

Catmull-Rom spline Channel Registration in Single Molecule Localization Microscopy

Mels Jasper Habold

frontpage.png

Delft University of Technology

layout/tudelft/logo-black.pdf

Catmull-Rom spline Channel Registration in Single Molecule Localization Microscopy

by

Mels Jasper Habold

Studentnumber: 4493508
Instructors: MSc. J. Cnossen
dr. C. Smith
Institution: Delft University of Technology
Place: Delft Center Systems and Control, Delft
Project Duration: February, 2021 - February, 2022

Cover Image: Example of Channel Registration via Catmull-Rom splines.

layout/tudelft/logo-black.pdf

Acknowledgements

Writing a thesis has been an emotional rollercoaster, switching between some weeks with a lot of motivation and progress and some weeks where nothing seems to work and I would just sit and play bass until I would get inspiration. First of all, I would like to thank my supervisor Jelmer for being patient with me, and accompanying me where necessary. I would also like to thank Carlas for introducing me into the domain of scientific figure creation, it has been a welcome surprise to find an output for creativity in my thesis.

I would also like to thank my study friends for the good laughs we have, the motivational words they spoke to me, and being a solid pillar in my life, even when covid seemed to never give us the chance to go on a weekend away together. Next, my house-mates have been essential in cooking for me when I was lazy, but also for probably being the only people I will ever meet that I can spend longer than 5 days in close proximity with. Also, some well deserved thanks goes out to my activist friends, for they inspired me in building a future after studying, to become a better human being, and giving me a purpose.

Special thanks goes out for my family, who will hopefully always be a constant in my life. They have always been ready to offer me emotional support, but lately I have started seeing them more and more as friends, and I hope that relationship will persist.

Most importantly, I would like to thank all the bands that pulled me through, and gave me energy in the last months when writing my thesis. In particular, I would like to thank *Unknown Mortal Orchestra*, *King Gizzard and the Lizard Wizard*, *The Beths*, *Courtney Barnett*, *Metronomy*, *Parquet Courts* and above all *LCD Soundsystem* for making amazing albums that I played on constant repeat.

Lastly I would like to congratulate myself. Working on your thesis is notorious for being a brain killer, a sponge that absorbs all your free time, and overall a hard process. Before I started last year February, I came out of my corona-depression demotivated and without energy, and had been postponing committing to my thesis out of fear for ending up in an even worse state. Somehow, while working on my thesis, I have been able to regain my motivation and energy, have kept good boundaries about personal time, and came out a better person overall. I am really proud of these accomplishments, and will most certainly take this with me for the rest of my life!

*Mels Jasper Habold
Delft, February 2023*

Summary

Single Molecule Localization Microscopy (SMLM) is a technique that uses techniques on fluorescence microscopy in order to bypass the Abbe diffraction limit and reach smaller levels of precision. It does this by fitting the Point Spread Functions (PSFs) of the lens to the pixel data to estimate the true emitter positions. In SMLM, different colour fluorophores can be used to flag different structures in the sample, which means two colour-channels are created. On this scale optical systems start to have effects called aberrations that mutate the PSF, and are dependent on the position and colour of the emitter. Both channels need to be aligned, and the aberrations need to be corrected for.

This thesis proposes a method called Catmull-Rom splines Channel Registration (*CRsCR*) to create a registration map that aligns one channel to the other. This method is based on a similar method described by Niekamp^[1] which we call piece-wise affine Chanel Registration (*PACR*). First, *CRsCR* links localizations between channels to form localization pairs. It then optimizes the distances between these localization pairs via an affine Linear Least Squares transform to correct for global aberrations. Lastly, local aberrations are corrected for via Catmull-Rom splines interpolation.

CRsCR reaches similar precision as *PACR*, as well as a similar distribution of errors over the field of view. The most important limitation of this method lies within the linking of localizations. This is only possible after channels have been sufficiently aligned such that localization pairs lie in the vicinity of each other. Therefore, an improvement to *CRsCR* that allows it to be applied to data that does not allow clean linking of localization pairs has been discussed.

Contents

Acknowledgements	i
Summary	ii
Nomenclature	iv
1 Introduction	1
1.1 Limits in Fluorescence Microscopy	1
1.2 Single Molecule Localization Microscopy	3
1.3 Channel Registration	5
1.4 Background and Outlook	5
2 CRsCR Pipeline	7
2.1 Linear Least Squares Affine optimization	8
2.2 Catmull-Rom Spline Mapping	9
2.3 Optimizing Catmull-Rom Splines	10
2.3.1 Steepest Descent Optimization in TensorFlow	10
2.3.2 Testing the Absolute Error	12
2.3.3 Choosing the Optimization Algorithm	12
2.3.4 Investigating the Convergence.	13
3 Results Fiducial Markers	15
3.1 Validating CRsCR parameters for Fiducial Markers	15
3.2 CRsCR vs PACR Results	16
3.3 Investigating the Cramer-Raó Lower Bound	19
4 Results Förster Resonance Energy Transfer Data	22
4.1 Registration of Clustered Data	22
4.1.1 affineLLS	22
4.1.2 CR-splines	24
4.2 Validation of CRsCR on FRET data	24
5 Discussion	26
5.1 Bias after CRsCR.	26
5.2 Linking the Data	27
6 Conclusion	28
6.1 CRsCR vs PACR	28
6.2 Implementing CRsCR Dense Data	28
6.3 Recommendations	29
References	31
A The CRLB of the Imaging Model	32
B The CRLB of Registered 2-Channels	33
C Validating Affine Linear Least Squares optimization	34
D The Effect of False Channel Pairs	37
E Results without Filtering Pairs	38
F Comparing distribution of CR-splines to the piece-wise affine transform	40

Nomenclature

Abbreviations

Abbreviation	Definition
SMLM	Single Molecule Localization Microscopy
MC-SMLM	Multi-colour Single Molecule Localization Microscopy
PSF	Point Spread Function
CRsCR	Catmull-Rom splines Channel Registration
PACR	Piece-wise Affine Channel Registration ^[1]
CR-splines	Catmull-Rom splines
affineLLS	Affine Linear Least Squares
MLE	Maximum Likelihood Estimate
LLS	Linear Least Squares
FOV	Field of View
SGD	Stochastic Gradient Descent
kNN	k-nearest neighbours
FRET	Föster Resonance Energy Transfer
CoMs	Center of Masses

Symbols

Symbol	Definition	Unit
NA	Numerical Aperture	-
λ	Wavelength of light	nm
μ	Bias	nm
σ	Precision	nm
$p_{2D}(r)$	the probability of finding channel-pairs at distance r from each other	-

Introduction

1.1. Limits in Fluorescence Microscopy

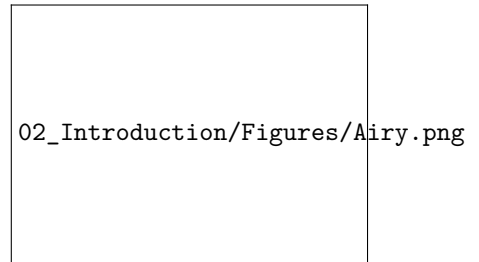
In understanding cell-physiology on a nanometer scale, fluorescence microscopy is a useful tool. This imaging method implements chemical compounds that can be forced to emit light upon excitation, which are called fluorophores. These fluorophores can be attached to certain molecular structures in order to dye them. This means that after using a dye to flag certain structures in a sample, one can shine a light-source on the sample, making the dye become fluorescent and allowing imaging. This method contrasts with conventional microscopes, which are dependent on the reflection and absorption of light to study properties of organic or inorganic substances. Some fluorophores naturally emit light, and to be certain no auto-fluorescent light reaches the camera, low-pass filters are often placed in between the sample and the camera^[2]. Using fluorophores as dye for certain structures also has the advantage that it allows for tracking, in which the dynamical behaviour of a structure can be followed precisely. Another advantage of fluorescence microscopy is that different fluorophores can be attached to different molecules, allowing different types of structures to be tracked simultaneously^[3]. An example of such a setup, together with its possible multi-color image is given in figure 1.1.



Figure 1.1: Example of multi-color fluorescence microscopy. **(A)** Experimental setup. A laser excites different fluorophores, which then emit their own color spectrum. Each color is then split using dichroic mirrors and are captured by their respective cameras. **(B)** An example of the excitation spectrum of the MitoTracker Red CMXRos. The excitation spectrum contains the normalized efficiency of exciting the fluorophore into a fluorescent state. This means the fluorophore will be best excited with a laser of 578nm, after which it will emit a spectrum of light around the 600nm. **(C)** An example of a multi-color fluorescence image, containing Bovine Pulmonary Artery Endothelial cell nuclei stained blue with DAPI, mitochondria stained red with MitoTracker Red CMXRos, and F-actin stained green with Alexa Fluor 488 phalloidin and imaged on a fluorescent microscope.

On the nanometer scale on which we wish to image fluorophores, new difficulties arise. One of these difficulties is the Abbe diffraction limit, which is a physical limit that originates from the wave nature of light and the finite aperture of an optical system. This means that for conventional imaging, reaching resolutions below a few hundred nanometer becomes impossible, as the image becomes blurred by the lens. This has been shown in figure 1.2A.

To quantify this limit, it is convenient to talk in terms of the Point Spread Function (PSF) which describes how the optical system responds to a single point of light. This means that, when imaging an object, it becomes convoluted with the PSF of said optical system. A lens is limited by its numerical aperture (NA), which is an indication to how much the incoming light is limited by the finite size of the lens. For the common case of the lens limited by its NA, the PSF takes the shape of an Airy disk. This means that a single point source of light will always have a blur of approximately $\lambda/(2NA)$ in the focal imaging plane when using the microscope.



Airy Disk



Figure 1.2: (A) Example of the workings of the Abbe limit. The limited numerical aperture (NA) of the lens introduces a blurring effect on the object, transforming the image to take the shape of an Airy disk function. This effect originates from the wave nature of light and is determined by the NA of the optical elements. Therefore the Abbe limit is a theoretical limit on the angular resolution of an optical system dependent on the wavelength of the light. (B) Examples of different optical aberrations described by Zernike Polynomial Functions. These aberrations originate from imperfections in the optical system, such as variations in the lens surface or refractive index or misalignment of optical elements, and can in principle be solved by increasing the optical quality of the system. (C) Example of how to use global affine maps to correct optical aberrations^[4]. (D) Comparison of an ideal image of a ring (top) a ring with only an axial (middle) and only a transverse (bottom) chromatic aberration.

With the diffraction limit, this means that for small structures that are on a nanometer scale in size, the maximum resolution using conventional microscopes lies at about 200nm in the optical plane and at least 500nm along the optical axis. One such example of a diffracted-limited fluorescent microscopy image is given in figure 1.3, where different excitation laser intensities are used, for different values of background, which is an indication of the strength of the filter used to absorb auto-fluorescent light. In figure 1.3C we see that, where point sources of light are used, the image becomes distorted to a Gaussian shape with a width of more than 400nm.

Another difficulty is produced by an imperfect optical system, and can range from misalignment in the optical setup to imperfections in optical elements. For example, because a lens has a slight chromatic dependent index of refraction, it splits light of different wavelengths. This means that different colours of light travel through different parts of the optical system. As optical equipment rarely is perfectly shaped, the different colours of light are deformed differently by local deviations within the optical system. The

02_Introduction/Figures/AbbeLimit.jpeg

Figure 1.3: An example of the Abbe Limit in imaging. **(a)** The complete dataset. **(b)** Is zoomed in on a certain region. **(c)** Comparison between single localizations for different laser intensities (I) and background (BG). As can be seen, the original point-like Fluorophores are convoluted by a Gaussian like PSF of a lens. The resulting image is imperfect and has a much lower resolution^[5].

aberrations impact the PSF of the system and therefore the total image becomes deformed. We like to describe aberrations with Zernike polynomials of which their impact on the PSF is shown in figure 1.2B. Due to local imperfections or fluctuations of the refractive index, aberrations often vary slightly throughout the FOV. As Zernike polynomials are described as a single global function, they are not able to correct these local variations. Therefore, it is useful to make a distinction between global aberrations, which are uniform over the whole field of view (FOV), and local aberrations, that are dependent on the position.

