Department of Physics and Astronomy

University of Heidelberg

Master thesis

in Physics

submitted by

Frank Herrmannsdörfer

born in Stuttgart

2013

SimpleSTORM an efficient selfcalibrating reconstruction algorithm for single and multi-channel localisation microscopy

This Master thesis has been carried out by Frank Herrmannsdörfer

at the

Interdisciplinary Center for Scientific Computing

under the supervision of

Prof. Dr. Fred A. Hamprecht

SimpleSTORM ein effizienter, sich selbst kalibrierender rekonstruktions Algorithmus für Einzel- und Mehrkanal-Lokalisationsmikroskopie:

Mikroskopie ist ein wichtiges Werkzeug der Zell-Biologie. Die Höchstauflösende STORM Mikroskopie gewährt genauere Einblicke in Zellen, über die Abbe'sche Auflösungsgrenze hinaus. Dies wird durch die Aufnahme vieler Bilder mit jeweils weniger und damit leichter voneinander trennbaren Bildpunkten erreicht.
SimpleSTORM ist ein effizienter Algorithmus der von Schleicher (2011) entwickelt wurde und in dieser Masterarbeit weiterentwickelt wurde. Dabei lag der Fokus neben der Verbesserung des Algorithmus im Allgemeinen auf der Vereinfachung der Bedienung. Die Kernpunkte sind hierbei eine Selbskalibrierung und die Modellierung des Hintergrunds.
Durch die Selbstkalibrierung müssen keine Parameter vorgegeben werden, diese werden von dem Algorithmus selbst bestimmt. Dies erleichter es noch unerfahrenen Benutzern mit SimpleSTORM zu arbeiten. Die Modellierung des Hintergrunds erlaubt es Inhomogenitäten des Hintergrunds zu korrigieren Aussagen über die Güte der Signale zu treffen.
Die weiterentwickelte Colorcomposer Software ermöglicht die automatische Registrierung mehrerer Farbkanäle und die Bestimmung der Kolokalisation zwischen verschiedenen Kanälen erlaubt die Untersuchung von Wechselwirkungen zwischen einzelnen Molekülen.

SimpleSTORM an efficient selfcalibrating reconstruction algorithm for single and multi-channel localisation microscopy :

Microscopy is an important tool for cell biology. Super resolution microscopy yields deeper insights into cells, further than Abbe's resolution limit. This is achieved by taking many pictures with fewer and therefore easier seperable fluorescent markers within each picture.
SimpleSTORM is an efficient algorithm which was originally developed by Schleicher (2011). The imporovements of the general performance and usability will be described in this thesis.
Using selfcalibration it is no longer necessary to provide any parameter like the camera gain, camera offset or the expected signals point spread functions width. All cucial parameters are determined automatically but can also be manually adjusted. A model for the background was used with which inhomogenous backgrounds can be corrected, it was also possible to determine the quality of the signals found.

The improved Colorcomposer software provides easy auto alignment of different color-channels and measurments of colocalisation, which is an important measure for investigation of molecular interactions.

Contents

1 Introduction

How does a virus reproduce? Which proteins are involved in chemical synapses? How do proteins interact? These questions and many more arise in biology and related sciences. Microscopy is a powerful tool to answer these questions. However light microscopy is limited in spatial resolution due to limited diffraction, as described by Abbe (1873). Recently there have been developed different methodes to increase the resolution of light microscopy beyond the diffraction limit. To name a few: Photoactivated Localization Microscopy (PALM) Betzig et al. (2006), Stochastic Optical Reconstruction Microscopy (STORM) Rust (2006) or ST. These techniques use many images with sparsely distributed signals, coming from flourophors which are blured by diffraction, to determine their center with a sub pixel accuracy, instead of one image with all signals, which would be blurred and it would be impossible to find the true position of the individual flourophors.
The STORM method can also be used to investigate the distribution of different proteins within a cell. To do so each protein is labeled with different flourophores. Images can be aquired showing just signal from one kind of flourophore. However, to be able to seperate the different signals from the flourophores, the emission spectrum must be distinct. This leads to cromatic aberration which results in images that are distorted and thus can not be aligned easily.
Many research groups have developed their own software to process STORM data sets. But most of these programs usually need many parameters, like the point spread functions width or the camera parameters, must be set by the user to get reasonable results, therefore these programs are difficult to use for somebody who is not familiar with image processing or does not know the parameters influence on the results.

How to build software that is easy to use even without prior information about the data or knowledge of image processing?

Our answer to this question is SimpleStorm a software that calibrates itself. It estimates the camera parameters and the width of the point spread function of the flourophores. In contrast to many other software applications, no threshold is needed.
If more than one kind of fluorophore was used to stain the sample, the resulting images are distorted relative to each other. The colorcomposer tool was developed to do this task automatically and to analyse the aligned images. The colocalisation of the molecules of the different channels is a useful measure in biology to determine whether or not near molecules might interact. The colorcomposer software calculates

both global and local measures of localisation.

2 Theoretical background

2.1 Distributions

2.1.1 Poisson distribution

One very important probability distribution in physics is the Poisson distribution. It describes the results of "counting experiments" and is therefore very important for image processing as the pictures taken with a camera are in principle counts of photons reaching the camera. Photon counting noise is one important example.
Poisson distributions are just defined for integer values and the variance is the same as the mean value of the distribution. Another important attribute is the skewnes which is the inverse of the squarerot of the mean or variance and describes the assymetry.
The probability mass function is:

$$p(n, \mu) = \frac{\mu^n}{n!} \exp(-\mu) \tag{2.1}$$

2.1.2 Skellam distribution

The probability mass function of a Skellam distribution is a function of the difference between two Poisson random variables

$$p(k; \mu_1, \mu_2) = \exp(-(\mu_1 + \mu_2)) \left(\frac{\mu_1}{\mu_2}\right)^{k/2} I_{|k|}\left(2\sqrt{\mu_1 \mu_2}\right) \tag{2.2}$$

where n_1, n_2 are the Poisson random variables and $k = n_1 - n_2$. $I_{|k|}$ means the modified Bessel function of the first kind.
Mean μ and variance σ of the Skellam distribution are given by

$$\mu = \mu_1 - \mu_2, \qquad \sigma^2 = \mu_1 + \mu_2 \tag{2.3}$$

$$\Rightarrow \qquad \mu_1 = \frac{\mu + \sigma^2}{2}, \qquad \mu_2 = \frac{-\mu + \sigma^2}{2} \tag{2.4}$$

2.1.3 Approach using skewness of poisson distribution

For every pixel there is a set of multiple values in the set. This allows to calculate the different parameters individually for each pixel. One can calculate mean and

variance of the measured intensities $I_{\text{meas}}(i,j)$ and gets

$$\text{mean}(I_{\text{meas}}(i,j)) = g \cdot \text{mean}(I_{\text{true}}(i,j)) + o \tag{2.5}$$
$$\text{var}(I_{\text{meas}}(i,j)) = g^2 \cdot \text{var}(I_{\text{true}}(i,j)) \tag{2.6}$$

Assuming a Poisson distribution as the true intensity, mean and variance would be the same. Unfortunately the mean true intensities are unknown and it is not possible to determin g and o so far. For large mean Intensities μ the Poisson distribution becomes more and more similar to a Gauss distribution with the same mean. However, for small means, the Poisson distribution is not symmetric. The skewness s_p of a Poission distribution is the inverse of the square root of the mean $(\mu)^{-.5}$. It can also be directly calculated from data

$$s_p = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{\sigma}\right)^3 \tag{2.7}$$

The skewness is invariant to shift and multiplication with a constant. This means that the transformation caused by the camera gain and the dark current does not affect the skewness. This gives a third equation to solve for g and o.
This approach has very strict limitation at least for background pixels not to be too bright. If the mean of the true Poisson distributin is higher than roughly 30 the skewness gives due to noise no stabel results and it is impossible to determin the mean intensity in this way.

2.2 the data

The concept of direct stochastic optical reconstruction microscopy (dSTORM) Heilemann et al. (2008) is, to label interesting structures with fluorophors that can be exited using a laser with the appropriate wavelength. After a short time the flourophors emit a photon and go back to the unexited state or lose the ability to get exited, they bleach out. The datasets for dSTORM microscopy that we recieve from our collaborators from Bioquant are big datasets of several gigabyte in the Andor .sif format. Each file conains a stack of pictures, normaly between 1000 and 10000, taken consecutively with exposure times between 20 and 200 milliseconds.

Picture /refrawStorm shows a typical frame of raw data. In each frame there might be multiple flourophores visible at the same time. Due to the large magnification, beyond the diffraction limit, the almost pointlike flourophores appear as gaussian shaped signals, their point spread functions. The flourophores are either attached to the biological structures that are of interest or they form a cluster, called a bead.

Beads are larger and brighter than spots form only one flourophore and are used

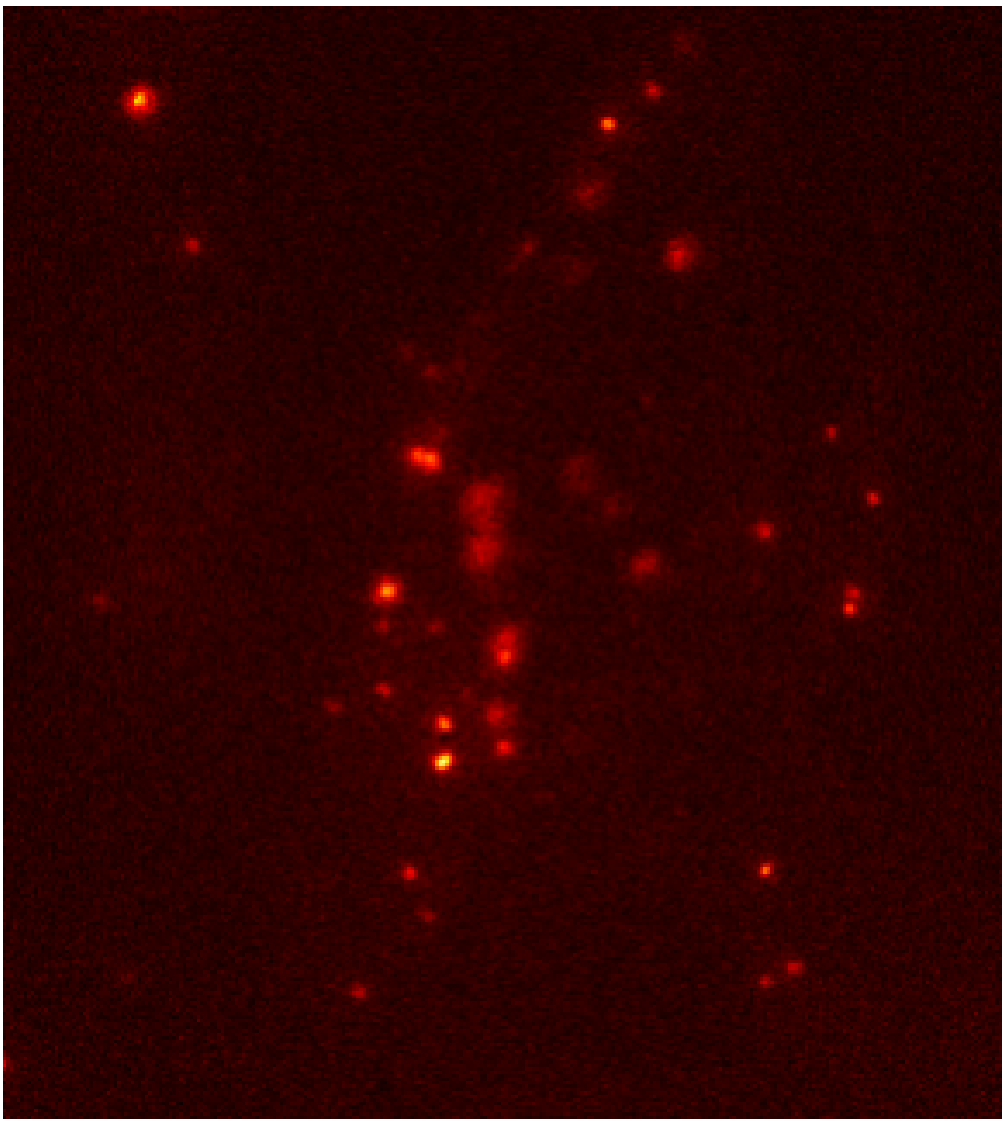


Figure 2.1: Raw image for dSTORM processing

to align multiple channels in the postprocessing step. Beads are designt to show up in every frame of the sequence at the same position. They are composed of flourophores of different color to be visible in every channel.

The other spots, bound to some proteins for example, are just lighting up for a very short time. This is the key aspect of dSTORM. Instead of one frame that shows all flourophores at the same time, thousands of frames are captured containing just a point spread functions per frame. This gives the possibility to determine the center of each point spread function with sub-pixel precision, and in the end when all points are displayed together in one picture, to an image with a resolution beyond the diffraction limit.

