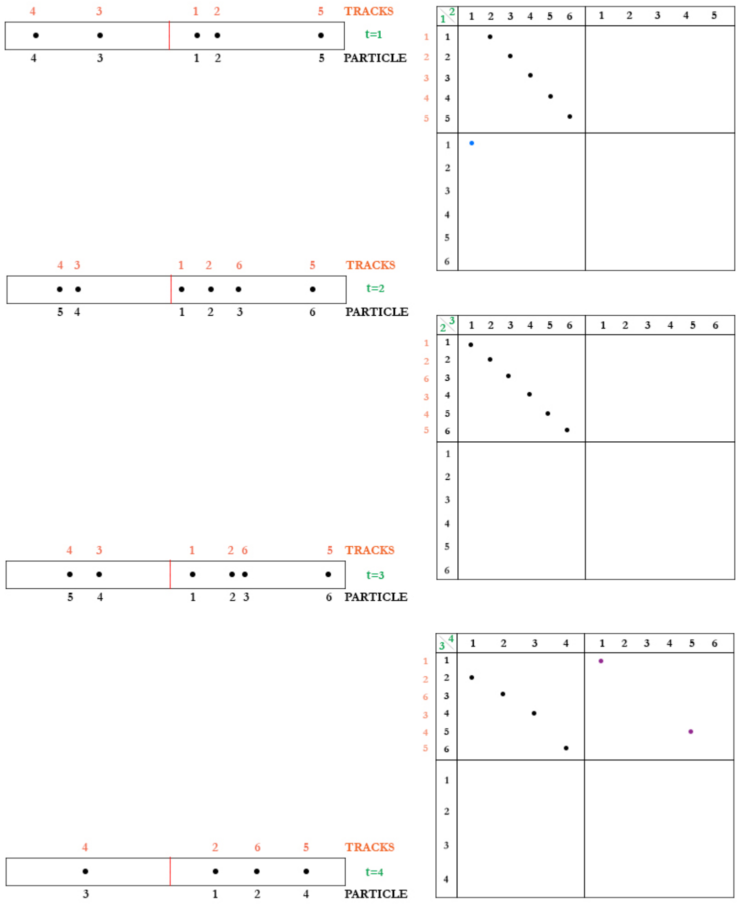

Figure C.1: An artistic representation of the temporal tracking method detailed by Jaqaman *et al.* across four frames. Where the black dots represent the particle-to-particle assignment from one frame to the next, the blue dots represent the start of a new track, the purple dots represent the end of a track, and the red line represents the center of mass of the image.

By using the temporal linking algorithm on the AF647 stained DNA dataset, tracks are generated from the kymograph, as shown in figure C.2, which eliminates short tracks and false localizations induced by noise.

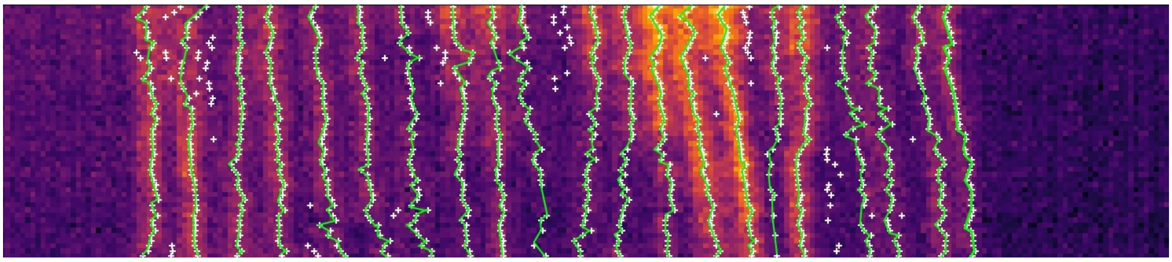

Figure C.2: A 50 frame sample of an AF647 stained DNA sequence that has had its fluorophores assigned to tracks.