1.2. Single Molecule Localization Microscopy

One method that circumvents the diffraction limit is called Single Molecule Localization Microscopy (SMLM). This technique relies on the isolation of single emitters after which the PSF is fitted per emitter to estimate the true emitter positions.

To be able to isolate single emitters, one needs to be certain that the data is sufficiently sparse such that emitter PSFs do not overlap such that single emitters are distinguishable. For this reason SMLM often uses Stochastic Optical Reconstruction Microscopy (STORM) or Photoactivated Localization Microscopy (PALM). Both methods implement the stochastic activation and deactivation of fluorophores. PALM uses the process of photobleaching, in which the fluorophore loses its fluorescent characteristics. STORM implements photo-switchable fluorophores which can be excited to a fluorescent state and forced to a non-fluorescent state in a stochastic process called blinking. In both processes, only a random subset of fluorophores is in a fluorescent state, meaning the data now consists of frames of sparse data. This means one can isolate single emitters and fit them to the PSF of a lens using a maximum likelihood estimate (MLE). The PSF, which has the shape of an Airy Disk, will be approximated by a Gaussian with standard deviation $\sigma^2 = \sigma_x^2 + \sigma_y^2$,

$$PSF(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-\theta_x)^2 + (y-\theta_y)^2}{2\sigma^2}}. \quad (1.1)$$

In which x and y are the position in the image and θ_x and θ_y the estimates of the true emitter position. Besides the PSF, the model should also incorporate an estimate for the emitter intensity θ_{I_0} and the background θ_{bg} .

As camera uses pixels, it can only measure discrete data, and therefore the imaging model that will be fitted over the emitter also needs to be discrete. Following the derivation of Smith^[6], the eventual

model has an expected value μ_k for the k-th pixel with surface A_k of,

$$\begin{aligned}\mu_k &= \theta_{I_0} \int_{A_k} PSF(u, v) du dv + \theta_{bg} \\ &= \frac{1}{4} \theta_{I_0} \left[\left(erf\left(\frac{x - \theta_x + \frac{1}{2}}{2\sigma^2}\right) - erf\left(\frac{x - \theta_x - \frac{1}{2}}{2\sigma^2}\right) \right) \left(erf\left(\frac{y - \theta_y + \frac{1}{2}}{2\sigma^2}\right) - erf\left(\frac{y - \theta_y - \frac{1}{2}}{2\sigma^2}\right) \right) \right] + \theta_{bg}.\end{aligned}\quad (1.2)$$

In which erf is the error function, which is part of the analytical solution of integrating a Gaussian. A visual representation of this process is given in figure 1.4, where we can see the original wide-field image where emitters are first isolated (figure 1.4AB). These isolated emitters are then fitted a Gaussian (equation 1.2), which returns an estimate of the localization together with an estimate to its intensity and background.



Figure 1.4: Principle of single-molecule localization microscopy. **(A)** A structure is labelled with photoswitchable or photoactivatable fluorescent probes and imaged by wide-field fluorescence microscopy. **(B)** Activation of only a subset of fluorophores at any time of the experiments allows the isolated detection. **(C)** Position determination of individual fluorescent probes with high precision. **(D)** Localization pattern of all localizations performed. To improve visualization of the localization pattern artificial subpixels with a size of 1/10 of the regular pixel size are usually applied to colour code the density of localizations **(E, F)**. dSTORM, direct stochastic optical reconstruction microscopy. ^([7])

The fact that we are fitting multiple parameters via MLE at the same time means the algorithm is limited by the Fisher information matrix,

$$I_{ij}(\theta) = E \left[\frac{\partial \log(L(X|\theta))}{\partial \theta_i} \frac{\partial \log(L(X|\theta))}{\partial \theta_j} \right]. \quad (1.3)$$

In which the E stands for the expectation value, ∂ is the partial derivative and $L(X|\theta)$ the likelihood of a Poisson process characterised by,

$$L(X, \theta) = \prod_k \frac{\mu_k(x, y)^{X_k} e^{-\mu_k(x, y)}}{X_k!} \quad (1.4)$$

which is the likelihood of the data X based on the imaging model parameterised by $\theta = [\theta_x, \theta_y, \theta_{I_0}, \theta_{bg}]$. This means that estimating a parameter there will always have an inherent limit to the precision of this estimation, defined by the Cramer-Raó Lower Bound (CRLB), which can be found on the diagonal axis of the inverse Fisher information matrix,

$$Var(\theta) \geq I(\theta)^{-1}. \quad (1.5)$$

This equation is further calculated by Smith^[6], and the results are given in appendix A.

Using SMLM, localizations with a precision below 10nm in the optical plane and 20nm along the optical plane can be achieved ^([8]).

1.3. Channel Registration

Now that SMLM bypasses the diffraction-limit and reaches resolutions down to a few nanometers, the super-resolution image can be constructed. In multi-colour single molecule localization microscopy (MC-SMLM), different channels need to be mapped over each other. This is because lenses diffract light differently, and often different cameras are used for imaging different colors. Often this is done via a translation map to realign the images. This method only focuses on aligning the channels, and does not take deformations caused by aberrations into account.

These aberrations can be mitigated by improving the optical system and its components. With imperfect equipment however, aberrations will always have an effect on the different channels. In the case only one color fluorophore is used, one has no reference to what these aberrations look like, and therefore correcting them is impossible. Therefore researchers often assume the effect of aberrations to be negligible. In the case of MC-SMLM, one can use the other channel as reference and can transform one channel to fit over the other. This process is often referred to as channel registration, and focuses on reducing the effect that optical aberrations have on co-localization in MC-SMLM.

These aberrations approximately have an effect that can result into differences up to 1.5 pixels between channels, which would mean an error greater than 200nm^[9]. One can use global affine translations to map these deformed channels to fit over each other. These global affine transformations can be decomposed into the four linear transforms given in figure 1.2C. In most cases global maps are not able to correct for all the aberrations, and therefore local maps should be implemented also^[10].

A good way to evaluate channel registration is via the co-localization error^[11]. This error takes the distance between linked localizations in different channels. Often, we either work with localizations that have a corresponding localizations in the other channel, or localizations that are anisotropically distributed around each other. Therefore we expect the error distribution after registration to have zero mean.

As noted before, SMLM is not able to perfectly estimate the localizations, and therefore all localizations are subjected to a normal distributed localization error. In appendix B we have derived the unbiased co-localization error to follow a zero-mean normal distribution. If aligned perfectly, the standard deviation of this distribution would equal $\sqrt{2}$ times the average localization error. This optimal curve would be the CRLB.

Of course, registration will not always happen perfectly. If the eventual error distribution is not zero mean, this would indicate a bias, which is a measure of systematic error. All deviations from the optimal standard deviation can be accounted by the co-localization precision, which is the statistical error.

1.4. Background and Outlook

Channel registration is often done by optimizing the cross-correlation between channels. Often cross-correlation is used to optimize a translation, which means it will be unable to correct for higher order aberrations. A method that improves upon this is described by Guizar-Sicairos^[9], and uses cross-correlation together with a translation to align subpixel cells. This method is very limited, as it still only corrects for low order aberrations and discontinuities in the transformations between different cells are expected. Therefore, it should be no surprise that Guizar-Sicairos has been unable to reach sub-20nm accuracy. Yan^[12] takes a more theoretical approach, quantifying the optical aberrations of the imaging system as linear combinations of Zernike polynomials. This approach teaches us that lower order Zernike polynomials have a stronger impact, however it only reaches 25nm precision. Another proposed method is that of Li^[13], which links localizations between channels that are in the same frame. The distance between these localization pairs is then optimized according to a global affine transform. All these methods, although able to solve global aberrations to some extent, are unable to correct for local aberrations^[10]. Therefore, a method is needed that is able to register channels both on a global and a local level.

Such a method is described by Niekamp^[1], which still focuses on optimizing the distances between localization pairs, but does so by using a piece-wise affine transform on top of a global one. For comfort, we name Niekamps method piece-wise affine Channel Registration (*PACR*). *PACR* succeeds in reaching accuracy below the 1nm and precision under the 10nm, but is dependent on the linking of localization pairs, which means that the dataset has to be sufficiently sparse to allow clear linking. *PACR* is also dependent on fiducial markers that show up in both channels and therefore should give us precise insight into the aberrations. A registration map that should represent the aberrations throughout

the FOV is based on the fiducial marker dataset, and will be applied to correct aberrations in the sample of interest.

The goal of this thesis is to implement a similar method, but instead of the piece-wise affine transform, which should be discontinuous on small scales, we focus on Catmull-Rom splines that force first-order continuity, and therefore false localization pairs should not have a strong detrimental impact on the registration map. Another goal is to make our method in such a way that it is able to register data that does not necessarily consist of localization pairs such as the fiducial markers. This method, named Catmull-Rom spline Channel Registration, abbreviated to *CRsCR*, will be derived in section 2, in which we look at the global affine transformation via Linear Least Squares, derive the Catmull-Rom splines interpolation technique, and show how to implement this in TensorFlow. In section 3, we will discuss the results by first bench-marking it to the results of Niekamp. In section 4, we will implement it for a more complex set of data, to show *CRsCR* is also able to register data that does not easily allow linking. Lastly, in section 5, we will discuss the limitations of *CRsCR*, and give some recommendations on how to improve on them.

2

CRsCR Pipeline

The pipeline of *CRsCR* has been visualized in figure 2.1. We start out with two channels, both subjected to global and local aberrations, of which the second needs to be mapped onto the first. Localizations between channels are then linked to each other according to a nearest neighbour algorithm to form localization pairs. Thereafter, an affine mapping is derived by optimizing the distances between localization pairs, or pair-distances, via a Linear Least Squares (LLS) method. After transforming via affine Linear Least Squares (*affineLLS*) localization pairs with pair-distances above a certain distance are assumed to be false localization pairs, and will be filtered out. Lastly, Catmull-Rom Splines (CR-splines) are used to optimize the residual pair-distances, after which we assume most of the global and local aberrations have been corrected.

00_Main_Figures/Figure_1.png

Figure 2.1: Catmull-Rom spline Channel Registration (CRsCR) pipeline overview. **(A)** Two channels with their localizations. The second channel has been deformed, which is why the localizations do not overlap. **(B)** localizations in the first channel will be linked to localizations in the second channel. **(C)** Channel 2 will be aligned using a Linear Least Squares Affine optimization after which localization-pairs that are further than 1.5 pixels apart are assumed to be incorrect pairs and will be deleted. **(D)** Channel 2 will be aligned via Catmull-Rom splines interpolation in which the localizations are mapped by changing the locations of the control-points according to equation 2.5.

CRsCR will return a registration map that maps the second channel onto the first, such that they can be combined to represent the super resolution image of the sample. This map will also correct for aberrations in the channels. However, as this map only maps the second channel onto the first, it implicitly assumes all aberrations to lie within the second channel. As we are mostly interested in the relative positions of the localizations, and because higher order aberrations only have a small effect on the positions, this assumption does not have a strong detrimental effect on the overall resolution. We could extend our assumption to say that the deformations are equally spread over both channels, meaning we have to correct both channels to each other. However, taking one channel as reference greatly simplifies the mathematics, and also makes it possible to register more than two channels. The first step of our algorithm would be to link all localizations between channels to create a set of matching localizations which we call localization pairs. For linking localizations, it is important to choose the right distance within which we search for localization pairs, as we want to have the least possible amount of false localization pairs to make sure the algorithm converges on the correct solution. For this reason, after each optimization step, localization pairs that lie above a certain distance from each other will be filtered out. This however will also result in some true localization pairs to be filtered out.

Therefore, the distances within which to search and above which to filter localization pairs should be chosen with care, as to filter out most false localization pairs while not losing information. As an indication, Guizar-Sicairos^[9] notes that all higher order aberrations are limited to a displacement of 1.5 pixels, which for a pixel size of 159nm should amount to a maximum pair-distance of about 240nm. Therefore, we initially choose to search for localization pairs within a distance of 1000nm, and filter localization pairs above 240nm after aligning both channels via an affine transform.