2.3 Transformations

2.3.1 Transformation to Poisson distributed signal

The images aquired from the camera show not the real intensities I_{true}, which result from the photon emission of the probe, but transformed ones I_{meas}. I consider two main reasons why the taken image differs from the true image, besides noise.
There is dark current which means that even a picture taken with closed shutter would get some intensity, even without any light hitting the sensor chip of the camera. This is a result of thermal movement of the atoms off the sensor chip and can be reduced by cooling. The dark current noise adds an almost constant value o to the output signal. Incoming photons create electrons via inner photoelectric effect. This electrons are collected for each pixel and might be amplified to get the final result. Assuming a linear relation between the number of incoming photons and the number of electrons created and a linear amplifier results in a factor g. This factor is multiplied with the number of photons captured during exposure time for each pixel.
If the gain factor g and the offset o are known the true intensity, the number of photons detected is:

$$I_{\text{true}} = \frac{I_{\text{meas}} - o}{g}. \tag{2.8}$$

2.3.2 Anscombe transformation

The Anscombe transform Anscombe (1948) is used to transform a random variable with a Poisson distribution into one with an approximatly constant standard deviation. The transformation is defined as:

$$A(x) = 2\sqrt{x + \frac{3}{8}}. \tag{2.9}$$

As one can see in figure 2.2 the Anscombe transformations result has for mean intensities greater than 4 a intensity independent standard deviation of one.

2.4 Estimation of camera gain

Given a sample prepared for STORM microscopy. A sequence of images captured from this sample will show some active flourophores, beads and also illuminated background that comes from flourophores that lie not in the focus plane. Assuming an inhomogenous background signal that follows a Poisson distribution, or even better almost homogenous background and beads both with a Poisson distribution in time with different mean values. If this Signal I_{true} is transformed in the following

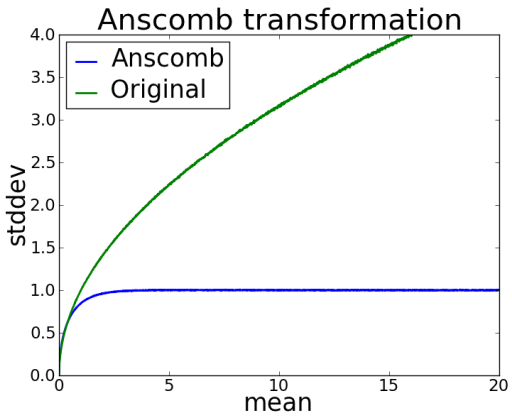


Figure 2.2: Standard deviation over mean intensities of different Poisson distributions

way:

$$I_{\text{meas}} = g \cdot I_{\text{true}} + o \tag{2.10}$$

This is the inverse transformation of equatione 2.8.
Considering two pixels with different Poisson distributions P_1 and P_2 with mean value and variances of this distributions λ_1 and λ_2. If this distributions are transfomed as given in equation 2.10 their mean and variance change like shown in equation 2.5 and 2.6. This gives the oportunity to determine the gain and offset using two or more pixels.

$$\text{var}(I_{\text{meas1}}) = g^2 \cdot \text{var}(I_{\text{true1}}) \tag{2.11}$$
$$\text{var}(I_{\text{meas2}}) = g^2 \cdot \text{var}(I_{\text{true2}}) \tag{2.12}$$
$$\text{mean}(I_{\text{meas1}}) = g \cdot \text{mean}(I_{\text{true1}}) + o \tag{2.13}$$
$$\text{mean}(I_{\text{meas2}}) = g \cdot \text{mean}(I_{\text{true2}}) + o \tag{2.14}$$

The values for $\text{var}(I_{\text{meas1/2}})$ and $\text{mean}(I_{\text{true1/2}})$ can be calculated from the data and can be used to get the gain as follows:

$$\frac{\text{var}(I_{\text{meas1}}) - \text{var}(I_{\text{meas2}})}{\text{mean}(\text{meas1}) - \text{mean}(I_{\text{meas2}})} = \frac{g^2 \cdot \lambda_1 - g^2 \cdot \lambda_2}{g \cdot \lambda_1 + o - (g \cdot \lambda_2 + o)} \tag{2.15}$$
$$= \frac{g^2 \cdot (\lambda_1 - \lambda_2)}{g \cdot (\lambda_1 - \lambda_2)} \tag{2.16}$$
$$= g \tag{2.17}$$

In the same manner the offset o can be calculated.

$$\text{mean}(I_{\text{meas1}}) - \frac{\text{var}(I_{\text{meas1}})}{g} = g \cdot \lambda_1 + o - \frac{g^2 \cdot \lambda_1}{g} \tag{2.18}$$

$$= o \tag{2.19}$$

3 CCD camera

3.1 Image acquisition

3.1.1 Photon sources with shot noise

The emission of photons is a random process that occures at unpredictible times. Therefore the number of photons passing through a plane is never constant but varies around some average value. The phenomena, that one can never determine exactly how many photons should hit the sensor chip of a CCD camera for example, is called shot noise. It playes a major role if the total number of photons is low, as from dark sources or with short exposure times of the camera.

3.1.2 Quantum efficiency

Quantum efficiency describes the fraction of photons that create a detectable electron in a sensor chip. The quantum efficiency is dependent of the wavelenght of the incoming photon. Photons with energies below the band gap can't produce a free electron that can be detected. The quantum efficiency has a maximum basically caused by two effects. The higher the photons energy the higher the kinetic energy of the freed electron, but it is absorbed earlier and can therefore recombinate with a electron hole more likely.

3.1.3 Gain

There are two different gain factors involved in the capturing process of a camera. First the electric signal for each pixel might be amplified. And there is also a gain factor that describes the proportionality between collected electrons and the digital number it is associated with.

3.1.4 Readout noise

The origin of readout noise is the amplifier. The aplification is never perfect, this means the exact number of electrons at the end of the amplification has some variation around the expected linearly increased value. There might also be some random signals of the electronics that add to the "true" signal. The readout noise is independent of the exposure time.

3.1.5 Dark current noise

Dark current noise is generated by the thermal movement of the atoms in the sensor chip. The movement of molecules and atoms is dependent of the temperature of the material, because of that dark current noise depends strongly on the temperature of the chip and can be reduced by cooling. Dark current noise generates electrons in the bins of each pixel even with closed shutter it is constantly increasing with time and follows Poisson statistics.

3.1.6 Quantisation

The signal must fit into the output color depth. It has to be rounded or truncated to fit in. This process introduces errors that can be seen as additional noise that is dependent on the intensity of the signal. High intensities are disturbed less relative to low intensies.

4 Data processing

STORM data from different cells or structures show many different features: there can be clusters of flourophores with a high density of spots, areas with low density, variable background in space and time, beads can be present or not. This section is about how the algorithm processes the data sets and how it is possible to find good settings that will work for all kind of input data.
All real world images shown in this thesis were aquired from members of Prof. Dr. Mike Heilemanns group.

4.1 Import and processing

The STORM data has usually a size of around 3 gigabyte. There are even larger datasets possible, so that it is important to work on smaller parts of the data, instead putting the whole dataset into memory. This is done using chunks of user defined size. The data is processed chunkwise, there is parallelisation for the frames of each chunk. This is possible because the signals in each frame are considered to be independent from each other.

4.2 Workflow

4.2.1 Chose parameters

At the begining the user has the option to set all important parameters, if no parameter is set the default ones are used and will give a good result because all crucial parameters are either determined from the data or set to reasonable values that work for every data set. The focus of the default parameters is to give a good result with no adjustment. This means parameters are chosen to produce almost 100 % precission. The loss of some points that are not detected using this conservative setting won't affact the final result as much as a lower precission will do.

4.2.2 Estimating camera gain and offset

First of all it is checked whether there exists a file containing settings for gain and offset from an earlier run. If this is not the case new parameters are estimated based on the first part of the data, usually 200 frames are sufficiant.
The method described by Hwang et al. (2012) is used to estimate the gain factor. For this methode a Skellam distribution is used. Each dataset is three-dimensional

where time is the third dimension. Therefore mean μ and variance σ^2, in time, can be calculated from the data for each pixel individually

$$\mu(i,j) = \frac{\Sigma_t(I_t(i,j) - I_{t+1}(i,j))}{n} \tag{4.1}$$

$$\sigma^2 = \frac{\Sigma_t(\mu - (I_t(i,j) - I_{t+1}(i,j)))^2}{n-1} \tag{4.2}$$

To determine the gain factor the Skellam parameter are plotted over the mean intensities. A straight line can be fitted and its slope gives the gain factor, the intersection is the offset.
The methode for the fit uses R R Core Team (2012). To suppress outliers, the fit is performed only on a compact subset of all points Rousseeuw and Driessen (1999), more details can be found in the manual of SimpleSTORM Kats (2013). This part of the algorithmn was developed and implemented by Ilia Kats.

4.2.3 Recursively adjusting gain and offset

After the estimation of gain factor and offset, the transformations described in 2.3.1 and 2.3.2 are applied and the background subtracted.
Due to the Anscombe transformation the background pixels of the image should only vary around a mean intensity of zero with a variance of 1. Therefore a histogramm of the pixel intensities is created. The after background subtraction the background pixels should contribute only to the lower intensities in the histogramm. A gaussian function is fitted to the histogramms values. This is done under the assumption that there is much more background in the image than signal or the intensities coming from signals are distributed over a larger range, so the gaussian for the background intensity distribution can be fitted correctly.
If the estimated value for the variance is too far off 1 the originally estimated gain factor is corrected, applied and the fit is done again. This is done until the background variance converges or the maximal number of iterations is reached. In this case the initial gain factor will be used and a warning printed to the screen.

4.2.4 Estimating the width of the point spread function

For a certain number of frames the Fourier transform is calculated and averaged. The result is called mean power spectrum. It can be used to estimate the variance of the point spread function of the signal. A two dimensional Gaussian functions Fourier transform is again a gaussian but with inverse variance. This relation is used to determine the variance of the point spread function in spatial domain, using the fit parameter for the variance in frequency domain.
The script for fitting the two dimensional Gaussian function was implemented by Ilia Kats.

4.2.5 Processing the data

Import Data

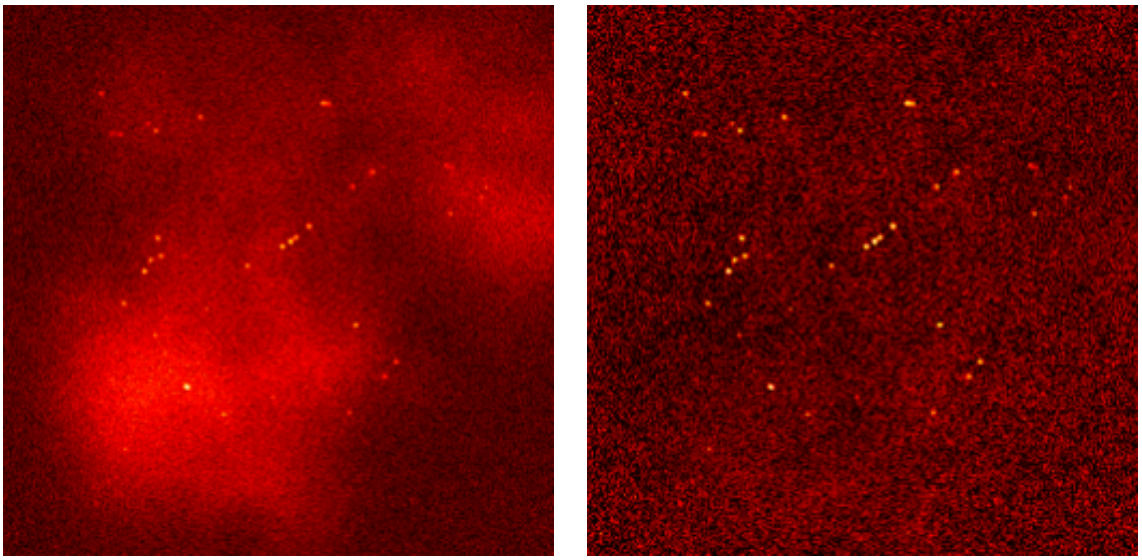
Storm data sets can consist of several thousand frames with resolutions up to one mega pixel per frame. This makes it necessary to break the data into smaller parts because otherwise it might be much larger as the RAM of an ordinary machine. Because of the background estimation it is not possible to process every frame completly independent as it was in the older version of this software Schleicher (2011). It is also faster with some datatypes to load a larger consecutive part of the dataset into memory.
This algorithm uses chunks of user defined size. There are some limitations to the chuncksize that are discussed later. The data set is split into parts of equal size in x- and y-dimensions and independently also in t-dimension. If this partition does not fit at the edge of the data set, the last chunks will be smaller.
The data is transformed to be Poission distributed and after that the Anscombe transform is applied. After the Anscombe transformation the background intensities have unit variance.
The implementation of the workflow using chunks was implemented by Ilia Kats.

Background estimation

For each chunk the median of the Anscombe transformated data is determined to get a robust estimate of the background value for this chunk. The Bspline interpolation implemented in vigra Köthe (2011) is used to get interpolated values for the full resolution of the current frames. For this interpolation three chunks in both spatial and temporal domain have to be available. Therefore the maximal chunksize in t-dimension must not be larger than a third of the total stacksize, the same holds for the chunksize in x and y direction which must not exceed the spatial resolution of the image stack.
This background is then subtracted from the transformed data to give finally background pixels with zero mean and unit variance, both in xy- and in t-dimension. Figure 4.1 shows an frame with variable background before and after background subtraction.