For the next step, we globally align both channels using an *affineLLS* transform. This transform, as shown in figure 2.1C, keeps co-linearity and ratios of distances intact, and can be decomposed to translation, rotation, shear and scaling transformation, and therefore is limited to a small number of parameters. The affine transform and its analytical solution are described in section 2.1. As described previously, all localization pairs that have pair-distances greater than 240nm after applying *affineLLS* will be assumed to be false pairs and will be deleted.

Lastly, Catmull-Rom splines are used to correct for local aberrations. This polynomial interpolation method keeps first-order continuity and uses a grid to create this interpolation. As it utilizes a great number of parameters to minimize the distances, this method is complex to solve analytically, and therefore a steepest descent algorithm will be implemented to solve this optimization problem. CR-splines will be described in more detail in section 2.2 and the computational optimization methods are discussed in section 2.3.

Using analytical affine transforms for channel registration is something that is done quite commonly (like by *PACR*^[1]), which uses it on both a global and a local level, and Li^[13], who only uses a global transform. The implementation of Catmull-Rom spline interpolation on SMLM has often been done to generate the PSFs to match the data^{[14][15][16]}, and although Linde^[7] does note the method as a possible registration method, no research has been done on the topic specifically. For these reasons, this thesis does not research the *affineLLS* algorithm to a great extent, but will focus on the implementation and validation of Catmull-Rom splines in Channel Registration.

2.1. Linear Least Squares Affine optimization

As discussed previously, the affine transform is a linear transform that keeps co-linearity and ratios of distances intact, and can be decomposed to a combination of the translation, rotation, shear and scaling transforms (figure 1.2B). The 2-dimensional affine transforms a coordinate \mathbf{x} following,

$$\mathbf{x}^* = A\mathbf{x} + \mathbf{d}. \quad (2.1)$$

In which \mathbf{d} is a 2-dimensional vector containing the translation and A is a 2x2 matrix containing the other transformations. To simplify this, we concatenate the vector \mathbf{d} to the matrix A , which will now become a 2x3 matrix representing the whole affine transform. This also means that the input vector \mathbf{x} becomes a 3-dimensional vector in which the last element equals 1, while the transformed coordinate \mathbf{x}^* stays 2 dimensional.

For N localizations, equation 2.1 becomes,

$$X^* = AX. \quad (2.2)$$

in which X is a 3xN matrix containing the vectors \mathbf{x}_i and X^* is a 2xN matrix containing the vectors \mathbf{x}_i^* with $i \in \{1, 2, \dots, N\}$. In this problem, the affine transform should map the coordinates of the second channel in matrix X to the coordinates of the first channel in matrix Y . In this case, we will use linear least squares (LLS) optimization,

$$A^{opt} = \underset{A}{\text{Min}} ||Y - AX||^2.$$

As this is a linear transform over two dimensions, we can split the two rows of matrix A into two 1x3 matrixes $\{A_1, A_2\}$ and the rows of Y into two 1xN matrixes $\{Y_1, Y_2\}$ to get two linear independent optimization problems,

$$\begin{aligned} A_1^{opt} &= \underset{A_1}{\text{Min}} ||Y_1 - A_1 X||^2 \\ A_2^{opt} &= \underset{A_2}{\text{Min}} ||Y_2 - A_2 X||^2. \end{aligned}$$

Of which the solution can be found by taking the derivative of both equations to their respective parameter A_j^T (with $j \in \{1, 2\}$) and equating it to zero. The solution to this LLS problem is known to be,

$$A_j^{opt} = Y_j X^T (X X^T)^{-1}. \quad (2.3)$$

In which the T superscript indicates the transpose of the matrix, and the -1 superscript the inverse of the matrix.

2.2. Catmull-Rom Spline Mapping

Next, we wish to more finely align the localization pairs. For this we utilize Catmull-Rom splines, or CR-splines, which are an interpolation technique based on a grid of control points on which the interpolation curve is dependent. This means that changing the locations of these control points will change the mapping curve in between the control points. Spline curves have the advantage that they are able to create a transform on a local level while also using neighbouring control points to force the curve to be smooth (C1 continuity). In the next section, we will derive CR-splines and show how to implement them.

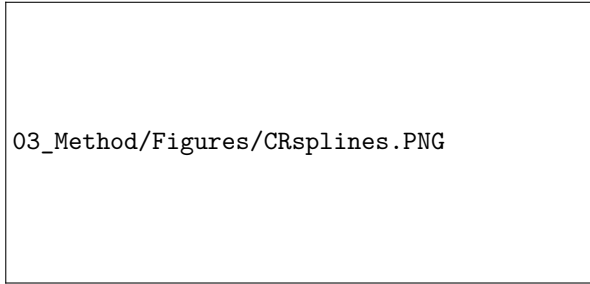


Figure 2.2: A spline segment as defined by^[17]. In this case, the points p_{i-1} and p_i define the position of the curve, while p_{i-2} and p_{i+1} guarantee the C1 continuity of the total curve. The tangent in p_{i-1} is chosen to be $0.5(p_i - p_{i-2})$.

We follow the 1D Catmull-Rom splines derivation as defined by^[17], which start out with 4 control-points ($p_{i-2}, p_{i-1}, p_i, p_{i+1}$) on which a polynomial curve is defined by,

$$p(u) = \sum_{k=0}^3 c_k u^k. \quad (2.4)$$

Where u is the distance between the input position and the control-point p_{i-1} normalised by the grid size $|p_i - p_{i-1}|$. Note that each control-point influences the interpolation curve of the two surrounding cells. Next we will choose the parameters c_k of the polynomial function $p(u)$ in such a way that they will force C1 continuity. Therefore, we take a look at how the curve is defined on the edges, and make sure there are no discontinuities in the curve and its tangent. As we have the freedom to choose how the tangents are defined, we choose the tangent at a control-point to be parallel to the slope of the surrounding control-points. This results in,

$$\begin{aligned} p(0) &= p_{i-1} \\ p(1) &= p_i \\ p'(0) &= 0.5(p_i - p_{i-2}) \\ p'(1) &= 0.5(p_{i+1} - p_{i-1}). \end{aligned}$$

of which the apostrophe indicates the first-order directional derivative to u . Applying these constraints to equation 2.4 and skipping over some linear algebra, we get,

$$\begin{aligned} c_0 &= p_{i-1} \\ c_1 &= 0.5(p_{i-1} - p_{i-2}) \\ c_2 &= 3(p_i - p_{i-1}) - 0.5(p_{i+1} - p_{i-1}) - (p_i - p_{i-2}) \\ c_3 &= -2(p_i - p_{i-1}) + 0.5(p_{i+1} - p_{i-1}) + 0.5(p_i - p_{i-2}). \end{aligned}$$

Which means that, written in matrix form, equation 2.4 becomes,

$$f(u) = \begin{pmatrix} u^3 & u^2 & u^1 & u^0 \end{pmatrix} \begin{pmatrix} -0.5 & 1.5 & -1.5 & 0.5 \\ 1 & -2.5 & 2 & -0.5 \\ -0.5 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} p_{k-2} \\ p_{k-1} \\ p_k \\ p_{k+1} \end{pmatrix}. \quad (2.5)$$

To implement 2D splines, we take the product of their 1D equivalent in both directions, which means the new curve is dependent on 16 control points instead of 4, of which the 2-dimensional orientation is given in figure 2.3.

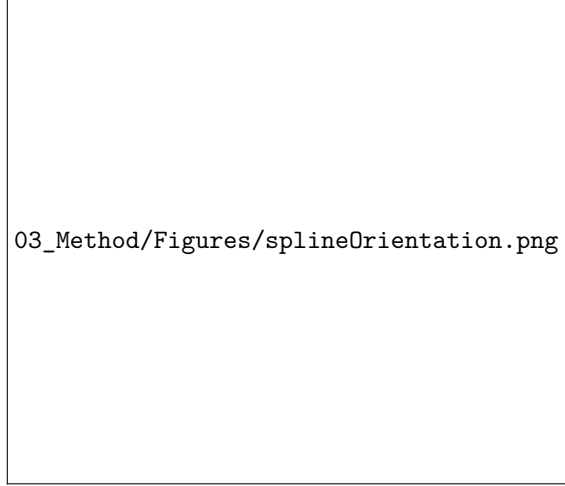


Figure 2.3: The orientation a 2D spline control-point grid. The area on which the curve belonging to control-point $W_{k,k}$ is defined is highlighted in red. All points within this area are controlled by the 16 surrounding points shown in the figure.

For simplicity, the resulting matrix multiplication is written in sum form,

$$f(u_x, u_y) = \sum_{a=0}^3 \sum_{b=0}^3 u_x^a u_y^b \sum_{i=0}^3 \sum_{j=0}^3 A_{a,i} A_{b,j} W_{i-1,j-1}. \quad (2.6)$$

Of which the control points $W_{i,j}$ are initially oriented as in figure 2.3. The variables u_x and u_y are the x-and-y-distance of the coordinate to control-point $W_{0,0}$, normalized by the gridsize. In practical terms this means that the grid is defined as a unit grid, and all positions in the grid are translated to the input u by taking the modulus of 1. The matrix elements $A_{i,j}$ are defined by

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -0.5 & 0 & 0.5 & 0 \\ 1 & -2.5 & 2 & -0.5 \\ -0.5 & 1.5 & -1.5 & 0.5 \end{pmatrix}.$$

An example of how the Catmull-Rom splines work is given in figure 2.2.

2.3. Optimizing Catmull-Rom Splines

2.3.1. Steepest Descent Optimization in TensorFlow

We can now define the optimization problem for CR-splines, again following the linear least squares approach, as,

$$\Theta^{opt} = \underset{\Theta}{Min} ||Y - M(X, \Theta)||^2. \quad (2.7)$$

In which Y is the $2 \times N$ matrix containing all the localizations of channel 1, and X the $2 \times N$ matrix containing all the localizations of channel 2. M is now defined as the CR-spline transformation from equation 2.6 depending on the locations of the control-points Θ .

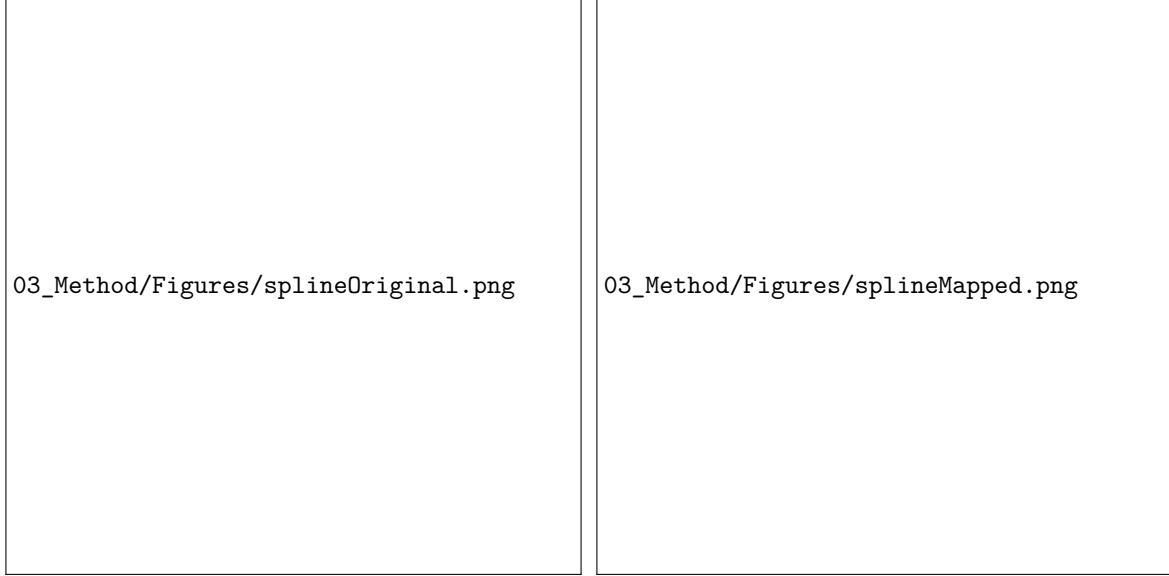


Figure 2.4: An example of the workings of the Catmull-Rom spline interpolation. **(left)** The original image. The blue points need to be optimized to lie closer to the red points. **(right)** Catmull-Rom splines are used to optimize the distance between the blue and the red points. A fine grid is given as example in light blue to show how points are transformed over the whole Field of View. The dark blue grid is the grid as created by the control-points. Note that the distances between points seem to be almost perfectly optimized while we can also see that there are no discontinuities in the fine grid, and it looks like there are no discontinuities in the first-order derivative also.

As the CR-splines need to correct aberrations on a local level, while not over-fitting the data, there will be a large amount of control-points present. This means that CR-splines will be complex and become hard to calculate via analytical methods. However, all these control-points have influence over each other, which means they cannot be optimized one at the time. Therefore, we will implement a gradient descent algorithm, which will allow the algorithm to find its own optimum by following the curve in which the loss function has the largest downward gradient,

$$J(\Theta) = ||Y - M(X, \Theta)||^2 \quad (2.8)$$

$$\Theta := \Theta - \mu \nabla_{\Theta} J(\Theta). \quad (2.9)$$

Where J is the cost-function, ∇ the nabla operator, or the vector containing all partial derivatives to the elements of Θ , and μ is the learning rate, which decides how strongly the parameters may change dependent on the gradient. An example of how the learning-rate influences the convergence is given in figure 2.5. If μ is chosen too small, the algorithm will converge too slowly, which means the algorithm becomes inefficient, and it might also result in the parameter getting stuck in a local minimum. Choosing μ too large however may result in divergent behaviour. Therefore, choosing the right learning-rate is essential for having an efficient algorithm. To implement the gradient descent algorithm, we use a python library called TensorFlow. TensorFlow already has a great number of gradient descent algorithms installed, which saves us a lot of time. These different algorithm all show different convergent behaviour, and therefore it is vital to research them. In the next section, we will investigate which one to use, together with optimization parameters like the learning-rate and the amount of iterations.

To test these parameters, we need to have a dataset to test them on. For this we generate a 400x400 grid, which will be copied to create two channels, as seen in figure 2.8A. Both channels will get a $\mathcal{N}(0, 1.4nm)$ localization error¹, after which the second channel will then get an affine transformation and a random CR-spline transformation to generate both global and local deformations, as seen in figure 2.8B. We already know the corresponding localizations, and therefore optimizing the pair-distances should converge upon the CRLB. We then apply CR-splines for different optimization algorithms, learning-rates and epochs in order to investigate the convergence for these parameters.

¹the localization error of the fiducial markers as noted by the manufacturer

03_Method/Figures/LearningRate.png

Figure 2.5: Example of how the learning-rate influences the gradient. **(left)** The learning-rate is too low and requires many updates before reaching the minimum point. It also has a higher probability to converge unto local minima. **(middle)** The optimal learning-rate swiftly reaches the minimum point and does not converge unto local minima. **(right)** Too large of a learning-rate, causes drastic updates which lead to strong divergent behaviour.

2.3.2. Testing the Absolute Error

After implementing the model, we needed to take a look at how the distances between localization pairs are distributed. These pair-distances have been used as an indication for the registration error, and they follow the 2D pair-distance probability^[18],

$$p_{2D}(r) = \left(\frac{r}{\sigma^2}\right) \exp\left(-\frac{\mu^2 + r^2}{2\sigma^2}\right) I_0(r\mu/\sigma^2). \quad (2.10)$$

In which $p_{2D}(r)$ is the probability of finding localization pairs at distance r from each other, σ is the precision and μ is the bias of the data. I_0 is the modified Bessel function of the zeroth order. When the bias μ is much larger than the precision, this distribution converges to a Gaussian, which is expected of a distribution with a large bias. Important to note however is that when fitting the data to a Gaussian instead of equation 2.10, the bias will result in an overestimation of $\sigma^2/(2\mu^2)$ ^[18].

03_Method/Figures/churchmanfig1.png

Figure 2.6: Example how the channel-pair distance distribution from equation 2.10 from^[18] looks like for different ratios between the bias μ and the precision σ . In this case, μ has been kept constant at 10nm, while σ has the values of 3, 6 and 10 respectively for figures **(A)** to **(C)**. As we can see, equation 2.10 converges on a Gaussian distribution for larger bias-precision ratios.

When investigating the bias equation 2.10 has its limits. This is because the distribution is calculated by integrating the error over a circle with radius r (figure 2.7). This means that the distribution is not well suited to investigate the bias of the distribution of error. This is because the highest accumulation of data does not necessarily have to lie on the circle that contains the mean of the Gaussian (the black line in figure 2.7), but could lie further outwards where the surface of the circle is larger and therefore can contain more data. This means that, although there is a strong correlation between the parameters of equation 2.10 and a Gaussian fit, they might deviate from each other. Therefore we prefer to investigate the bias by looking at the distribution of the x-and-y error. Equation 2.10 does give a clear indication to the precision of the error, and thus we will still use this distribution to investigate it.

2.3.3. Choosing the Optimization Algorithm

We will start out by investigating the optimization algorithms that TensorFlow has installed. These optimizers are *SGD* (Stochastic Gradient Descent), *Adam* (*SGD* plus first and second order moments),

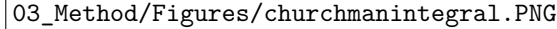


Figure 2.7: Example of difference between the distribution described by equation 2.10, and a Gaussian distribution. The original distribution is a bi-variate Gaussian which has been shown in gray scale. Equation 2.10 reduces this 2D Gaussian to a 1D distribution by integrating the data that lie on the surface of a circle with radius r . This means that, although the actual mean of the Gaussian lies on the black dotted circle, the mean of equation 2.10 will lie on the red dotted circle as it contains more data.^[18]

AdaMax (different implementation of Adam), *AdaDelta* (SGD plus learning-rate decay), *AdaGrad* (SGD plus adaptive learning-rate), *NAdam*, *RMSprop* (both of which are SGD plus some type of momentum) and *FRLT* (Follow The Regularized Leader). We test these optimizers on the grid for variable amount of iterations, and fit the resulting distribution of pair-distances to equation 2.10. The resulting bias is compared per optimizer against the amount of iterations in figure 2.8C. We can see that all optimizers seem to be convergent, although some more strongly than others. The best possible optimizer seems to be SGD, which stands for Stochastic Gradient Descent, and follows equation 2.9. As calculating multiple derivatives becomes computationally expensive for optimising a big dataset, SGD replaces the actual derivative that is calculated over the whole dataset by an estimate of the derivative which is calculated over a random subset of the data, thus reducing the computation time.

Paired with the optimizer, the learning-rate is the next parameter for our optimisation problem. The learning-rate dictates how much we are adjusting the parameters of our model with respect to the loss function, and choosing the right learning-rate is essential to get convergent behaviour in our optimization problem, which is shown in figure 2.5. Again we implement CR-splines on the grid, but now for variable learning-rates. The resulting pair-distances are again fitted by equation 2.10, and the resulting bias and precision are plotted against the learning-rate in figure 2.8D, both for half of the data on which the parameters were estimated, and on another half which is used to test the results via cross-validation. As we can see, a learning-rate larger than $8e-4$ returns the lowest bias and precision, although the learning-rate should be limited to below about $8e-2$ as after which the optimization algorithm diverges strongly.

2.3.4. Investigating the Convergence

Looking at figure 2.8E, which plots the x-and-y-error over the whole Field of View (FOV) after optimization using the SGD optimizer with a learning-rate of $2e-3$ and 300 iterations, we can see that the program reaches a low and uniform error, meaning it seems no bias is present, and the precision converges on the CRLB.

To verify that the algorithm converges, we would like to look at how the precision depends on the density of data. For this we generate a grid of variable N points with affine and spline deformations. We let the algorithm register the data, after which the resulting pair-distances are fitted with equation 2.10. The different distribution curves that have been calculated are plotted in figure 2.8E, together with their fitted precision values. Also shown is the CRLB curve, which is the same as equation 2.10 with a co-localization error of $\sqrt{2}\sigma_{loc} = 1.98nm$. As we expected, the algorithm seems to converge on the CRLB curve as the density of data becomes higher and more information is present.

00_Main_Figures/Figure_2_Grid.png

Figure 2.8: Optimizing the Catmull-Rom splines Registration parameters for a deformed grid of N points. **(A)** A 400x400 grid of localizations is created and copied to create 2 channels. Both channels are given a $\mathcal{N}(0, 1.4nm)$ localisation error and the second channel is given an Affine and a Spline deformation. **(B)** The x-y-distances between corresponding localizations shown in a contour plot. **(C)** The fitted bias and precision vs the amount of iterations given for the 8 different TensorFlow optimizers (Adam, AdaDelta, Ftrl, RMSprop, AdaGrad, AdaMax, Nadam and SGD). CR-Splines are trained on half of the data for different optimizers to investigate the convergence of the optimizer. The resulting registration map will be tested on the other half of the data, after which the resulting distribution of pair-distances will be fitted by equation 2.10. The fitted bias and precision are shown for each of the optimizer against the amount of iterations. **(D)** The fitted bias and precision vs the initial learning-rate. The CR-splines with said learning-rate are initialized and trained for 300 iterations on half of the data to create a map that optimizes the pair-distances. This registration map is then used on another half of the data as a test. The resulting pair-distance distribution of both the estimation data (in blue) and the testing data (in red) after registration is fitted by equation 2.10 of which the resulting bias is plotted together with its precision against the initial learning-rate. From this figure we can derive that an initial learning rate higher than about $8e-4$ and lower than $8e-2$ needs to be used. **(E)** The x-and-y-distances between corresponding localizations after CRsCR optimization shown in a contour plot. **(F)** The convergence of the algorithm depending on the density of the data. For this, the same grid is generated and deformed, but now with a variable amount of localizations. The resulting data is fitted according to equation 2.10 and the resulting fit-curves are plotted together with the optimal curve of $\mu = 0nm$ and $\sigma = 1.98nm$ ($\sqrt{2}\sigma$ derived in appendix B). We can see that as the density increases, the plot becomes more accurate as more information is present, and the data converges on the CRLB.

Results Fiducial Markers

We would like to start with validating *CRsCR* by comparing it to *PACR*. The latter uses a sparse sample of fiducial markers to base their registration map on. This single sample of fiducial markers shown in figure 3.1A should be sufficiently sparse to allow accurate linking of localizations, and is moved using a piezo stage over a 20x20 grid to create 400 frames which combined form a dense dataset (figure 3.1B). When applying *CRsCR*, this means that linking of localizations should happen per frame, in which the data is sparse, and optimization should happen over the whole dataset, which is dense and therefore allows for precise correction of the aberrations over the FOV.

After creating the fiducial marker dataset, *PACR* images the sample of interest within the same optical system, after which the same fiducial marker sample will be imaged again to create a second fiducial marker dataset. The first set of fiducial markers is assumed to be representative of the aberrations within the system at any time, and will be used to estimate a registration map on. This registration map will then also be applied to the second set of fiducial markers as a means to test the map. When the registration map has been validated, it can be applied to register the sample of interest.

PACR is implemented via *μManager*^[19] and its method is similar to *CRsCR*, with the main difference being that the former implements a piece-wise affine transform and the latter CR-splines. This piece-wise affine is generated by optimizing the distances between isolated localization pairs via *affineLLS* optimization, which means that for N localization pairs, N *affineLLS* transforms have been generated. The localizations of this dataset are then used as control-points for the registration map. When applying the piece-wise affine transform, the average transform of a number of neighbouring control-points will be used. This should result in a non-uniform transform that is dependent on the location as it takes the average of k -nearest neighbours (kNN) transforms per localization. As we expect the aberrations to be continuous over the FOV, we too expect this approach to be continuous to some extent, although some small scale discontinuities will always arise where kNN switches between neighbours. This approach offers a lot of freedom in estimating single affine transforms, meaning that this method is also prone to errors induced by false localization pairs, which induce discontinuities in the registration map. Therefore, *PACR* is restricted by the correct linking of localization pairs. On the other hand, CR-splines offers a lot less freedom as constricts the registration map to be $C1$ continuous. Therefore, we expect the influence of false localization pairs to be greatly reduced in *CRsCR*.