Create mask for background suppression

With the given p-value from the settings a global threshold can be determined, because inhomogenities of background intensities has been removed. The threshold value is that intensity that is explained from a gaussian distribution with mean zero and variance one with the given p probability. This is possible because the background intensity follows such a distribution after all the applied transformations.
The threshold is applied to the current frame and stored as a mask. Due to the probability that background pixels intensitis might exceed the threshold the connected components of the mask are calculated. Pixels that belong to connected



(a) Original image (b) image after background subtraction

Figure 4.1: Effect of background subtraction on inhomogenous background. For the human eye no more points are visible but for computers it is much easier to find the bright spots in the background subtracted image, because a global threshold can be applied.

components with too few members are discarded. The idea behind using connected components is, that there is a probability for a background pixel to be brighter than the threshold, but it is unlikely that two neighbouring pixels exceed the threshold in the same frame and even more unlikely that three of them does. Therefore the number of pixels necessary for not discarding a connected component is set to three at least. If the width of the point spread function is very large the number of pixels can be set depending on the psf width. But to suppress background pixels efficiently it has to be larger than three.

Filter data and finding maxima

To imporve the accuracy of the spot detection the transformed signal is convolved with a two dimensional gaussian function with the previously determined or user set width. The convolved image will further be used to find the maxima. Each maxima found is tested to be covered by the mask or discared otherwise. A region of intrest around the remaining maxima is interpolated to a higher resolution. In the interpolated region it is searched for maxima for the last time. This maxima will be detected with super resolution.
To determin the signal-noise-ratio the unfilterd and uninterpolated pixelintensity is used.

Table 4.1: Comparismn of results created using almost no smoothing or Wiener filter on high density data. While the accuracy almost the same the number of detections and the scores are higher for the unsmoothed data. For the evaluation the evaluation software from Biomedical Imaging Group (2013) were used.

	intercetions	Jaccard	F-Score	Precision	Recall	RMSE
Gaussian filter width = 0.01	16955	19.99	33.26	81.10	20.92	27.21
Wiener filter	14480	17.06	29.15	79.19	17.87	27.28

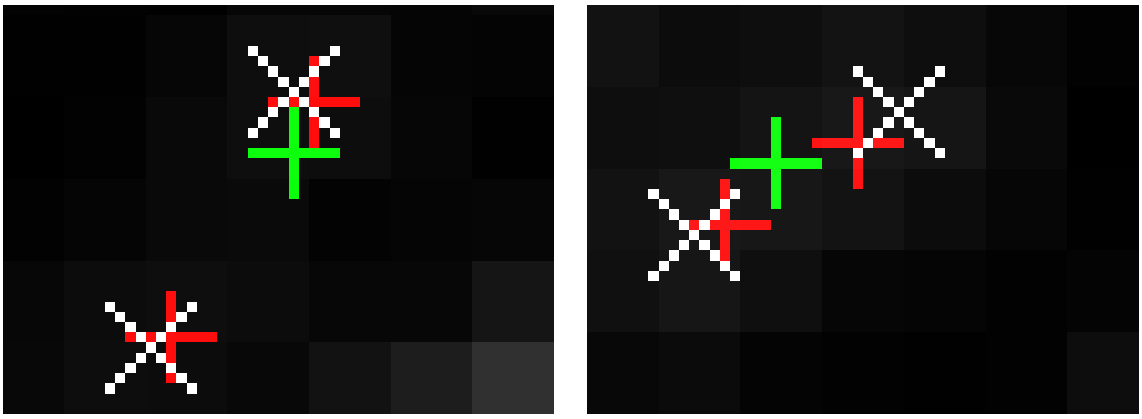
Quality control for detections

Especially in data sets with a high density of spots it can happen, that two spots are near each other and the point spread functions overlap. It may happen that instead of two maxima just one maximum will be detected right between the true ones, as it can be seen in 4.2. This leads to large errors in the localisation. To avoid this a threshold for the asymmetry of the spots can be set.
The calculation of the asymmetry was already implemented by Joachim Schleicher Schleicher (2011).

4.3 Comparison with older version of the SimpleStorm algorithm

4.3.1 Adjustable filter width

If two or more point spread functions overlap, applying a smoothing filter can lead to merging point spread functions. This becomes a problem if the two distinct maxima of the original unsmoothed psfs form a new maxima inbetween. This leads to just one detection somewhere between the true maxima. The number of merging psfs increases with greater filter widths. For high density data it might be better to use a filter with smaller width and therefore with less accuracy, but with less merging psf. In total this might give a better result depending on the number of incorrectly merged PSFs. This is the reason why the filter ing was changed to use just a Gaussian filter instead of the Wiener filter originally used. The effect of the smaller filter width can be seen in 4.2. The red crosses indicate detections found by the new SimpleStorm version, the blue crosses show the estimated positions from the previous version of SimpleStorm and the white x marks the groundtruth. The predictins by the newer version are not perfect but better than the prediction of the older version. The scores of the two prediction can be seen in 4.1. Both results have almost the same accuracy, but they differ in the scores. There are two effects canceling each other out related to the accuracy. Less merged PSFs increase the accuracy, but the positive effect of filtering with the appropriate filter, as shown in 5.5 or in 5.6, is also lost.



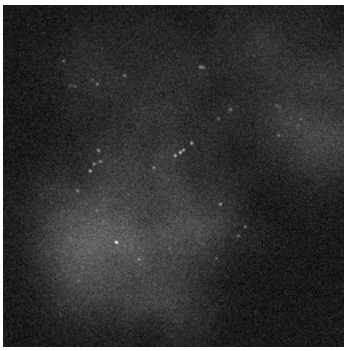
(a) Advanced setting widget (b) Easy setting widget

Figure 4.2: This pictures show the difference a smaller filterwidth has. The red crosses show detections found with a filter width of 0.01, the blue crosses the results using a Wiener filter. The white x marks the groundtruth.

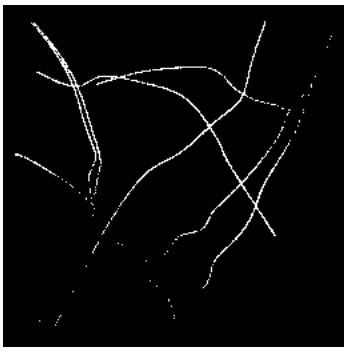
4.3.2 False positive suppression

In the previous version the background was determined by first estimating a baseline, the minimum of the current frame was taken. This baseline was subtracted and the resulting image smoothed with a gaussian filter of widht 10. The smoothed image was subtracted from the original image to give a background free image. This works fine for subtracting background with variations larger than the filters width, but gives as a result just intensities without any information about the variability of the background intensities. If for example the resulting intensity, after background subtraction is 5, this can either mean it is definitly signal if the variance of the background in the original image was very small, but it can also mean nothing, if the variance of the original background was 20, the probability that the difference of 5 between the original image and the smoothed one has a high chance to result from the background variation.
In the older version of SimpleSTORM the intensity of a candidate, that was considered to be signal, was checked by comparing its intensity after the background subtraction with its intensity before. If the maximums intensity was at least twice as high as the subtracted background it was taken for further processing, otherwise discarded. If the background has a high variability in the spatial domain, the baseline gets a value that is too low for the regions with a higher mean intensity. For this regions the background that is subtracted has high values and therefore the maxima are discarded because the resulting intensity is lower than the twice the background. This phenomena can be seen in figure 4.3. The old version of SimpleSTORM just findes about 8000 spots while the new version finds more than 44000. This results in a better reconstructed image that shows the underlying structure in a better way.

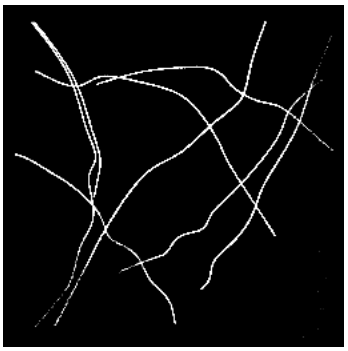
The great advantage of the newer version of SimpleStorm is its model for the background. For each pixel the probability to be signal can be determined, based



(a) Typical frame showing variable background intensities



(b) Result old SimpleSTORM



(c) Result new SimpleSTORM

Figure 4.3: This pictures show the drawback of the old background treatment. On the left a typical frame with variable background is shown. The old version of SimpleSTORM discards many detections in the regions of high background intensity. The new SimpleSTORM software discards detections based on their signal-to-noise ratio and is less affected by variable background.

on its signal to noise ratio (SNR). Using the number of connected components of each cluster, false positive detections caused by bright background pixels can be suppressed. This is a great advantage over background suppression methodes that work just on intensities.

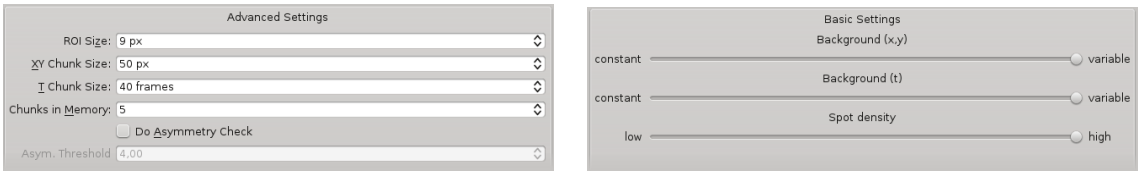
4.3.3 Comparable results based on the signal-to-noise ratio

The accuracy of the maxima detection relies on the signal to noise ratio. With a given SNR the correct standard deviation of the localisation error can be used for further calculations. This is not possible if just an intensity is saved because without information about the backgrounds variance the reliability of the detection can't be estimated.
To save the signal-to-noise ratio is also an advantage when results are compared that result from either different cameras or different settings or a different environment that might lead to a higher background variability.

4.4 New graphical user Interface (GUI)

4.4.1 Input widget

The new GUI for SimpleStorm was designed because where are many new features. Figure 4.5 shows the new design. The GUI was mainly designed by Ilia Kats. In principle there are three categories of parameters. The first category tells the program which upsampling factor shall be used or what the pixel width of the input



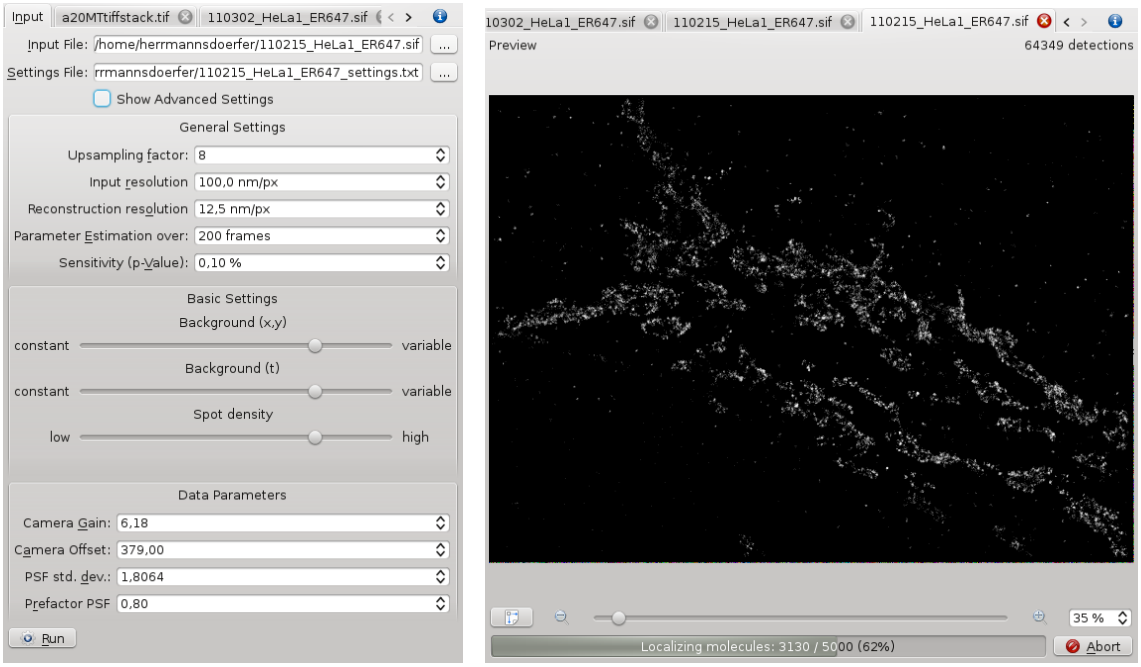
(a) Advanced setting widget (b) Easy setting widget

Figure 4.4: There are two different ways to set the parameters important for the algorithm. In the picture on the left the standard way of setting parameters can be seen. On the right, there are sliders that can be adjusted between two extrems each. The program sets the value for the parameters in a way to produce the best results for the selected attributes of the data set.

data is given in nanometers. This are informations that can be chosen by any kind of user without knowing anything about the algorithms the SimpleStorm software uses. The most challenging parameter of this section is the alpha value that sets the sensitivity for false detections. If the user doesn't know what this means or which values gives the best results, he or her can stick to the default setting and alternate it after the run.
The next category of parameters is about region of interest (roi) widths for the estimation of the point spread function, and chunk sizes. There are two different ways to set this parameters (see figure 4.4). One way is to give values for all parameters. This is difficult without understanding the influence of this parameters on the algorithm. Therefore there is also a second way to set this parameters. The user should know some properties of the data that shall be processed, such as is the spot density high or low, is there variable background in time and space or not? Depending on the sliders position the best parameters are set automatically, how this is done will be described later. With this second option the user can process his or her data, treating variable background or dense data without deep insight or understanding of the used algorithms.
The last catagory of parameters describes which camera gain and offset was used, the width of the signals point spread function and a prefactor that can be used to alter the estimated gain.