3.1. Validating *CRsCR* parameters for Fiducial Markers

Before comparing the two algorithms, we would first like to validate the parameters derived in figure 2.8 on the fiducial marker data. For this we register both channels via *affineLLS* optimization. The *affineLLS* optimization has been validated in appendix figure C1, but has been left out of the main text as it is not of interest. After *affineLLS* optimization, we are left with a dataset on which parameters of the CR-splines (which have been derived in figure 2.8) will be tested. The x-and-y-errors of the fiducial marker data after linking and optimizing via affine LLS in contour plot is shown in figure 3.1C.

Following the same method as described by figure 2.8C, we again test all 8 different TensorFlow optimizers for a varying number of iterations on the fiducial markers data. Thereafter the pair-distance distribution will be fitted according to equation 2.10. The resulting error curve is given in figure 3.1C,

from which we can derive that only the *AdaGrad*, *AdaDelta* and *SGD* algorithm are stable and convergent. In figure 2.8 we chose *SGD* as optimizer, but in figure 3.1 we see that *AdaGrad* converges faster. This difference however is small, and therefore we decide *SGD* is a valid algorithm to use for the fiducial marker data. We choose to run the algorithm for 300 iterations, to ensure that an optimum has been reached.

Next, we will investigate the learning-rate on the fiducial marker data, which is again tested as previously described in figure 2.8D. From this figure we can see that any learning-rate in between $5e-5$ and $3e-3$ gives the best possible results, where we trade off a bit of the bias in order to get better precision. Lastly, we would like to verify the gridsize of the CR-splines. We want to have a gridsize sufficiently small to correct local aberrations, but not so small as the CR-splines will have too little data per control-point and the registration map will over-fit the data, incorporating localization error into the map. As a rough hand-rule, we would want to have about 100 localizations per control-point, which according to figure 3.1B (second figure) would mean a gridsize of about 6000nm. To test this rule, we optimize the fiducial marker dataset for variable gridsize, after which we can fit the resulting pair-distances by equation 2.10. The resulting bias and precision are plotted for both the estimation and testing data in figure 3.1F. We see that for a small gridsize, the CR-splines will over-fit the data, resulting in a registration map that does not represent the system, and we see the estimation and testing precision diverge. Notable however is that the bias seems to still decrease with the gridsize for both estimation and testing data, which would indicate that even when over-fitting, the CR-splines still represent part of the global aberration throughout the system. We elect our CR-spline gridsize to be 7500nm, which should not result in any over-fitting while still correcting on a local level.

3.2. CRsCR vs PACR Results

Now that the parameters have been validated, *CRsCR* can be applied to the fiducial marker data, after which we will compare the results to *PACR*. We first take a look at how the pair-distances are distributed after optimization via both *CRsCR* and *PACR* in figure 3.2, together with a fit by equation 2.10. We can see that *CRsCR* reaches a precision that lies close to that of *PACR*, although a bit worse.

As noted in section 2.3.2, equation 2.10 is not accurate when estimating the bias. Therefore, we further investigate this bias in figure 3.4C, where we compare the x-and-y pair-distance distribution of both the estimation and testing data after optimization via both *CRsCR* and *PACR*. If we look at the precision and bias of the estimation data for both *PACR* and *CRsCR*, we see that there is almost no bias present, and both algorithms reach almost the exact same precision and bias.

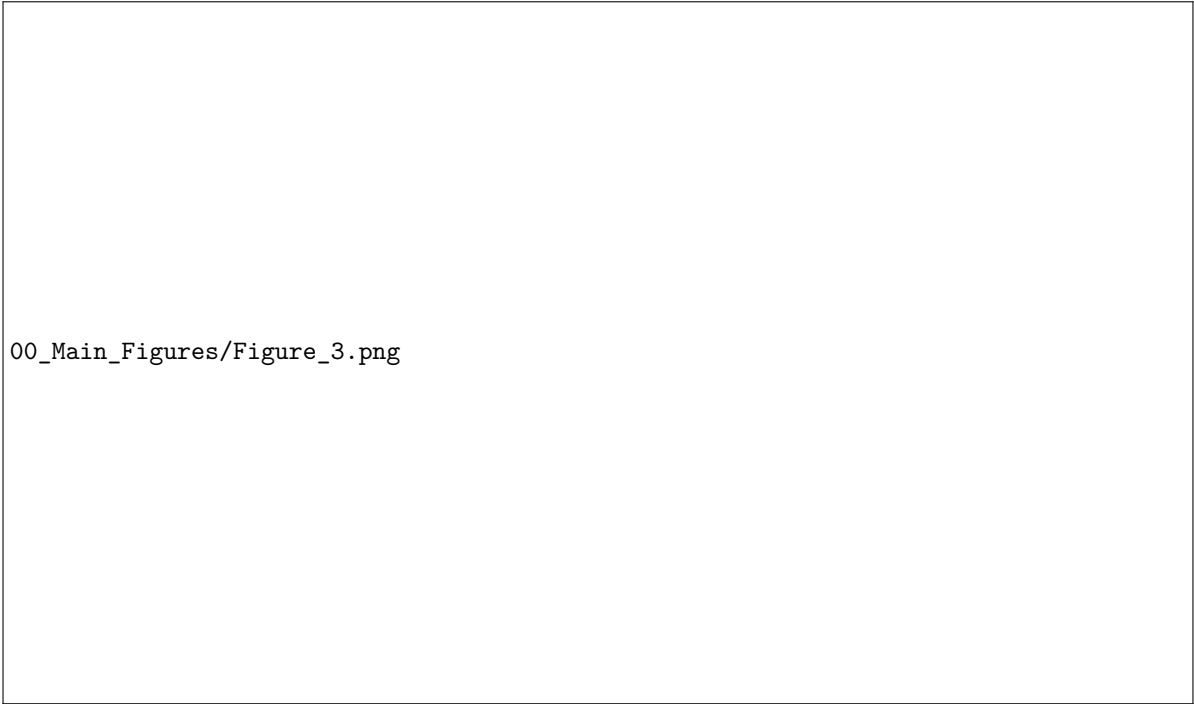
For the testing data, we see that both *CRsCR* and *PACR* has some bias left in the x-error and a lower precision in the y-error. Looking at the distribution of the x-and-y errors over the whole FOV in figure 3.4D, we see that the estimation data seems to have a small and uniformly distributed error. For the testing data we can see that there is a general bias over the whole FOV for the x-error, and there seems to be some global aberration present in the y-error, which contributes to the lower precision. If we compare this result to the results in appendix figure C1, we see that also after optimizing via *affineLLS*, the same kind of bias and global aberration is present in the testing data. This would either indicate the optimization problem getting stuck on a local minimum for both the *affineLLS* and the CR-splines. Another possibility is that some extra aberration has been created in between measurements of the estimation and testing data. This seems possible as some time has passed in between measurements, as also the sample of interest has been imaged in the meantime.

Most notable however, is that both methods reach results that lie closely together. In figure 3.4D, we can also see that the patterns in the x-and-y-error are similar. As discussed previously, the piece-wise affine transform uses k-nearest neighbours to base a registration map on. When the true mapping is continuous, which we assume it to be, we expect that for the right values of k the registration map would converge to the true mapping, and become smooth to a certain extent. An example interpolating a set of data using *kNN*, together with the interpolation via CR-splines, has been shown in figure 3.3. One disadvantage of using the piece-wise affine transform, is that it is prone to errors induced by outliers. An example of such a case is given in appendix figure F1. Outliers are often filtered out, and therefore this effect is negligible when filtering is done correctly.

We would like to investigate how the map created by CR-splines resembles the map created by the piece-wise affine transform. For this, a mapping is estimated on the fiducial marker data using the CR-splines. We then create a grid of 400x400 localizations on which we can apply this map to get

00_Main_Figures/Figure_2_Niekamp.png

Figure 3.1: Optimizing the Catmull-Rom Splines Channel Registration parameters for the fiducial marker dataset^[1] described in section 3, after it has been aligned via Affine Linear Least Squares optimisation. **(A)** The sample of fiducial markers. This sample is sufficiently sparse to allow easy linking of points. This sample has been used to create **(B)**. The complete fiducial marker dataset containing both channels in different colours. As we zoom in, we can clearly see the deformations between the localizations. **(C)** The bias and precision vs the amount of iterations given for the 8 different TensorFlow optimizers (Adam, AdaDelta, Ftrl, RMSprop, AdaGrad, AdaMax, Nadam and SGD). CR-Splines are used to trained on half of the data for different optimizers to investigate the convergence of the optimizer. The resulting registration map will be tested on the other half of the data, after which the resulting distribution of pair-distances will be fitted by equation 2.10. The resulting fitted bias is shown for each of the optimizer against the amount of iterations. **(D)** The error vs the initial learning-rate. The CR-splines with said learning-rate are initialized and trained for 300 iterations on half of the data to create a map that optimizes the pair-distances. This registration map is then used on another half of the data as a test. The resulting pair-distance distribution of both the estimation data (in blue) and the testing data (in red) after registration is fitted by equation 2.10 of which the resulting bias is plotted together with its precision against the initial learning-rate. From this figure we can derive that an initial learning rate higher than $5e-5$ needs to be used, but lower than $5e-3$ as the algorithm then maps to infinity. **(E)** the error vs the CR-spline gridsize. The CR-splines with said learning-rate are initialized and trained for 300 iterations on half of the data to create a map that optimizes the pair-distances. This registration map is then used on another half of the data as a test. The resulting pair-distance distribution of both the estimation data (in blue) and the testing data (in red) after registration is fitted by equation 2.10 of which the resulting bias is plotted together with its precision against the gridsize. As we can see, the precision of both the estimation and the testing data start to diverge below the 5000nm, as overfitting starts to take over.



00_Main_Figures/Figure_3.png

Figure 3.2: CRsCR on the fiducial marker data^[1]. **(A)** The sample of fiducial markers. This sample is sparse, which means it allows easy linking of points without too much false pairs being produced. The sample is moved using a piezo stage over a 20x20 grid to create **(B)**, the complete fiducial marker dataset containing both channels in different colours. As we zoom in, we can clearly see the deformations between the localizations as patterns in localizations are shifted **(C)** CRsCR is used to train a registration map which is then tested on a second dataset. The resulting distributions of both the estimation and the testing dataset for both CRsCR and PACR are then compared in histograms and fitted by equation 2.10 to get an estimation on the bias and the precision of both methods.

04_Results/Figures/ksplinescomp.png

Figure 3.3: Comparison between Catmull-Rom splines (LLS) interpolation, and piece-wise affine interpolation techniques that take the average of the k -nearest neighbour (kNN) values, both fitted on semi-random data given in blue. Both curves have been shown in black, and consists of 500 individual points. The control-points of the splines have been given in red. The curve of the moving-average starts to become smoother as k increases. This is because the randomness per point has less effect on the piece-wise affine as more points are used.

insight in the local dependencies of the map. We also use the piece-wise affine method that uses a kNN algorithm to estimate the map using the fiducial markers as control-points to create another map. We also apply this map for variable k to the grid of 400×400 localizations and compare it to the map of the CR-splines in appendix figure F2 to get an insight into the discontinuities of the piece-wise affine in comparison to the CR-splines.

We can see that for low values of k , the transform is extremely discontinuous, as large jumps in error exist throughout the FOV. As k is increased, we see these jumps in error getting smaller, and the x -and- y -error of the piece-wise affine transform starts to resemble that of the CR-splines. However, the piece-wise affine transform used in appendix figure F2 uses only one value of k , whereas *PACR* searches for a maximum of 100 neighbours within a distance of $2\mu\text{m}$. *PACR* also places a minimum value of 10 neighbours on the algorithm to counter over-fitting of data. For these reasons, we expect the piece-wise affine transform to converge on the same precision as that of the CR-splines.