4.4.2 Result widget

After the run button at the lower left edge of the input widget was pressed, a new tab opens. In this widget the reconstructed image is shown. At the bottom there is a progress bar that displays at which processing step the program is currently and its progress. Buttons to zoom in and out or to fit the displayed image best into the window. On the lower right there is the stop/save button. It either stops the program if it is still runing or opens a dialog to save the result image and the coordinates of the detection if the program has already been stoped.



(a) Input widget (b) Result widget

Figure 4.5: The new gui design. On the left is the window for selecting input file and parameters show. On the right the result widget showing the procession of a data set in progress.

4.4.3 Easy parameter selection

As mentioned above with the sliders for background and spot density it is easy to set reasonable parameters without knowing its influence on the algorithms. The sliders can be placed inbetween two extrem cases, each.
The background sliders have direct influence of the corresponding chunksize. A more constant background gives better results with larger chunks. This is the case because the median of all pixels in a chunk is used to estimate the mean value of the background. Therefore the chunk have to contain more background pixels than signal otherwise the mean value will be overestimated. With large chunks this assumption is satisfied. But the smaller the chunksize the more likely it is that a cluster of points lies in the chunk and because of that the median gives to high values. On the other hand, the chunksizes should be in the range of the changes in background. The best chunksize is therefore in the range of the variable background. The sliders position sets the chunksizes linearly to a value that lies inbetween the minimal chunksize possible and the largest chunksize possible for the data set. The minimal possible chunksizes are 3 pixels and the largest chunksize possible is the the half of the shorter border in x and y dimension and the number of of frames over which the parameters are estimated divided by the number of chunks in memory in t dimension.
In general the more dense the dataset is the more likely two neighbouring PSFs are merget due to the filtering, as showed in 4.3.1. There are two ways to avoid inaccurate detections. One is less filtering to avoid merges the second one is checking the symmetry of the detected spots. Merged spots become more and more asymmetric the further the two true centers of the PSF lay apart from each other. A high asymmetry is a good indicator for a poorly detected spot. The slider about the spot density influences the prefactor for the estimated sigma and the value for the asymmetry checks. It sets the threshold for the asymmetry in the same way the sliders for background work, within appropriate limits, with higher values for less dense data sets. Also the prefactor is set to values between 1 for sparse data and zero for dense data.

5 Check of the assumptions

The whole workflow depends on some assumptions about the data or the results of certain transformations. This assumptions have to be met at least roughly to ensure the correctness of the results.
This section gives proofs for most of the assumptions like accuracy of the PSF scale estimation or that the matched filter of the matching size compared to the PSF performs best.

5.1 Calibration measurement

As described by Newberry (1998) the gain can be determined using the mean and variance using data with different mean intensities. Therefore a calibration dataset was aquired. The temperature and the gain factor were set to be like the settings usuall used. A sample with cells was prepared for the Storm measurement. First eleven time series were taken, with opend shutter and with increasing exposure time from 0 milliseconds up to 1000 milliseconds. Afterwards the same procedure were repeated with the shutter kept closed. Figure 5.1 shows the results for the first 6 exposure time. All the points lie on a straight line. The gain factor determined from the calibration measurement was 3.9, the offset 315. This offset is too small. An other methode was used to determine the offset. There were also an other series of images taken with closed shutter. This series shows the response of the camera without any light around. The mean value of the shortest exposure time were taken as offset. The value is 380.

5.2 Correction to Poisson distributions

It is very important for the whole workflow that the first transform results in background intensities that follow a Poisson distribution. To test this a real world image was transformed with parameters taken from the calibration measurement described in the previous section. The offset was estimated from the minimal intensity of the raw image. The histograms for two generic background pixels are shown in figure 5.2. A Poisson distribution with the mean of the pixels intensities is also shown in the figures. This histograms were aquired using the pixel intensities of one pixel for 3000 frames. The histogram matches the expected Poisson distribution.

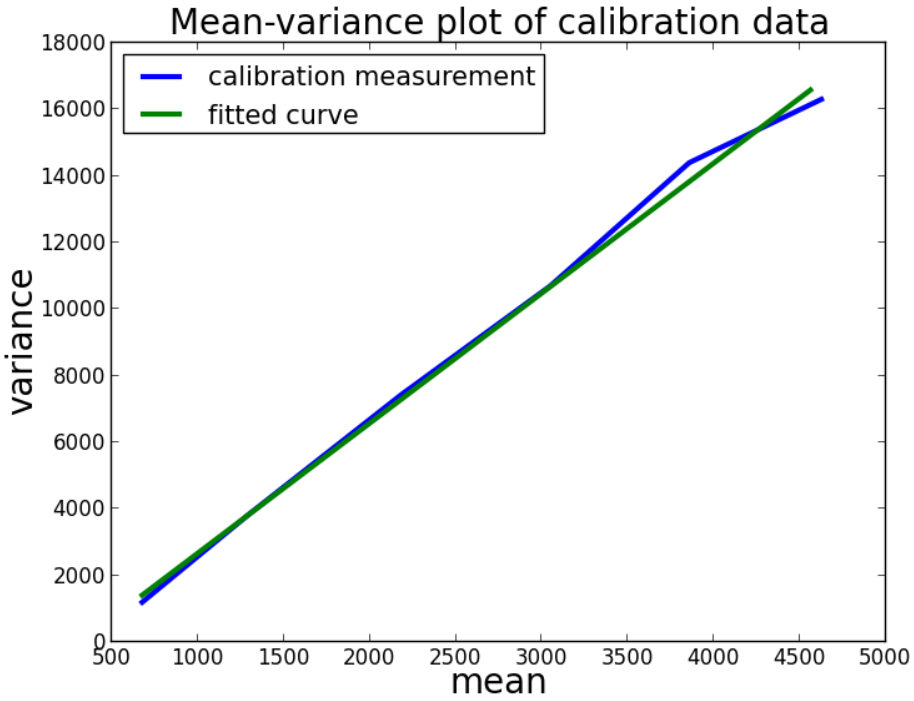
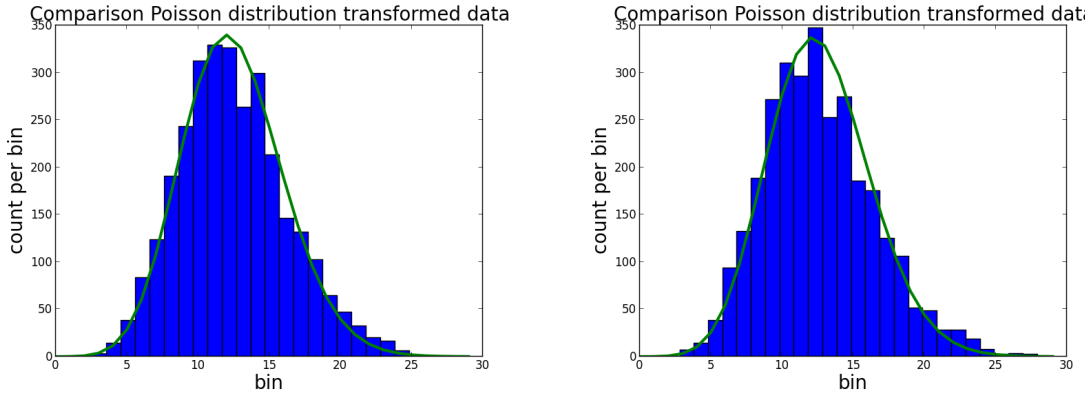


Figure 5.1: Result of the calibration measurements. The gain determined from this variance-mean plot is 3.9.



(a) Tubulin2 frame 10

(b) Tubulin2 frame 1010

Figure 5.2: This pictures show how well the histograms of background pixels taken over the first 3000 frames, follow a Poisson distribution. For the gain the parameter from the calibration measurement of $g = 3.9$ were used. The offset was estimated from the minimal intensities of the original image.

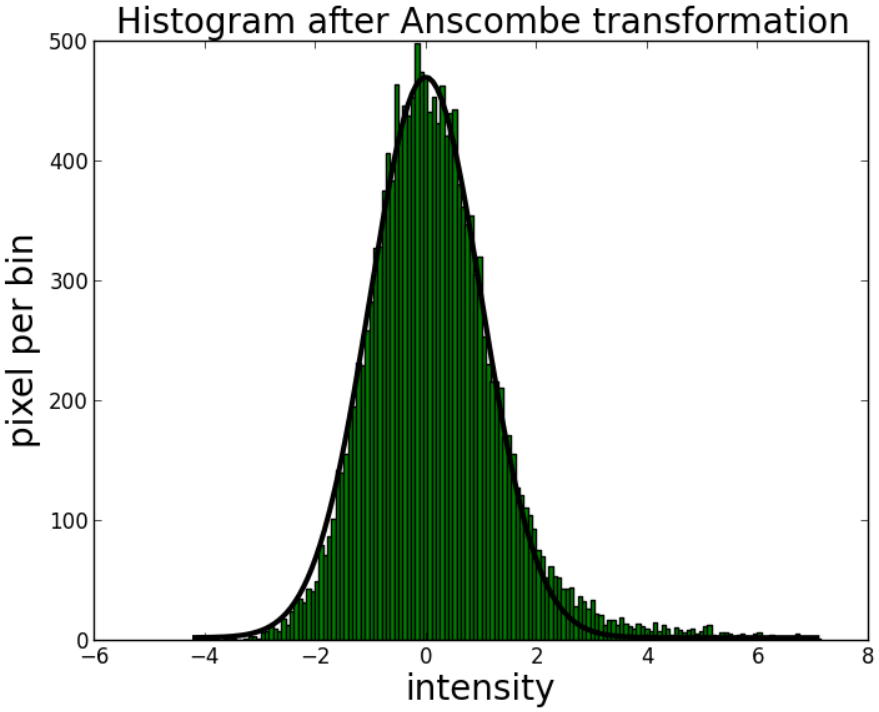


Figure 5.3: After the Anscombe transformation the background pixel intensities should be distributed with mean zero and variance one. This figure shows the histogram of the pixel intensities and a fitted gaussian with variance one and mean zero.

5.3 Result Anscombe transformation

The same data set used in the previous section was used. After the transformation described in 2.3.1, the Anscombe transformation was aplied and the background subtracted. Figure 5.3 shows the histogram of the intensities of one randomly chosen frame and a Gaussian with zero mean and unit variance. The histogram fits very well to the Gaussian. The tail exceeding the Gaussian on the right side can at least partially explained by the signal that is present in the image and does not follow the Gaussian distribution, but has higher intensities.

5.4 Accuracy of detection

Unfortunately the position of the flourescent molecules can't be detected perfectly. There are three main contribution to the error in detection.
On the one hand, there is the problem of finding the maximum in a noisy signal. Due to noise the pixel next to the true maximum might gain some intensity and be therefore brighter.
On the other hand, the position is deteted by upscaling the pixel grid and inter-

polation. After that the maximums position of the upscaled grid is taken as the resulting position. This gives an error from roughly a 12th pixelwidth. This error becomes less important with higher upscaling factors.

Because there is no groundtruth for the real data, one has to produce test data and groundtruth. This was done simular to the method described by Grüll et al. (2011). The difference was, that instead of generating just one spot per frame a different number of spots were simulated for each frame. For different signal to noise ratios, a dataset with 40 times 40 pixels and 1000 frames was created. Each frame containing one, three or five point spread functions. The position of the spots was determined beforehand, to use the same spots for each signal to noise ratio.

To determine the accuracy the standard deviation between the true position of the signal and its detection were used. For data sets with more than one spot per frame, it was searched for the best match within a certain distance around the true position. If a pair of true spot and detection was found, both were removed for further matching of the remaining signals. In the end the averaged standard deviation of the detection relative to the true positions were calculated as follows:

$$\text{std. dev.} = \sum_i \sqrt{\left(\mathbf{p}^{\mathbf{i}}_{\text{true}} - \mathbf{p}^{\mathbf{i}}_{\text{detec}}\right)^2} \tag{5.1}$$

i runs over all found pairs of groundtruth and detections, $\mathbf{p}$ describes the two dimensional spatial vector of the groundtruth or detection respectivly.

The results can be seen in 5.4. It can be seen, that the more spots are present per frame the harder it is to detect them properly. This is a result of the fact that spots that lay near each other might be detected as just one spot, what gives rise to higher errors in the detection accuracy.

5.5 Matched filter is best filter

For denoising the image a matched filter is used. Since we assume the point spread function to be gaussian, a two dimensional gaussian filter with the estimated size of the point spread function is used. The following calculation shows why this is the best choice for the standard deviation.
This calculation is similar to the calculation in Köthe (2007). The assumptions are, the point spread function is gaussian shaped with a true standard deviation of σ_{PSF}, the image contains white gaussian noise with mean zero and unit variance, the applied filter for denoising is a gaussian filter with standard deviation σ_{filter}.
The image f is composed of gaussian filter convolved with the signal s and the above

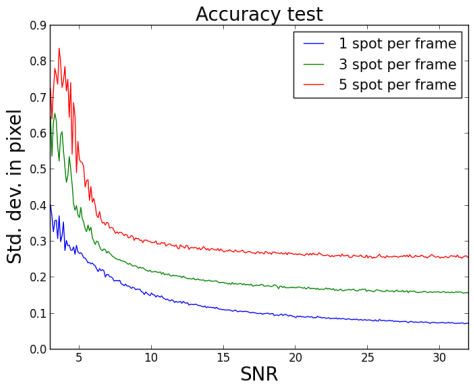


Figure 5.4: Result of the accuracy test. For datasets with one, three or five point spread functions per frame, evaluated for different signal to noise levels. The more dense the spots are the less accurate the detections are.

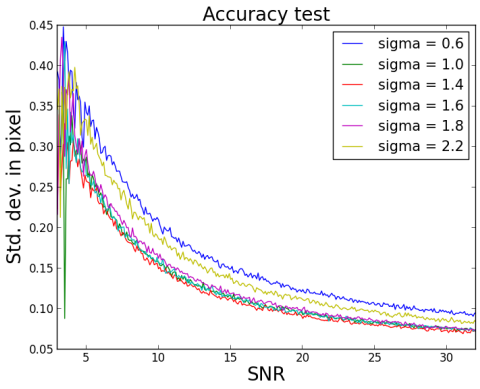


Figure 5.5: Result of the accuracy test for different sigmas for the gaussian smoothing. One data set was processed with different sigma values and evaluated for different signal to noise levels. The true standard deviation of point spread function of the simulated data is 1.4.

mentioned white noise n. s and n are describing the filtered signal respectivly noise.

$$f = s + n \tag{5.2}$$

Without noise the first derivative of f which is equal to s in the noise-free case vanishs at the center of the gaussian signal. This center can be set to the origin without losing generality. But noise shifts the zero crossing of the first derivatives to $\Delta x,\ \Delta y$. For now just one dimension is considered.

$$\frac{\partial f}{\partial x}(\Delta x, \Delta y) = \frac{\partial s}{\partial x}(\Delta x, \Delta y) + \frac{\partial n}{\partial x}(\Delta x, \Delta y) \qquad = 0 \tag{5.3}$$

Using Taylor expansion around $x = 0$ at $y = 0$ for the signal, yields:

$$f_x(\Delta x, \Delta y) \approx \underbrace{s_x(0,0)}_{=0} + s_{xx}(0,0) \cdot \Delta x + n_x(\Delta x, \Delta y) \qquad = 0 \tag{5.4}$$

$$\Rightarrow \operatorname{var}(s_{xx}(0,0) \cdot \Delta x) = \operatorname{var}(n_x(\Delta x, \Delta y) \tag{5.5}$$

$$\Rightarrow \operatorname{var}(\Delta x) = \frac{\operatorname{var}(n_x(\Delta x, \Delta y)}{s_{xx}^2(0,0)} \tag{5.6}$$

The result for the variance of the first derivative of white noise convolved with a

gaussian filter of width σ_{filter} can be found at Köthe (2007). It is:

$$\text{var}(n_x) = \frac{N^2}{8\pi\sigma_{\text{filter}}^4}, \text{ with } N\text{: noise std. dev} \tag{5.7}$$

The convolution of the Gaussian PSF with an other Gaussian results in Gaussian with combined scales.

$$\mathcal{N}(0, \sigma_{\text{PSF}}) * \mathcal{N}(0, \sigma_{\text{filter}}) = \mathcal{N}\left(0, \underbrace{\sqrt{\sigma_{\text{PSF}}^2 + \sigma_{\text{filter}}^2}}_{=\sigma_{\text{comb}}}\right) \tag{5.8}$$

The value of the second derivative in x-direction of the convolved PSF at (0,0) is:

$$\frac{\partial^2 S_0 \cdot \mathcal{N}\left((\mu_x, \mu_y), \sigma_{\text{comb}}\right)}{\partial x^2} = -\frac{S_0}{2\pi\sigma_{\text{comb}}^4} \tag{5.9}$$

Combining 5.6, 5.7 and 5.9 yields:

$$\text{var}\left(\Delta x\right) = \frac{\frac{N^2}{8\pi\sigma_{\text{filter}}^4}}{\frac{S_0^2}{4\pi^2\sigma_{\text{comb}}^8}} \tag{5.10}$$

$$= \frac{N^2 \cdot 4\pi^2\sigma_{\text{comb}}^8}{S_0^2 \cdot 8\pi\sigma_{\text{filter}}^4} \tag{5.11}$$

$$= \frac{N^2}{S_0^2} \frac{\pi\left(\sigma_{\text{filter}}^2 + \sigma_{\text{PSF}}^2\right)^4}{2\sigma_{\text{filter}}^4} \tag{5.12}$$

$$= \frac{N^2\pi}{2S_0^2} \left(1 + \frac{\sigma_{\text{PSF}}^2}{\sigma_{\text{filter}}^2}\right)^2 \left(\sigma_{\text{filter}}^2 + \sigma_{\text{PSF}}^2\right)^2 \tag{5.13}$$

The same variance can be calculated with respect to y, yielding a total variance that is square root of 2 bigger, because the variances are orthogonal. Therefore the standard deviation of the localisation is:

$$\text{std. dev loc} = \frac{N}{S_0}\sqrt{\frac{\sqrt{2}\pi}{2}} \left(1 + \frac{\sigma_{\text{PSF}}^2}{\sigma_{\text{filter}}^2}\right) \left(\sigma_{\text{filter}}^2 + \sigma_{\text{PSF}}^2\right) \tag{5.14}$$

To verify this results a test data sets has been created containing 3500 frames with one PSF with different scales. This data sets have been filtered with Gaussian filters of different sizes. After that the maximum for each frame was detected and compared with the true position. Figure 5.6 shows the accuracy for different PSF scales and filter widths.
As expected the best results were achieved using a Gaussian filter with the PSFs size. The larger the PSFs scale the more inaccurate the localisation becomes.

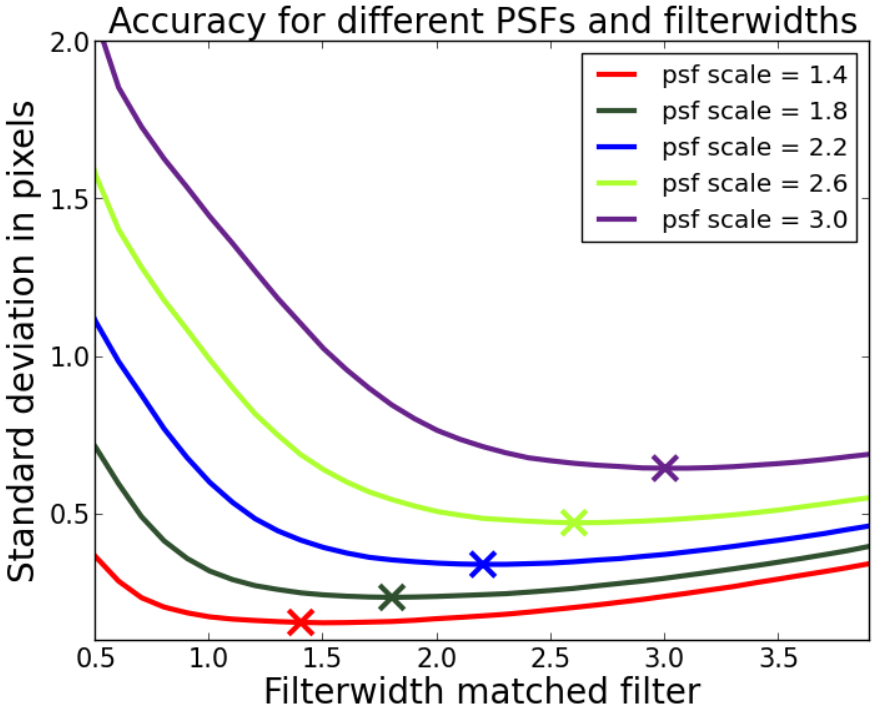
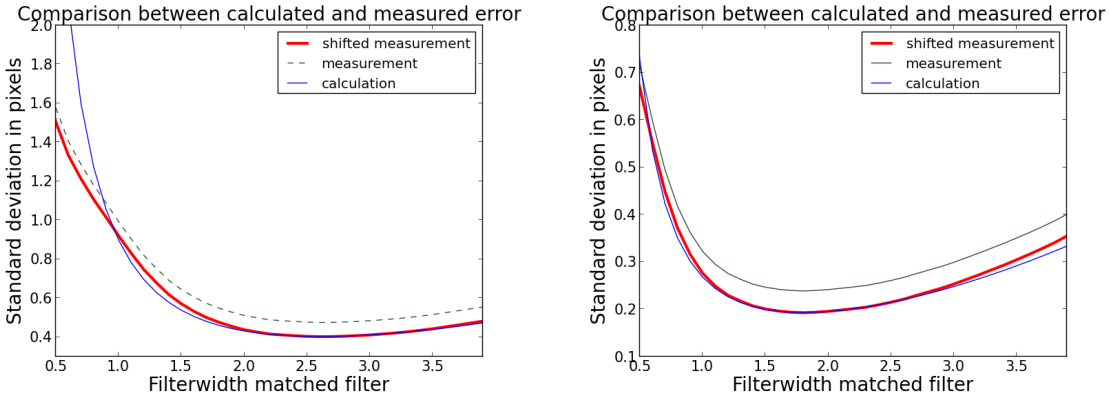


Figure 5.6: This figure shows the accuracy of detection achieved using different Gaussian filters for denoising. The cross marks the minima of the curves.

The figures in 5.7 show the measurement, a shifted measurement curve and the result of the calculation. The measured error was larger than the calculated error, therefore the shifted line is also shown to state that the curves shape match but there is an additional error that shifts the line.
This results show that the calculated standard deviation of the localisation can be used to estimate the localisation error of SimpleSTORM for further calculations or error propagation.

5.6 Test PSF estimation

The estimation of the width of the point spread function of the signal is very important. The localisation error increases with larger deviations of the matched filters width from the true width of the PSF. Figure 5.8 shows the results of an accuracy test. For this test data was created artificially. One PSF was simulated at a random location with a fixed width, poisson noise was added so that the signal-to-noise ratio was 10. Then this width was estimated by the SimpleSTORM algorithm. This was done with values for the standard deviation of the gaussian in a range from 0.2 up to 4. The plot shows that for standard deviations up to 2.5 the estimated value



(a) PSFs scale = 2.6 (b) PSFs scale = 1.8

Figure 5.7: Comparison between the measured and the calculated localisation error. The dashed lines show the unchanged measurements results, for better comparism the line was shifted to match the calculated error.

for the width is very close to the actual value. The wider the PSF becomes in the spatial domain the smaller it gets in the Fourier domain. This is the reason why the accuracy of the estimation is decreasing for larger widths. This effect can be reduced by choosing a larger region of interest for the power spectrums acumulation. The parameter chosen for the plot in figure 5.8 was the default value. The typical width of the PSF is in a range of 1 to 2, this is a range in which the estimation gives reliable results.

5.7 Bleaching signal

One assumption was that the backgrounds illumination is caused mainly form out of focus fluorophors and therefore Poission distributed. If this is the case the background illumination should decrease over time. Bleaching of flourophores describes the process in which the flourescent molecules change their conformation and lose the ability to emit light or the spectrum of the emission changes so that they can't be detected any more. All flourophores used for the STORM images we got are bleaching over time. This decay of the mean backgrounds intensity is shown in figure 5.9.

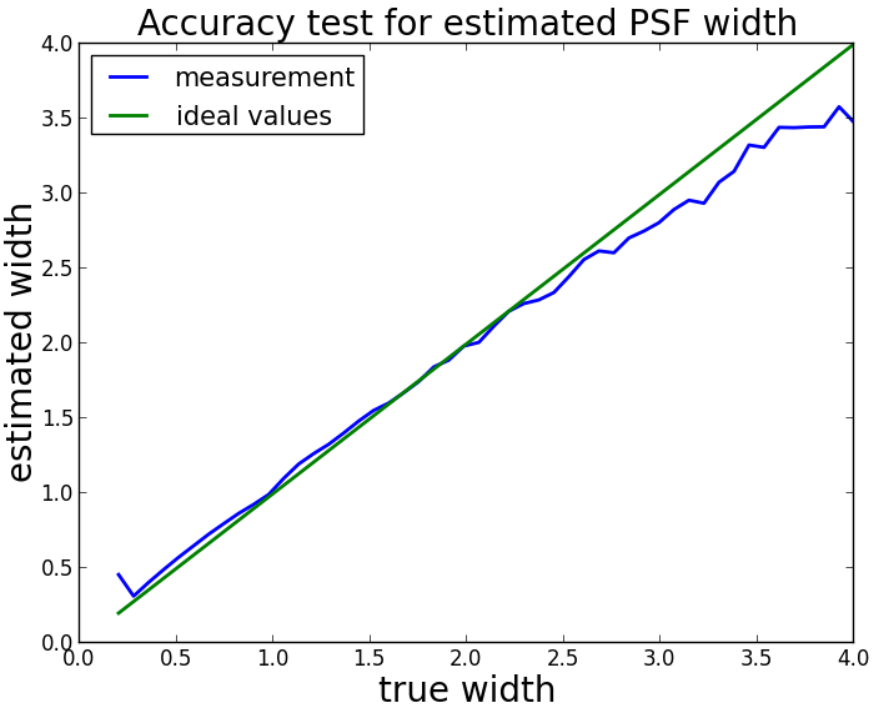


Figure 5.8: The figure shows the estimated widths of the point spread function plotted over the true widht of the simulated signal. The signal-to-noise ratio was 10 for this test. The green line shows the correct results.