CR-splines should not be as prone to false localization pairs as the piece-wise affine. In appendix figure D we tested the influence of these false pairs by giving a random subset of localization pairs a 30nm offset. As can be seen from this figure, even having 10% false pairs does not have a strong detrimental effect on the precision, and only after 30% false pairs does the precision become worse than 10nm. We have also researched *CRsCR* for data without filtering in appendix E. Again, although the precision is a bit lower than with filtering, a low precision has been reached that converges upon the CRLB.

3.3. Investigating the Cramer-Raó Lower Bound

Niekamp compares its results to a localization error of 1.4nm. This is unrealistic as the localization error and thus the Cramer-Raó Lower Bound (CRLB) should be a function of the intensity, position and the background instead of a constant. Furthermore, *μManager* does not return a trustworthy uncertainty in position when applying SMLM. Therefore, it becomes near impossible to compare fiducial marker data after registration via *PACR* and *CRsCR* of figure 3.2 to a theoretical CRLB. We are however able to apply SMLM via the *ThunderSTORM* plugin of ImageJ, which automatically returns the CRLB per localization. We can take the average CRLB over all positions and multiply it by $\sqrt{2}$ (as derived in appendix B) to get the CRLB of the co-localization data.

The SMLM data calculated via *ThunderSTORM* cannot be implemented via *μManager*, so we are not able to compare *PACR* to its CRLB. We are however able to register this data with *CRsCR* and com-

00_Main_Figures/Figure_3bias.png

Figure 3.4: Investigating the bias of figure 3.2 **(A)** The sample of fiducial markers used by Niekamp (^[1], and figure 3.2) as described in section 3. This sample is used to create **(B)**. The complete fiducial marker dataset containing both channels in different colours. As we zoom in, we can clearly see the deformations between the localizations. **(C)** *CRsCR* is used on the fiducial marker data to train a registration map which is then tested on a second dataset. The resulting distributions of both the estimation and the testing data are then compared in histograms for both the x-and-y-error and fitted by equation 2.10 to get an estimation on the bias and the precision of both methods. As comparison, the parameters of the curve that forms after we use *PACR* are also shown (the actual curve is left out as both curves overlap almost completely). **(D)** The x-and-y-errors of both the estimation and testing data after applying *CRsCR* or *PACR* are given over the whole FOV in contour plot. The error seems to be uniformly distributed, with only the y-error of the testing data seeming to have some residual bias left.

pare that to its CRLB. As the CRLB of co-localization is only dependent the CRLB of SMLM, the CRLB of *PACR* should be the same as with *CRsCR*. In figure 3.2 we saw that *CRsCR* and *PACR* have similar precision, and therefore calculating the CRLB for *PACR* would be redundant.

The fiducial marker data localized by *ThunderSTORM* and registered via *CRsCR* is compared for both the x-and-y error and the absolute error for both the estimation and testing data to its CRLB in figure 3.5. We can see that the fit of the estimation data approaches the CRLB, but does not completely reach it. This is probably due to the fiducial marker color centers not perfectly overlapping^{[1][10]}, which would result in deviations from its CRLB. This effect should be isotropic and would therefore not result in a bias but would lower the precision.

It is important to note that *ThunderSTORM* was not able to reach similar levels of precision as *μManager* does, which results in a lower co-localization precision when comparing figure 3.5 to figure 3.2. This however is due to the workings of SMLM algorithms and is therefore beyond the scope of this thesis.



Figure 3.5: Applying *CRsCR* on the fiducial marker data of figure 3.2 and comparing it to its Cramer-Raó Lower Bound (CRLB). Instead of creating the SMLM localizations via *μManager*, we now have used the *ThunderSTORM* plugin of ImageJ. This is because *ThunderSTORM* does allow calculation of the CRLB per localization, of which the average is taken to calculate the total CRLB. Given are the x-and-y-error of the estimation data (**A**) and the testing data (**B**) together with their Gaussian fit and the CRLB, and the absolute error of both the estimation (**C**) and testing (**D**) data together with a fit by equation 2.10 and the CRLB curve with $\sigma = \sqrt{2\sigma_{CRLB}}$ which is centered around the fitted bias to compare the precision.

Results Föster Resonance Energy Transfer Data

We would now like to take a look at more complex data. For this we use a Föster Resonance Energy Transfer (FRET) microscopy dataset. FRET is a phenomenon in which one donor fluorophore excites an acceptor fluorophore to a higher state via dipole-dipole coupling, after which it emits a certain wavelength of light. The frequency of FRET interactions is inversely proportional to the sixth power of the distance between donor and acceptor, which means that most excitations take place when the donor lies within 10nm of the acceptor. Our sample consists of one set of acceptor fluorophores, which have been anchored in place, and one set of donor fluorophores, that move around in a Brownian motion. Donors however move too fast to be viewed via the microscope, and therefore we can only view the donors that emit light while they are bound to an acceptor. A visual representation of the FRET interaction has been shown in figure 4.1A, in which the FRET excitation spectra shows up in the first channel, and the emission localizations of the donors, shows up in the second channel.

In this chapter, we will take a look at how to register common SMLM data via *CRsCR*. Although the workings of FRET interaction are very interesting, the most essential aspect of this data is the difference between the previously used fiducial marker data. Fiducial markers show up at the same location for both channels, which means that the localizations should correspond between channels. For FRET data, but also for many other different types of data, this is often not the case. In this case we still expect the FRET localizations to lie close to the donor localizations, as the donors that we can viewed by microscope are bound to the acceptors. Different than earlier now is that one FRET localization can be linked to multiple donor localizations.

4.1. Registration of Clustered Data

This data still has the problem that it is dense, and therefore hard to correctly link the localizations. Therefore, we first use *ImageJ* to filter for donor localizations that are clustered. This is done to ensure the donors we register are bound to an acceptor.

Now we would like to correct the drift via *DME*^[20]. Furthermore, we align the channels using a cross-correlation optimization method, ensuring corresponding localizations lie close together. Both the FRET localization channel and the filtered donor localization channel, after clustering, drift correction and rcc registration, are shown in figure 4.1B.

4.1.1. affineLLS

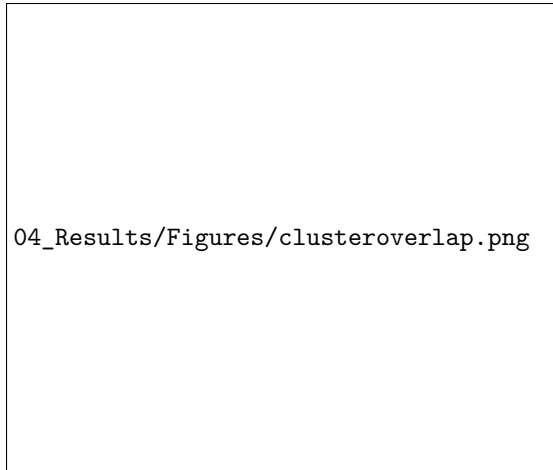
When we look at figure 4.1B, but also the pair-distribution distances shown in appendix figure C2, we see that most clusters after optimization via cross-correlation lie within 200nm of each other. This however means localizations still cannot be correctly linked with high accuracy. This is because, first of all, the clusters still lie too far from each other, which means it is possible localizations within one cluster will be linked to localizations within the wrong cluster. Furthermore, the dataset is dense, which means that unless corresponding localizations lie close together, linking will result in a big number of

00_Main_Figures/Figure_4.png

Figure 4.1: CRsCR on Fluorescence Resonance Energy Transfer (FRET) data after drift correction via $DME^{[20]}$. **(A)** Förster Energy Resonance Transfer (FRET). **Left:** A donor fluorophore is excited by an incoming photon and emits a certain wavelength of light, creating the first channel. **Right:** A donor fluorophore is excited by an incoming photon and transfers this energy to the acceptor via FRET which then emits its own wavelength of light, creating the second channel. **(B)** The clustered FRET data containing both channels in different colours, together with a zoomed in subset. All data has been grouped into clusters, this however means that a lot of data that did not belong to these automatically selected clusters has been deleted. This is primarily noticeable for the donor localizations of channel 1, which should be more uniformly distributed over the FOV. **(C)** After channel registration, the clusters seem to overlap. Four clusters have been zoomed in on, with the total size of the zoom being 300x300nm. **(D)** The data has been split into two random subsets, one of which will be used to train the registration map on and the other to test the map on. The data has been split per cluster, as the data is dense and splitting the total data would result in two somewhat equal subset. The distribution of absolute pair-distances of the estimation data and the testing data after registration respectively are given together with a fit from equation 2.10 and compared to their CRLB of $\sigma = \sqrt{2\sigma_{CRLB}}$. Note that the precision is a bit lower than the CRLB. As the donors are able to move through the FOV we expect the donor localizations lie not exactly at the FRET localizations, but would most likely be distributed around them. This would result in the actual CRLB being a bit larger than we have calculated before, and can explain the difference between the two lines.

false localization pairs.

For clustered data, not aligning in such a way that corresponding localizations lie close together before linking will result into an error of equal to the radius of one of the clusters, which is further described in the figure on the right. To circumvent these problems, we will not base the affine map on the linked localizations, but on the cluster center of masses (CoMs). This is done because we expect the CoM data to be sufficiently sparse to allow clean linking, and because the clusters are not sufficiently aligned to allow the clean linking of individual localizations. We then implement an *affineLLS* optimization on the linked CoMs, after which we expect the clusters to precisely overlap. In appendix figure C2, the distribution of linked cluster CoM distances before and after *affineLLS* optimization have been compared. We see from the noisy plot that *affineLLS* optimization has not been able to fully optimize all cluster CoMs. We therefore assume all cluster CoMs that are further than 120nm apart to be false cluster pairs.



Optimizing clusters that do not overlap. **(A)** Localizations within the left clusters are linked to those of the right cluster that are closest, which all lie on the edge. Optimizing will result in **(B)** where the distance between the cluster has an error of half the radius of the second cluster ($\approx 50nm$).

4.1.2. CR-splines

Now that we made sure clusters overlap, we are still left with the problem that the data is too dense to allow for clean linking of localizations. To try to force the maximum amount of correlation between the data while linking, we will only link localizations that lie within the same frame. This should give a higher probability that the localization of the FRET interaction is linked to the localization of the emission event from a donor that has caused this FRET interaction.

Previously, we decided that cluster CoMs that are further than 120nm apart should not be linked. Therefore, we only link localizations that lie within this distance from each other. We will link the donor localizations to the FRET localizations, which means that multiple donors will be linked to a single FRET-interaction. This results in 11.3% of the donors and 12.9% of the FRET localizations to be filtered out. The resulting localization pairs should now be able to be used in registration via CR-splines.

4.2. Validation of CRsCR on FRET data

After optimization, we can compare the data by looking at how the clusters are positioned over each other, as is done in figure 3.2C. The distribution of absolute errors, together with its fit by equation 2.10 and its CRLB, have been shown in figure 3.2D. We can see that a low bias has been reached, and the precision lies close to the CRLB. Note that the precision is a bit lower than the CRLB. As the donors are able to move we expect the donor localizations to not exactly lie at the FRET localizations, but would most likely be distributed around them. This would result in the actual CRLB being a bit larger than we have calculated before, and can explain the difference between the two lines.

We would like to investigate the results further by looking at the x-and-y-error of both the estimation and testing dataset, which we do in figure 4.2. From this we can see again that the precision improves, but there is still some bias left in both directions of both the estimation and testing data. This time, comparing it to appendix figure C2B, we see no reason why CRsCR do not reach zero bias. This would indicate that the CR-splines optimization process has become stuck in a local minimum.

Next, we would like to validate the behaviour of the error depending on the CR-spline gridsize used. For this we again run the CRsCR 10 times for different gridsizes. The resulting distribution will then be fitted by equation 2.10 to get the bias and the precision for both the estimation and the testing data. The results are plotted in figure 4.2B. Now the behaviour of the bias is stranger, as it seems that for too low of a gridsize, the bias of the estimation data seems to converge on the testing data, although we

would expect it to keep decreasing. We can also see that over-fitting by having a small gridsize does not result in a worse precision or bias for the testing data. This is due to the testing and estimation data being split from each other per cluster. This means that there will be no overlap in between the two data sets. Over-fitting the estimation data will then result in local errors in the registration map, but do not negatively impact the rest of the FOV.