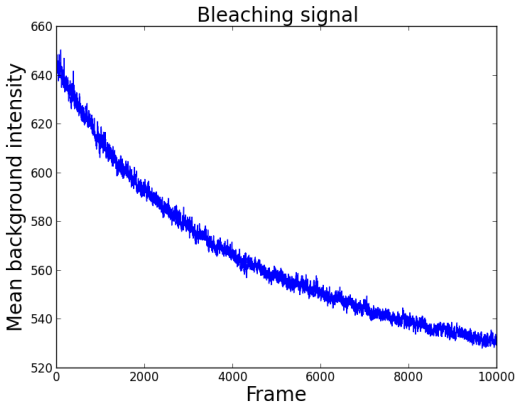


Figure 5.9: The number of active fluorophores decrease over time and so does the mean intensity of the image. This picture shows the mean intensities over 10 background pixel over time.

6 Multicolor registration

In microscopy it is often desirable to label different structures in a cell with different colors. To do so our collaborators use different fluoroscent molecules that emit light at different and distinguishable wavelengths. Using different filters it is possible to capture pictures just containing light emited from one fluorophore. To get a mulit-channel picture the different channels must be aligned. Because different flourophores emit different wavelengths, cromatic aberration apears. This means that the light for the same spot but with different wavelenghts is not mapped to the same spot in the image. To align the different channels despite cromatic aberration, beads are used. Beads are flourophores added to the probe, that emit light in all wavelengths the different markers do and therefore are visible in all channels. The beads can be used as landmarks, because their position in the original image is at the same spot. The task is to find a transformation that maps corresponding beads on each other.

6.1 Colorcomposer GUI

The goal for the colorcomposer tool is to provide software that is easy to use, flexible and powerful. The current version of the colorcomposer is easy to use, because the different channels can be aligned by just selecting the auto align option.
But it is also flexible. If the user wants to select the beads on his or her own, the beads can manually selected and deleted. After the transformation the user might use the implemented tools for colocalisation detection or save the transformed images and process them with the tool of his or her choice.
Also the colorcomposer is powerful as it provides for example information about the number of points currently under the cursor or its intensity. Also the estimated transformation error and the localisation error are computed and stored.
The basic framework for the colorcomposer was set up by Schleicher (2011). It contained the workflow for importing and exporting images the handling of bead objects and a linear transformation that used the beads in the order they were found. This early version was not usable and were improved.
Figure 6.1 shows improved colorcomposer GUI with two datasets loaded. The buttons on the right give the user the option to add or remove beads also in addition to the autodetected beads. There are also different sliders to control the values used for bead detection. In the lower right corner there is additional information provided about the total number of points within a rectengular with the selected cursor radius' size, also the sum of the intensities and the total number of frames for each data set is given. This information are helpful to determin whether or not

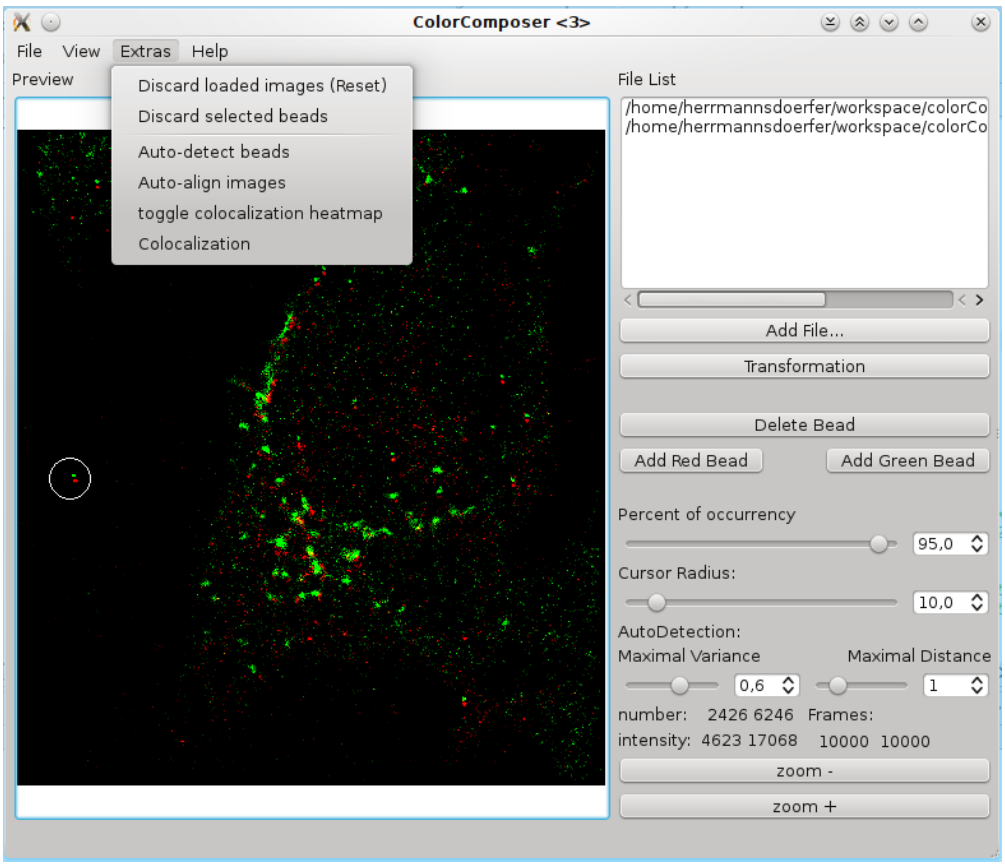


Figure 6.1: Improved colorcomposer GUI. On the right sliders to set the parameters are present. There are also buttons to add or delete singel beads. On the lower right additional information about the area under the coursor is displayed, such as number of pixels, total intensity and the number of frames in total.

a cluster of points is a bead or not.
On top there is a menu bar with new options, like discarding all beads, automatically detecting beads, calculate colocalisation measures and show or hide the colocalisation heatmap.

6.2 Features of the colorcomposer application

6.2.1 Invariance of input datas units

The resulting coordinate files from SimpleSTORM or other STORM algorithms may be given in units of pixels relative to the unprocessed data or in nanometers. Treating coordinates given in nanometer as pixlel units would lead to very huge and very sparse images to display in the colorcomposer. Therefore the colorcomposer reads out additional information from the coordinates text files header. This informations are the pixel to nanometer ratio and the used factor. With this information the

picture can be reconstructed as an upsampled version of the input image by the used factor regardless of the units used to save the coordinates file. If none of this information is given a pixel to nanometer ratio of 1 is assumed which garantues backward compatibility with older coordinate files given in pixel units.

6.2.2 Manual bead selection and removal

With the improved version of the colorcomposer it is possible to add beads manually. Therefore the desired location is clicked in the preview image and after that either the button "add green bead" or "add red bead" is hit. If there are enough frames containing localisations near the given location a bead is added to the center of mass of the intensities in that area. If the button "delete bead" is pressed all beads in the selected area are deleted.
This feature can be used to add beads which the automatic bead detection missed.

6.2.3 Automatic bead detection

The input for the colorcomposer application is a text file created by the storm algorithm that contains information about the position, intensity, symmetry, framenumber and signal-to-noise ratio of each detection. The beads should ideally appear in most of the images, this means they can be found by searching for detections that appear in almost every frame at the same position.
There was already an automatic bead detection implemented by Joachim Schleicher. This was improved in the following ways.
All important parameters for the bead detection can now be set in the GUI. The bead detection works by searching for points that appear in most of the frames. Instead of taking all localisations from the first frame as expected bead positions without considering locations that does not appear in the first frame, now a good subset of positions from the first 50 frames by skipping redundant positions based on the minimal distance of a new position to all positions already in the set. The range of 50 frames to look for beads is sufficiant because it is very unlikely that all 50 detections of the bead have been missed.
After good candidates are found their number of points, variance and mean position is determined like described by Schleicher (2011).
In the end beads that are too close together are merged to for a new bead with its center right between the merged beads.

6.2.4 Alignment of two multicolor images

6.2.5 Information about localisation certainty

The detected spots from SimpleSTORM contain some localisation error. This error will be derived in section refdetectionError. It depends on the signal-to-noise ratio and the scale of the point spread function of a single fluorophore.

6.2.6 Heatmap

6.3 Align Beads

After the beads for each channel are found the next task is to find the corresponding beads in each channel. It can happen that some beads occure in just one channel, if this is the case there will be no corresponding bead in the other channels.
To align the beads, the minimal number of beads, three, that are neccesary to calculate the transformation are chosen randomly from the first channel. After that, based on a probabilistic approach and a distance matrix containing information about the distances between all beads of the two channels, three beads from the second channel are chosen. It is more likely for nearer beads of the other channel to be selected, but any bead within a certain range can be chosen.
Using this pairs of beads a linear transformation is found like described by Schleicher (2011).
This transformation is used to tested how many other beads, that were not used to calculate the transformation match in total. It is assumed that the correct transformation will match other bead pairs in addition. This is very important because with every set of three points a valid transformation can be found that perfectly aligns this three beads. After that the whole procedure is done multiple times. In the end the best transformation is chosen based on the total number of bead pairs that match. If there are multiple transformations that match the same number of bead pairs, it is searched for the transformation with the lowest root mean square error for the matching beads.
In principle shearing should also be allowed for a linear transformation, but tests indicate that shearing does not occure, so it is disabled to improve stability. If there are just three beads in each channel, then every time a perfect transformation is found, but with the constraint of forbidden shearing, the right solution can be identified. There is an other problem with this transformation if the bead density is very high it might be that a transformation with much shearing is found that compresses the beads of one channel to a slim band. The probability to find a matching point by chance then is much greater then. Figure 6.2 shows the result of an incorrect transformation of simulated data. The red channel was created by randomly placing beads. The green channel was slightly shifted and rotated. The green channel was transformed to match with the red channel. Just a subset of beads were used to calculate the transformation and so this solution was found and chosen from the algorithm because of the additional matching point.
This effects can be suppressed by not allowing shearing for the transformation.

6.4 Total localisation error

There are three contributions to the total localisation error considered in this thesis. First there is the error introduced from the linear transformation to align the beads.

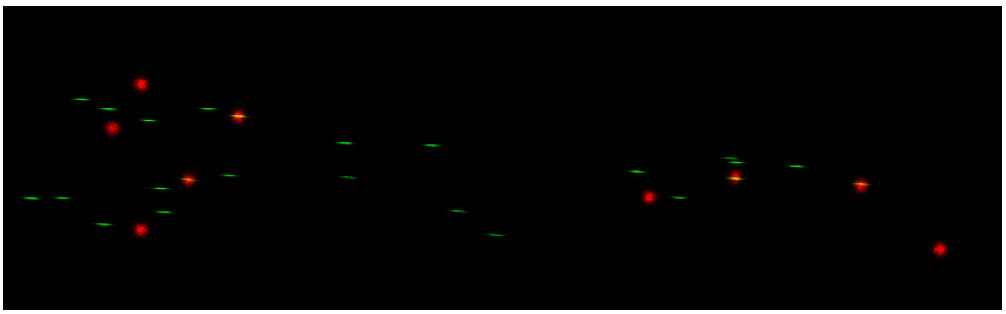


Figure 6.2: Affine transformation with shearing enabled. The green channel is deformed and one bead matches by chance which led to selection of this transformation.

To estimate this error the variance of the estimator is used. Since the transformation matrix is calculated by linear regression from the matricies B and R of the localisations of the first and the second layer of beads, with B beeing the matrix of the first layer to that the second layer R is transformed to. The variance of the estimator is

$$\text{variance registration} = \sigma^2 x_0 \left(R^T R\right)^{-1} x_0^T, \text{ with } x_0 = (x_o, x_1, 1) \tag{6.1}$$

Using equation 6.1 the variance for all pixel can be calculated.
The second contribution to the total localisation error is the localisation error of the SimpleSTORM algorithm. Its formula is

$$\text{variance localisation} = \frac{N^2\pi}{2S_0^2}\left(1+\frac{\sigma_{\text{PSF}}^2}{\sigma_{\text{filter}}^2}\right)^2\left(\sigma_{\text{filter}}^2+\sigma_{\text{PSF}}^2\right)^2 \tag{6.2}$$

It's derived in 5.5 and can be calculated for each detection individually based on the known signal-to-noise, which gives directly the signals intensity S because of the known noise variance of one which gives the nois' standard deviation N of also one. The PSFs width was either given or estimated and is passed to the colorcomposer in the detection coordinate file's header along with the prefactor f for the actually used filter. Using this the localisation error can be writen as

$$\text{variance localisation} = \frac{N^2\pi}{2S_0^2}\left(1+\frac{\sigma_{\text{PSF}}^2}{f^2\sigma_{\text{PSF}}^2}\right)^2\left(f^2\sigma_{\text{PSF}}^2+\sigma_{\text{PSF}}^2\right)^2 \tag{6.3}$$

$$= \frac{N^2\pi}{2S_0^2}\left(1+\frac{1}{f^2}\right)^2\left(1+f^2\right)^2\sigma_{\text{PSF}}^4 \tag{6.4}$$

$$= \frac{N^2\pi}{2S_0^2}\left(2+\frac{1}{f^2}+f^2\right)^2\sigma_{\text{PSF}}^4 \tag{6.5}$$

The third and minor contribution to the total localisation error for each pixel is the quantization noise. It occures when continous intensities, as the photons forming the PSF, are transformed into integer values. Quantisation noise describes the round-off

error. The round-off error can be any value between -0.5 and 0.5 and is uniformly distributed. Its variance σ_Q^2 is

$$\sigma_Q^2 = \int_{-\frac{1}{2}}^{\frac{1}{2}} x^2 dx = \frac{1}{12} \tag{6.6}$$

The unit of this error is the pixel size.
To get the total localisation error all three variances are summed up for each pixel.

6.5 Colocalisation

Colocalisation in wide field microscopy is a measure of the overlap of data point from different channels. It can provide information whether or not two molecules interact. With increasing resolution of the images colocalisation becomes more and more a measure of similar structures near each other. Two structures can't be at the very same position in the cell. The colorcomposer software provides both global and local colocalisation measurements.

6.5.1 Global colocalisation

The most common colocalisation measure is Pearsons correlation coefficient Rodgers and Nicewander (Feb., 1988). It is given as:

$$\text{Pearson correlation coefficient} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{6.7}$$

It is the ratio between the covariance between the points of two channels and their standard deviation.

Also the Manders correlation coefficients M_1 and M_2 and the overlap coefficient (E. M. M. Manders (1993)) are calculated.

$$M_1 = \frac{\sum_i R_{i,\text{coloc}}}{\sum_i R_i} \qquad M_2 = \frac{\sum_i G_{i,\text{coloc}}}{\sum_i G_i} \tag{6.8}$$

With $R_{i,\text{coloc}} = R_i$ if $G_i > 0$ and $R_{i,\text{coloc}} = 0$ otherwise and $G_{i,\text{coloc}} = G_i$ if $R_i > 0$ and $G_{i,\text{coloc}} = 0$. G_i, R_i are the intensities of the pixel of the green and red channel.

$$\text{overlap coefficient} = \frac{\sum_i R_i \cdot G_i}{\sqrt{\sum_i (R_i)^2 \cdot \sum_i (G_i)^2}} \tag{6.9}$$

6.5.2 Local colocalisation

Global colocalisation has the drawback that there is just one value for the whole image. If there are regions in the image that show much colocalisation and other regions without colocalisation the same colocalisation coefficient might be achieved as if one channel is distributed randomly.
For local colocalisation analysis the algorithm from Malkusch et al. (2011) are used and were further developed to gain a speed boost. The algorithm runs now approximatly 40 times faster. This was achieved by using the scipys Jones et al. (2001–) ckdtree function, which is a k-d tree implemented in C.

7 Related work

There are many groups all over the world that developed their own software to work on STORM images. This chapter shows the related work to give a better understanding what other concepts are used.
The DAOSTORM algorithmn Seamus J Holden (2011) adepts algorithms from software that is used to investigate crowded stellar fields. For spot localisation a fit of multiple PSFs is used. This is done to small clusters of molecules and not globaly. A model PSF is automatically generated from single PSFs in the data. DAOSTORM runs on the CPU.
FPGA Estimator Grüll et al. (2011) is an algorithm developed from a group at the Kirchoff-Institut für Physik in Heidelberg. It provides background subtraction by smoothing the pixels intensity over time, assuming Poisson distributed intensities. For high density data a methode is used that sets all further pixel to zero once a local minimum is detected. A gaussian estimator is used to determine the parameters of the Gaussian. The estimated scale can be compared with the given scale.
A maximum likelihood estimate of the PSFs parameters is used by GPUgaussMLE Carlas S Smith and Lidke (2010). For background subtraction either a constant threshold or a dark imaged aquired from all frames is used. Two filters of different size are applied to the data to determin signal candidates. A patch around the candidates is then used for maximum likelihood estimation of the PSFs parameters. Based on the estimated parameters filters are applied that compare this values with given input values for the PSF width. Also the uncertainty of the estimation was used to filter the candidates. This algorithm runs on the GPU. And can also be applied to 3d data.
An other algorithm is called M2LE Starr et al. (2012) which uses an user defined threshold for candidate selection and the median of a roi for background suppression, an user specified ellipticity threshold which can be made intensity dependend to discard candidates with an too high ellipticity. The PSFs position is determined using a maximum likelihood estimater, separated Gaussians are used for speed up.
The ImageJ plugin PeakFit Alex D. Herbert (2013) uses a 2d Gaussian non-linear least squares Levenberg-Marquardt fitting, the candidates are aquired subtracting two filters of different size. The PSFs width can either be specified or estimated from the data. The fitted spots are filtered based on their signal-to-noise ratio. For high density data multiple Gaussians are fitted to the data. As an ImageJ plugin it can deal with any kind of data processable via ImageJ.
PYME Baddeley et al. (2011) is an python package with functionality that can be used for localisation microscopy and other forms of widefield microscopy. Besides the STORM data analysis it also provides reconstruction and postprocessint algo-

rithms. It is applicable to 2d and 3d data. For noise treatment a Gaussian-Poisson noise mixture model is used.
QuickPALM Wolter et al. (2011) uses the methods from rapidSTORM WOLTER and SAUER (2012) which was earlier published. RapidSTORM uses a Spalttiefpass filter and the Högborn "CLEAN" method for background suppression and a to select the local maxima, a Gaussian function with scales either given or estimated from the data is fitted. An amplitude threshold is used to distinguish between signal and background pixels.

8 ISBI Challenge 2013

8.1 Introduction

The goal of the ISBI Challenge, as announced on their website (Biomedical Imaging Group (2013)), is to give an overview and understanding of availible algorithms for single particle localization microscopy. The focus was on 2d localisation, to give information about the depth of a localisated spot was optional. To benchmark results one needs groundtruth. Therefore the organisers created synthetic datasets of biologically relevant structures such as tubulins. To match realistic conditions the data was transformed to introduce different kinds of noise and background to it.
The participants were given training data sets and the corresponding groundtruth and one month before the deadline of the challenge the test sets. There were two different kind of datasets in principle. One with very dense spots and shorter sequences, the other with longer sequences and fewer spots per frame.
All paricipants were asked to submit their results and also the time it took to run the algorithm and the hardware configuration of the used system.

8.2 Terminology

To be able to compare different algorithms there must be a way to determine the correctly detected spots. To do so for each estimated positon of a flourophor, the nearest correct position of the molecule in the groundtruth data was searched within a lateral tolerance disc. Once a match was found this two spots were taken out of consideration for the matching.
One important parameter for this evaluation is the radius of the lateral tolerance disk, because it has big influence on the number of detections considered to be true positives (TP).
Detections with no associated spot in the groundtruth are called false positives (FP), spots in the groundtruth with no matching detection are called false negatives (FN).
This matching is done frame by frame, it is not possible to match a point from different frames even if the x and y coordinates match perfectly but the frame differs.
The precision (p) of a classification task is defined as the ration between the number of true positives and the sum of true positives and false positives:

$$\text{precision: } p = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8.1}$$

It is a number between 0 at worst and 1 at best, telling how reliable the result is, how likely it is that a labeld sample really belongs to the predicted class. In this context it means how certain a detected spot has its origin in a fluorophore attached to the investigated structure and it's origin is not wrongly detected background noise.
An other important value is the recall r that is defined as the ratio of true positives and the sum of true positives and false negatives:

$$\text{recall: } r = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8.2}$$

The recall lies also in a range from 0 to 1 and gives an impression on how many relevant spots were found.

8.3 Measures

For the evaluation three different measures were used. The f-score index f, the Jaccard index J and the rot-mean square distance RSME.

$$\text{f-score: } f \quad = \frac{2 \cdot p \cdot r}{p + r} \tag{8.3}$$

8.3.1 Jaccard index

Let A be the set of points of the groundtruth and B be the set of detected points. The Jaccard index J is defined as:

$$\text{Jaccard: } J = \frac{|A \cap B|}{|A \cup B|} \tag{8.4}$$

The intersection is done frame by frame. This means two spots from the groundtruth and the detection set just match if they occure in the same frame.

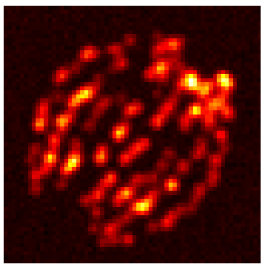
8.3.2 RSME

The root-mean square distance gives an impression how big the squared distance between a spot in the groundtruth and an associated detection was in average. It can be calculated like this:

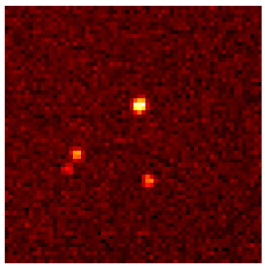
$$\text{RSME} = \frac{1}{|A \cap B|} \sum_{i=1}^{|A \cap B|} \left(p_a(x, y) - p_b(x, y)\right)^2 \tag{8.5}$$

8.4 Trainingsdata

All pictures shown in this section are reconstructed from original images or original images as provided by Biomedical Imaging Group (2013).



(a) High spot density



(b) Low spot density

Figure 8.1: One frame from bundled tubes training data set

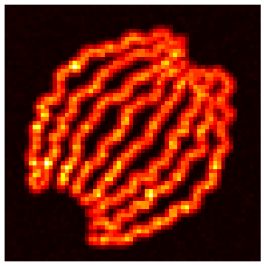


Figure 8.2: Maximum projection of bundled tubes data set

8.4.1 Bundled tubes datasets

There was two kinds of bundled tubes data sets, both created from the same underlying structure, one set with a high spot density and a short sequence of 360 frames, the other with fewer spots per frame but 12000 frames in total. Picture 8.1 shows one frame of each data set.

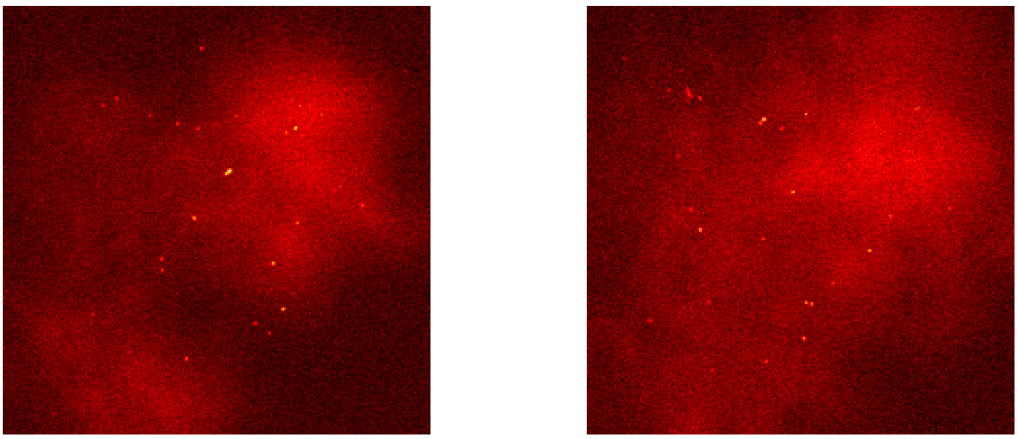
The original images very small, just 64 pixels in each dimension. Both sequences had spatial and temporal constant background. Picture 8.2 shows the maximum projection of the bundled tubes data set. The maximum projection is used to reduce the dimensionality of a data set. In this case for each pixel in x- and y-dimension, in the three dimensional dataset, the brightes value from all frames is taken.

8.4.2 Tubulin data sets

The other training data sets models 7 microtubules, a structure that is a long filament up several micrometers long and a diameter of about 25 nanometers. The spot density lies somewhere between the high density and the low density of the bundled tubes data sets. This data sets show strong inhomogeneity in spatial dimensions and moderate inhomogeneity in temporal dimension, see figure 8.3. This is the reason why in the lower left corner of the maximum projection 8.4 a brighter area can be seen.

8.5 Submissions

The task for the STORM algorithms was to perform best. In an ideal world this might be clear, it means, find all spots without any false detections and with perfect accuracy. But in the non perfect world this tasks leaves some room for interpretation. Is the best result the one that has the most true positives regardless of the number of false positives and the accuracy of detection. Or is it more favorable to find



(a) Tubulin2 frame 10

(b) Tubulin2 frame 1010

Figure 8.3: This pictures show the variability of the background in the spatial and temporal dimensions

exclusivly correct spots, but less?
For that reason we run each dataset with three different settings described below. All parameters were set manually.