To find possible outliers, we would like to take a look at the error over the FOV. This is done in figure 4.2C. From this figure we can see that the error seems to be distributed uniformly throughout the FOV. In some locations however, the error seems to diverge somewhat stronger. These are probably clusters that have been falsely linked. As these errors are smaller than 20nm, we expect the effect that these clusters have on the overall precision to be negligible.

We further zoom in on both the estimation (figure 4.2D) and testing (figure 4.2E) data. From this we see that the estimation data clusters are centered well. The testing data is not as consistent. This would indicate the error to be locally dependent, and thus would indicate a limit for the CR-splines.

00_Main_Figures/Figure_4bias.png

Figure 4.2: Investigating the bias of figure 4.1 **(A)** CRsCR is used on the FRET data to train a registration map which is then tested on a second dataset. The resulting distributions of both the estimation and the testing data are then compared in histograms for both the x-and-y-error and fitted by equation 2.10 to get an estimation on the bias and the precision of both methods. The CRLB curve with $\sigma = \sqrt{2}\sigma_{CRLB}$ has also been added and centered around the fitted bias to compare the precision. **(B)** Investigating the Catmull-Rom splines optimization. CRsCR has been used for different spline gridsizes to estimate a map of half of the data, and the other half is used to test the map on. Both the registered estimation and test data are then fitted by equation 2.10. The resulting bias and precision are plotted against the spline gridsize, together with their uncertainties. We can see that the bias of the test data seems not to be strongly affected by the gridsize, probably because the Affine Linear Least Squares optimization already takes care of this. The precision seems to be affected a bit stronger, but only after using a gridsize that does not seem physically representative of any deformation. **(C)** The x-and-y-errors of both the estimation and testing data after applying CRsCR are given over the whole FOV in contour plot. The error seems to be uniformly distributed, with only some local outliers. **(D)** A zoom in on part of the estimation data after registration via CRsCR, and the same is done for the testing data in **(E)**. We can see that the estimation data aligns the clusters well, while in the testing data there seems to be some error for some clusters. As this does not happen for all clusters in the testing data, this should be a local error, and therefore this is probably a limitation by the Catmull-Rom Splines.

5

Discussion

Although, for both the fiducial markers and the FRET data, the *CRsCR* algorithm reaches low precision and bias, there is still a lot to improve. First of all, *CRsCR* does not consistently reach zero bias. Secondly, the algorithm is not applicable to all data. This is mainly due to the linking of localizations being a difficult process. Both problems will be discussed in the coming sections.

5.1. Bias after *CRsCR*

As noted before, equation 2.10 does not deduce the correct bias (although it does come close). For this reason, we would like to investigate the bias by only looking at the Gaussian distributions in figures 3.4C and 4.2A. We have noted that the bias does become lower than 5nm for both the fiducial markers and the FRET data. It however does not reliably reach sub-1nm bias. This would indicate the algorithm getting stuck on a local minimum. As this happens both after *affineLLS* optimization and CR-spline optimization (for both fiducial marker and FRET data) and for the piece-wise affine transform (for the fiducial marker data), this would indicate some local minimum on which the optimization gets stuck. An improvement on *CRsCR* would thus be to add a penalty to the LLS optimization such that it is able to overcome these local minimum.



Figure 5.1: Example of difference between the distribution described by equation 2.10, and a Gaussian distribution. The original distribution is a bi-variate Gaussian which has been shown in gray scale. Equation 2.10 reduces this 2D Gaussian to a 1D distribution by integrating the data that lie on the surface of a circle with radius r . This has been demonstrated for both the estimation and testing data of both the fiducial and FRET data.

5.2. Linking the Data

We have noted that one of the limits of *CRsCR* is the correct linking of data, especially for data which is dense. *PACR* circumvents this problem by using a sparse sample, that allows for clean linking, for multiple frames to create a dense set of data. This technique, although very precise, does require the use of fiducial markers in a experimental setup that cost a lot of effort. Our goal was to eliminate the need of these fiducial markers by implementing advanced computation techniques.

One of the ways to bypass the need of fiducial markers, would be to use some form of nearest neighbour algorithm in the optimization problem. One such method could implement a Minimum Entropy cost function together with a *kNN* algorithm as has been done for drift correction at minimum entropy^[20]. However, CR-splines are computationally heavy to implement due to the enormous amount of variables being present. Implementing some form of nearest neighbours would mean the computation time would become extremely high, and therefore this program would need to be implemented via parallel programming or some extra steps have to be taken to simplify the problem.

One could also focus on improving the linking of localizations process by insuring localization pairs to lie closely together before linking them. We have done this with the FRET data by first aligning the channels using a cross-correlation method. This method, although only able to utilize a translation transform, already brought localizations within a distance of 120nm of each other. This would indicate that using a cross-correlation method together with a more complex transform, like an affine, would bring the localizations even closer together before linking. However, implementing cross-correlation optimization for the affine transform would require a more advanced algorithm to optimize the 6 parameters.

Another method that was used for the FRET data to bring localizations closer together for linking, was the optimization of cluster center of masses (CoMs) via *affineLLS*. This technique of linking and aligning cluster CoMs obviously is limited to data that contains clusters.

To extend such a method to data that is not clustered we can also look at alignment algorithms that do not depend on cluster CoMs. Lorenz^[21] proposes one of these methods, in which B-splines are used to register channels and frames for pixel data only. Implementing this method would bring all corresponding localizations closer together, meaning the linking process would be improved greatly. Implementing this method into *CRsCR* would therefore greatly improve overall precision as less information is filtered out and less false localization pairs should be present.

Conclusion

In this thesis, a method called Catmull-Rom spline Channel Registration (*CRsCR*) that registers two channels in Single Molecule Localization Microscopy has been proposed. This method relies on linking localizations between channels to create localization pairs. The squared distance between localization pairs has been optimised analytically via an affine Linear Least Squares transform, after which Catmull-Rom splines interpolation will be used to optimise the remainder via a steepest descent algorithm.

The main goal of this thesis was to implement a method similar to that described in Niekamp (*PACR*^[1]), but without the use of the piece-wise affine transform. Instead, this paper focuses on implementing Catmull-Rom splines. A second goal was to implement an algorithm that does not necessitate localizations between channels to correspond to each other. This would eliminate the need to use fiducial markers to base a registration map on. Both goals will be discussed in the next sections.

6.1. *CRsCR* vs *PACR*

We have shown that implementing both *CRsCR* and *PACR* will result in a similar bias and precision that converges upon its CRLB. Looking at how the errors are distributed over the field of view, a lot of similarities are found between the two methods. This would indicate that both methods result in almost the exact same registration map.

After investigating both methods more closely, we have discovered that the piece-wise affine transform converges on the same continuous solution as Catmull-Rom splines do. As the piece-wise affine transform is dependent on the average transformation of a certain amount of neighbouring localizations, discontinuities will always be present. Often, these discontinuities and errors due to outliers are negligible as the true mapping is smooth, meaning there are no large jumps in data that can impose these discontinuities in the map. We have also shown that the piece-wise affine transform is prone to errors due to false localization pairs. These false localization pairs are able to impose discontinuities onto the registration map, and therefore they need to be filtered out. We have shown that because Catmull-Rom splines forces smoothness onto the registration map, the map is limited and therefore less prone errors due to false channel pairs. Therefore, *CRsCR* does not necessitate the filtering of false localization pairs to find a solution that converges onto the true mappings.

6.2. Implementing *CRsCR* Dense Data

We implemented a method that is able to register more complex data that does not allow easy linking of localization pairs. This method is dependent on aligning the channels before linking the localizations. This was done for clustered data by first aligning via cross-correlation and then transforming the cluster center of masses via affine linear least squares. This allowed the localizations to be optimized to lie within 120nm of each other, and therefore they could be linked and Catmull-Rom splines could be applied to further optimize the localizations to a precision near to the CRLB.

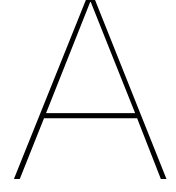
6.3. Recommendations

We have discussed that the primary limits in *CRsCR* lie within the process of linking localizations. To improve this process, it is recommended to align both channels to an extent before linking the localizations. A method that implements a version of *CRsCR* on clustered data has been proposed, but does not allow general use apart from clustered data. Therefore our main recommendation would be to improve the alignment of data before generating the localizations by following the method described by Lorenz^[21]. This method optimizes the pixel data via B-splines, which would ensure the channels to be better aligned before applying SMLM. After generating the localizations, *CRsCR* can be applied to reach close to optimal co-localization precision for different kinds of dense datasets that are not clustered.

References

- [1] Stefan Niekamp et al. "Nanometer-accuracy distance measurements between fluorophores at the single-molecule level". In: *PNAS* 116.10 (2019), pp. 4275–4284. DOI: 10.1073/pnas.1815826116. URL: <https://www.pnas.org/content/116/10/4275/tab-article-info>.
- [2] Mickaël Lelek et al. "Single-molecule localization microscopy". In: *Nature Reviews Methods Primers* 1.1 (2021), p. 39. DOI: 10.1038/s43586-021-00038-x. URL: <https://doi.org/10.1038/s43586-021-00038-x>.
- [3] Aryal Sagar. *Fluorescence Microscopy- Definition, Principle, Parts, Uses*. 2021. URL: <https://microbenotes.com/fluorescence-microscope-principle-instrumentation-applications-advantages-limitations/> (visited on 01/04/2022).
- [4] Carlos S. Smith et al. "An automated Bayesian pipeline for rapid analysis of single-molecule binding data". In: *Nature Communications* 10.1 (2019), pp. 1–9. ISSN: 20411723. DOI: 10.1038/s41467-018-08045-5. URL: <http://dx.doi.org/10.1038/s41467-018-08045-5>.
- [5] Carlos S Smith et al. "Probability-based particle detection that enables threshold-free and robust in vivo single-molecule tracking". In: *Molecular Biology of the Cell* 26.22 (Sept. 2015), pp. 4057–4062. ISSN: 1059-1524. DOI: 10.1091/mbc.E15-06-0448. URL: <https://doi.org/10.1091/mbc.E15-06-0448>.
- [6] Carlos S Smith et al. "Fast, single-molecule localization that achieves theoretically minimum uncertainty". In: *Nature Methods* 7.5 (2010), pp. 373–375. ISSN: 1548-7105. DOI: 10.1038/nmeth.1449. URL: <https://doi.org/10.1038/nmeth.1449>.
- [7] Sebastian Van De Linde. "Single-molecule localization microscopy analysis with ImageJ". In: *Journal of Physics D* 52.20 (2019).
- [8] Xiaowei Zhuang. *Super-Resolution-Microscopy*. [Online; accessed 12 februari 2021]. 2013.
- [9] Manuel Guizar-Sicairos, Samuel T. Thurman, and James R. Fienup. "Efficient subpixel image registration algorithms". In: *Optical Society of America* 33.2 (2008), p. 156. ISSN: 0146-9592. DOI: 10.1364/ol.33.000156.
- [10] Kim I. Mortensen et al. "Optimized measurements of separations and angles between intramolecular fluorescent markers". In: *Nature Communications* 6 (2015). ISSN: 20411723. DOI: 10.1038/ncomms9621.
- [11] Miklos Erdelyi et al. "Correcting chromatic offset in multicolor super-resolution localization microscopy". In: *Optical Society of America* 21.9 (2013), pp. 12177–12183. DOI: 10.1364/OE.21.010978.
- [12] T. Yan et al. "Computational correction of spatially variant optical aberrations in 3D single-molecule localization microscopy". In: *Opt. Express* 27.9 (Apr. 2019), pp. 12582–12599. DOI: 10.1364/OE.27.012582. URL: <http://www.osapublishing.org/oe/abstract.cfm?URI=oe-27-9-12582>.
- [13] Yiming Li et al. "Global fitting for high-accuracy multi-channel single-molecule localization". In: *bioRxiv* 7 (Jan. 2021), p. 2021.09.22.461230. DOI: 10.1101/2021.09.22.461230. URL: <http://biorxiv.org/content/early/2021/09/22/2021.09.22.461230.abstract>.
- [14] Taehwan Kim, Seonah Moon, and Ke Xu. "Information-rich localization microscopy through machine learning". In: *Nature Communications* 10.1 (2019), p. 1996. ISSN: 2041-1723. DOI: 10.1038/s41467-019-10036-z. URL: <https://doi.org/10.1038/s41467-019-10036-z>.
- [15] Daniel Sage et al. "Super-resolution fight club: assessment of 2D and 3D single-molecule localization microscopy software". In: *Nature Methods* 16.5 (2019), pp. 387–395. ISSN: 1548-7105. DOI: 10.1038/s41592-019-0364-4. URL: <https://doi.org/10.1038/s41592-019-0364-4>.

- [16] Hazen P Babcock. "Multiplane and Spectrally-Resolved Single Molecule Localization Microscopy with Industrial Grade CMOS cameras". In: *Scientific Reports* 8.January (2018), pp. 4–11. ISSN: 2045-2322. DOI: 10.1038/s41598-018-19981-z. URL: <http://dx.doi.org/10.1038/s41598-018-19981-z>.
- [17] Christopher Twigg. "Catmull-Rom splines". In: *Computer* 41.6 (2003), pp. 4–6.
- [18] L Stirling Churchman, Henrik Flyvbjerg, and James A Spudich. "A Non-Gaussian Distribution Quantifies Distances Measured with Fluorescence Localization Techniques". In: *Biophysical Journal* 90.January (2006), pp. 668–671. DOI: 10.1529/biophysj.105.065599.
- [19] Nico Stuurman et al. "Computer Control of Microscopes using μ Manager". In: *Curr Protoc Mol Biol* Unit14.20 (2010), pp. 1–22. DOI: 10.1002/0471142727.mb1420s92.Computer.
- [20] Jelmer Cnossen et al. "Drift correction in localization microscopy using entropy minimization". In: *Optics Express* 29.18 (2021), p. 27961. ISSN: 1094-4087. DOI: 10.1364/oe.426620.
- [21] Kevin S. Lorenz et al. "A multi-resolution approach to non-rigid registration of microscopy images". In: *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*. 2012, pp. 198–201. DOI: 10.1109/ISBI.2012.6235518.



The CRLB of the Imaging Model

The CRLB of the imaging model defined by equation 1.2 of a Poisson process with likelihood 1.4 as derived by *Smith 2010*^[6] is the inverse of the Fisher information matrix approximated by

$$I_{ij}(\theta) = \sum_k \frac{1}{\mu_k(x, y)} \frac{\partial \mu_k(x, y)}{\partial \theta_i} \frac{\partial \mu_k(x, y)}{\partial \theta_j}. \quad (\text{A.1})$$

In which the partial derivatives are defined by,

$$\frac{\partial \mu_k(x, y)}{\partial \theta_x} = \frac{\theta_{I_0}}{\sqrt{2\pi}\sigma} \left(e^{-\frac{(x_k - \theta_x - \frac{1}{2})}{2\sigma^2}} - e^{-\frac{(x_k - \theta_x + \frac{1}{2})}{2\sigma^2}} \right) \Delta E_y(x, y) \quad (\text{A.2})$$

$$\frac{\partial \mu_k(x, y)}{\partial \theta_y} = \frac{\theta_{I_0}}{\sqrt{2\pi}\sigma} \left(e^{-\frac{(y_k - \theta_y - \frac{1}{2})}{2\sigma^2}} - e^{-\frac{(y_k - \theta_y + \frac{1}{2})}{2\sigma^2}} \right) \Delta E_x(x, y) \quad (\text{A.3})$$

$$\frac{\partial \mu_k(x, y)}{\partial \theta_{I_0}} = \Delta E_x(x, y) \Delta E_y(x, y) \quad (\text{A.4})$$

$$\frac{\partial \mu_k(x, y)}{\partial \theta_{bg}} = 1. \quad (\text{A.5})$$

Defined by,

$$\Delta E_i(x, y) = \frac{1}{2} \operatorname{erf} \left(\frac{i - \theta_i + \frac{1}{2}}{2\sigma^2} \right) - \frac{1}{2} \operatorname{erf} \left(\frac{i - \theta_i - \frac{1}{2}}{2\sigma^2} \right) \quad (\text{A.6})$$

B

The CRLB of Registered 2-Channels

When applying SMLM the estimation of the true emitter position will always be prone to an inherent localization error which follows a normal distribution. We will consider both the x-and-y direction to have localization errors independent of each other. Therefore we will try to derive the CRLB for only one dimension. This means that, for true emitter location μ and localization error σ , the probability of estimating the localization at coordinate x equals

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (\text{B.1})$$

If both channels have not been deformed by aberrations, or have been registered perfectly, the true emitter position should be the same in both channels. However, the localizations in both channels show up at a different position due to the localization error. When we found a localization in the first channel at position x , the probability of finding the localization in the second channel at a distance d from the first will then be

$$p(x_2 = x + d | x_1 = x) = \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x+d-\mu)^2}{2\sigma_2^2}}. \quad (\text{B.2})$$

In which σ_2 is the CRLB of the second localization. This means that finding both localizations at a distance d from each other would equal equation B.2 times the probability of $p(x_1 = x)$ integrated over all possible positions x ,

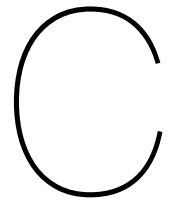
$$p(x_2 - x_1 = d) = \frac{1}{2\pi\sigma_1\sigma_2} \int_{-\infty}^{\infty} e^{-\frac{1}{2} \left(\frac{(x+d-\mu)^2}{\sigma_2^2} + \frac{(x-\mu)^2}{\sigma_1^2} \right)} dx \quad (\text{B.3})$$

$$= \frac{e^{-\frac{d^2}{2\sigma_2^2}}}{2\pi\sigma_1\sigma_2^2} \int_{-\infty}^{\infty} e^{-\frac{1}{2\sigma_2^2} \left(\left(1 + \frac{\sigma_2^2}{\sigma_1^2}\right) z^2 + 2dz \right)} dz \quad (\text{B.4})$$

where we used the substitution of $z = (x - \mu)$ to simplify the problem. This problem is not analytically solvable. However, if we assume all CRLBs to be equal to their mean ($\sigma_1 = \sigma_2 = \bar{\sigma}$), we simplify the problem to be

$$p(x_2 - x_1 = d) = \frac{e^{-\frac{d^2}{2\bar{\sigma}^2}}}{2\pi\bar{\sigma}^2} \int_{-\infty}^{\infty} e^{-\frac{1}{2\bar{\sigma}^2} (2z^2 + 2dz)} dz = \frac{1}{2\pi\bar{\sigma}^2} e^{-\frac{d^2}{4\bar{\sigma}^2}} \quad (\text{B.5})$$

Which is a normal distribution with zero mean and $\sqrt{2}\bar{\sigma}$ standard deviation.



Validating Affine Linear Least Squares optimization



00_Main_Figures/AppendixFigure_1.png

Figure C1: Investigating the efficiency of Affine Linear Least Squares (*AffineLLS*) on the fiducial marker data of figure 3.2. **(A)** The absolute pair-distance distribution before and after applying *affineLLS* optimization for both the estimation and a testing data, together with a fit curve following equation 2.10. **(B)** The x-and-y pair-distance distribution after applying *affineLLS* optimization for both an estimation and a testing data set, together with a Gaussian fit curve.

00_Main_Figures/AppendixFigure_2.png

Figure C2: Investigating the Affine Linear Least Squares (*AffineLLS*) on the cluster center of masses (CoMs) of the Förster Resonance Energy Transfer (FRET) data of figure 4.1 after rough optimization via cross-correlation alignment. **(A)** The absolute pair-distance distribution before and after applying *affineLLS* optimization for both the estimation and a testing data, together with a fit curve following equation 2.10. The estimation and testing data have been generated by splitting the localizations per cluster. **(B)** The x-and-y pair-distance distribution after applying *affineLLS* optimization for both an estimation and a testing data set, together with a Gaussian fit curve.

D

The Effect of False Channel Pairs

07_Appendix/Figures/linkingerror2.png

Figure D1: Investigating the effect of false linked pairs on the fiducial marker data from figure 3.2. For a percentage of the localization pairs, one localization is shifted in a random direction by 30nm to simulate a false channel pair. The data is then optimized via *CRsCR* after which the data is reloaded and mapped again to see the effect that the false localization pairs had on the true dataset. We have taken a random shift in all directions to mainly investigate the precision, if the error would have a preference (e.g. for misaligned clusters) it would mainly result in a stronger bias. We see the precision stays low for a small percentage of false channel pairs. However for larger percentages the precision decreases strongly. This seems unrealistic for sparse data, but becomes a problem for data that is too dense to allow clean linking of pairs.

Results without Filtering Pairs



Figure E1: *CRsCR* on the fiducial marker data^[1] without filtering. **(A)** *CRsCR* is used to train a registration map which is then tested on a second dataset. The resulting distributions of both the estimation and the testing dataset for both *CRsCR* and *PACR* are then compared in histograms and fitted by equation 2.10 to get an estimation on the bias and the precision of both methods. **(B)** The x-and-y-errors for both the training and the testing dataset given in distribution. We can see that without filtering *CRsCR* gives similar results to *PACR*. Figure 3.5 already did not implement filtering, so we have also shown that *CRsCR* without filtering still converges upon the CRLB.

07_Appendix/Figures/Figure_4_nofilter.png

Figure E2: *CRsCR* on Fluorescence Resonance Energy Transfer (FRET) data after drift correction via $DME^{[20]}$ without filtering. **(A)** The clustered FRET data containing both channels in different colours, together with a zoomed in subset. **(B)** After channel registration, the clusters seem to overlap. Four clusters have been zoomed in on, with the total size of the zoom being 300x300nm. **(C)** The x-and-y-error given over the FOV/ **(D)** The absolute error together with its fit by equation 2.10 and its CRLB. **(E)** The x-and-y-error as distribution together with a Gaussian fit and its CRLB. We see that without filtering, the optimization still reaches low precision and converges upon the CRLB.

F

Comparing distribution of CR-splines to the piece-wise affine transform

07_Appendix/Figures/ksplinescomp_outlier.png

Figure F1: Comparison between Catmull-Rom splines LLS interpolation, and moving-average interpolation techniques that take the average of the k -nearest neighbour (kNN) values, both fitted on semi-random data given in blue. Both curves have been shown in black, and consists of 500 individual points. The control-points of the splines have been given in red. The curve of the moving-average starts to become smoother as k increases. This is because the randomness per point has less effect on the piece-wise affine as more points are used. Three outliers have been added, and we can see that the moving-average interpolation technique is prone to errors whereas the Catmull-Rom splines LLS interpolation is not.

00_Main_Figures/AppendixComp.png

Figure F2: Comparing the discontinuities of the piece-wise affine transform from $PACR^{[1]}$ with the CR-splines from $CRsCR$. First, $CRsCR$ is used to register the fiducial marker data from figure 3.2. The CR-splines that are based on the fiducial marker data are then applied to a grid of 160.000 localizations, which is to be used as reference. The x-and-y-error of the reference image, given over the FOV, are shown in **(A)**. Now we will implement the piece-wise affine transform, which first creates a linear transform for each fiducial markers. These fiducial markers, together with their linear transform, are then used as control-points for the transform. Each localization will uses a k-nearest neighbour (kNN) algorithm to find the k closest control-points, of which the average linear transform is chosen as the transform for this point. In figure **(B)-(F)**, the x-and-y-error for the grid dataset after optimization via piece-wise affine for increasing values of k (respectively k=4, 8, 16, 32, 64) are shown. In figure **(G)**, a zoomed in version on the red square in figure **(A)** is given, and also for the optimization via piece-wise affine transforms for increasing values of k. It can be clearly seen that the more k increases, the less discontinuous the resulting distribution becomes, and the more it converges onto the solution of the CR-splines.