8.5.1 High precision

A biological question to be answered by STORM might be how are the labeled proteins, distributed over the cell? There might be just a couple of flourophors attached to each protein. In that case a high precision is crucial. Otherwise for example homogenously distributed false positives all over the image might leed to the conclusion that the protein of intest is distributed all over the cell, but clustered in some spots.
The high precision submission set the α value to low values to get almost 100 percent precision.

8.5.2 High score

The parameters for the high score submission were set to maximize the Jaccard index. This was done by increasing the α value and estimating the number of true positives and false positives that were found in addition compared to the previously used α value. Based on tests on the trainings date the highest Jaccard index is achieved if the number of new false positives is equal to the number of new true positives.

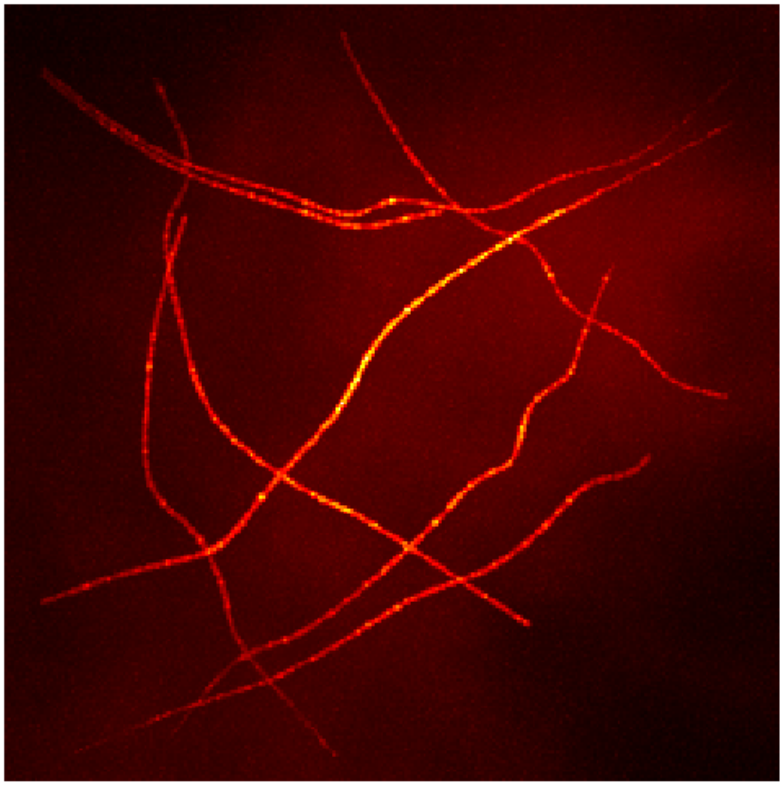


Figure 8.4: Maximum projection of tubulin data set

(a) Image before postprocessing (b) Mask (c) Result of postprocessing

Figure 8.5: Overview over the steps of postprocessing.

8.5.3 Highest score via postprocessing

The higher the α value the more likely it is to detect false positives. But this false positives are distributed all over the image. For the postprocessing the assumption that true structur in the image shows a higher density of spots than background with false positives. A mask was created by smoothing the recunstructed image and then thresholding it. This results in a mask that covers image regions with high density. This mask was used to discard all detections that are not covered and therefore are assumed to be false positives in the background.
For the submission a high α was chosen to get also darker true positives. Afterwards the postprocessing was applied. The postprocessing worked well on the contest data because it contained just dense structures. For biological applications the postprocessing has to be used carefully because it discards small, isolated structures, which might be not noise but just widly spread proteins.
Figure 8.5 shows the data before postprocessing, the mask and the result.

8.6 Results

All tables shown in this section and all diagrams are created from the results released by Biomedical Imaging Group (2013).

8.6.1 High density data

Table 10.1, 10.2 and 10.3 show the results of the SimpleSTORM algorithm for the different submissions. The submission with postprocessing applied gave the best results for the Jaccard index, compared to the other two submissions. As expected the settings for high score, but without postprocessing yielded better Jaccard indices than the high precission settings, but also worse precision. As expected the high precision settings resulted in the highest possible precision of one hundred percent.

Also the postprocessed data has a perfect precision, this demonstrates that the postprocessing removed the false positives, which decreased the precision in the high score (without postprocessing) results. The accuracy was the the same for all three submissions.
The time the software needed was submitted with the results, this means that it depends strongly on the system the algorithms run on. Over all SimpleSTORM took about 90 seconds to process the high density data sets. This was an average runtime. Most of the faster algorithms used the GPU.

Figures 10.1 and 10.2 show the mean of the Jaccard index and the RSME score taken over all high density data sets. The results for SimpleSTORM are highlighted. SimpleSTORM achieved an average rank for the Jaccard index and a top five rank for the RSME score. With 100 percent precision it was among 6 other participants on the first rank.

8.6.2 Low density data

The tables 10.4, 10.5 and 10.6 show the same trends as for the high density data, but with this data sets, the difference between the precisions is greater between the high precision submission and the other two submissions. The runtime was about five minutes which was, like for the high density data sets, average.

As for the high density data sets the mean scores were calculated for the low density data set and are shown in figure 10.3 and 10.2. The results for the precision are also shown in figure 10.5. SimpleSTORM got the third rank in the Jaccard index and intermediate ranks for accuracy and precision.

8.6.3 Overall

Of the 29 participants of the challenge just 15 submitted results for the high density data sets. There is no winner since no software performed well on each score. This can be explained by a trade-off between high accuracy (a low RSME) and the number of spots detected (influence on Jaccard index). For example, bright spots are easier to localize. Just finding the brightest spots yields a good performance on the RSME score but less points. This trade-off can be seen not just in our results, where a high Jaccard index goes along with an intermediate RSME score and vice versa, but also for other algorithms like Fast-ML-HD, which got the highest Jaccard index on the high density data but the second worst RSME score for the same data set.
To compare the different algorithms some metric has to be used that takes the shown trade-off between Jaccard index and accuracy into account. One way is to use the rank each software achieved on the different data sets for different scores and average over it. The results averaging just the mean rank for the Jaccard index and for the RSME score are shown in figure 10.6 for the low density data sets and in figure 10.7 for the high density data sets.
SimpleSTORM achieves the best mean rank for high density data and the third best result for the low density data sets.

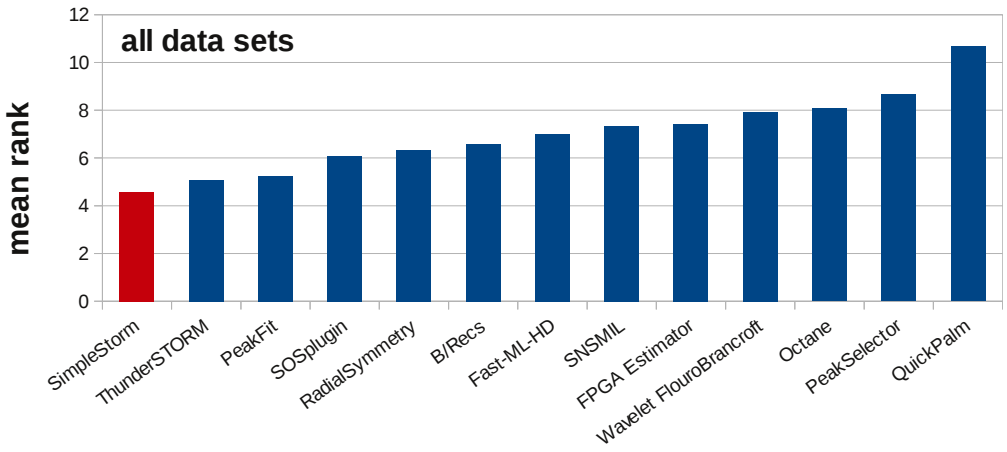


Figure 8.6: Averaged rank over Jaccard index and RSME score for all data sets. Lower ranks are better

Figure 8.6 shows the result if the average rank for each data set and for RSME score and the Jaccard index is calculated just based on the ranks of all algorithms that processed both, high density and low density data. All other algorithms were taken out of this consideration. Although this methode of comparison did not weight the ranks which means that extraordinary good performance makes no difference to slightly better performance between two consecutive softwares, but it shows that SimpleSTORM provides a good compromise between many correct detections and high accuracy.

9 Conclusion

10 Appendix

10.1 ISBI challenge

Table 10.1: Results for the main submission (with postprocessing)

Dataset	Jaccard (in %)	Precsision (in %)	Recall (in %)	RMSE (in nm)
HD1	40.44	100	41	40.13
HD2	30.84	100	31	63.18
HD3	12.55	100	13	61.8

Table 10.2: Results for the high precission submission

Dataset	Jaccard (in %)	Precsision (in %)	Recall (in %)	RMSE (in nm)
HD1	37.62	100	38	40.23
HD2	28.37	100	28	60.70
HD3	12.66	100	13	62.58

Table 10.3: Results for the high score submission (without postprocessing)

Dataset	Jaccard (in %)	Precsision (in %)	Recall (in %)	RMSE (in nm)
HD1	37.96	100	38	40.22
HD2	29.73	94	30	63.33
HD3	13.00	99	13	63.95

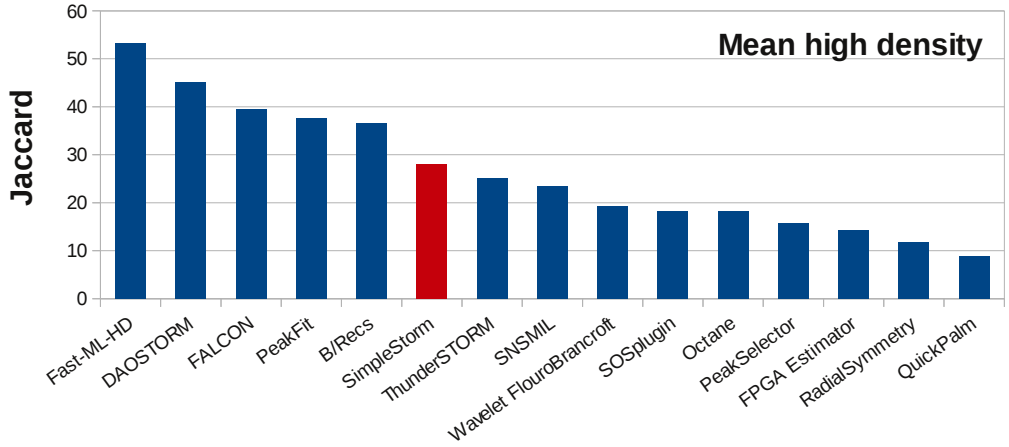


Figure 10.1: Results for high density data sets. The Jaccard indices are averaged over all three data sets. For this score higher is better.

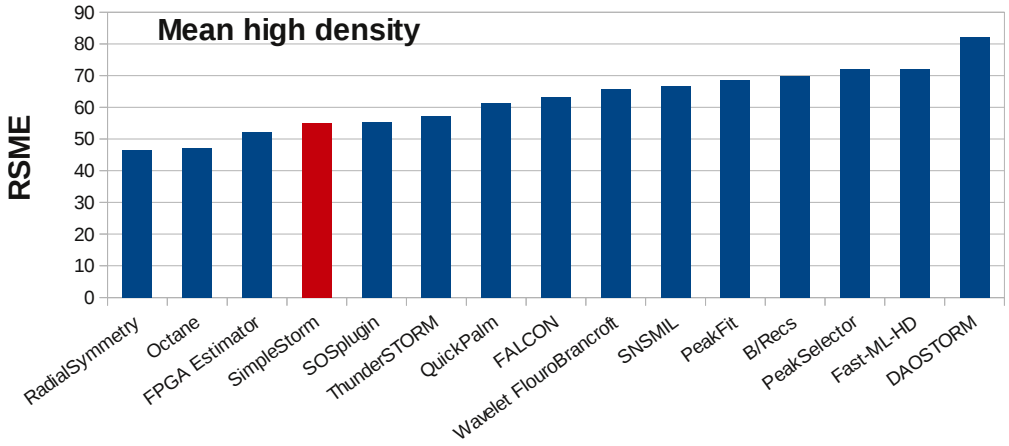


Figure 10.2: Results for high density data sets. Averaged RSME scores over all three datasets. For this score lower is better.

Table 10.4: Results for the main submission (with postprocessing)

Dataset	Jaccard (in %)	Precsision (in %)	Recall (in %)	RMSE (in nm)
LS1	86.12	100	86	26.32
LS2	69.64	97	71	37.43
LS3	47.48	99	48	54.2

Table 10.5: Results for the high precission submission

Dataset	Jaccard (in %)	Precsision (in %)	Recall (in %)	RMSE (in nm)
LS1	83.39	100	83	27.91
LS2	63.21	100	63	39.57
LS3	40.82	100	41	49.55

Table 10.6: Results for the high score submission (without postprocessing)

Dataset	Jaccard (in %)	Precsision (in %)	Recall (in %)	RMSE (in nm)
LS1	87.23	99	88	28.10
LS2	65.95	89	72	44.60
LS3	40.82	96	48	54.40

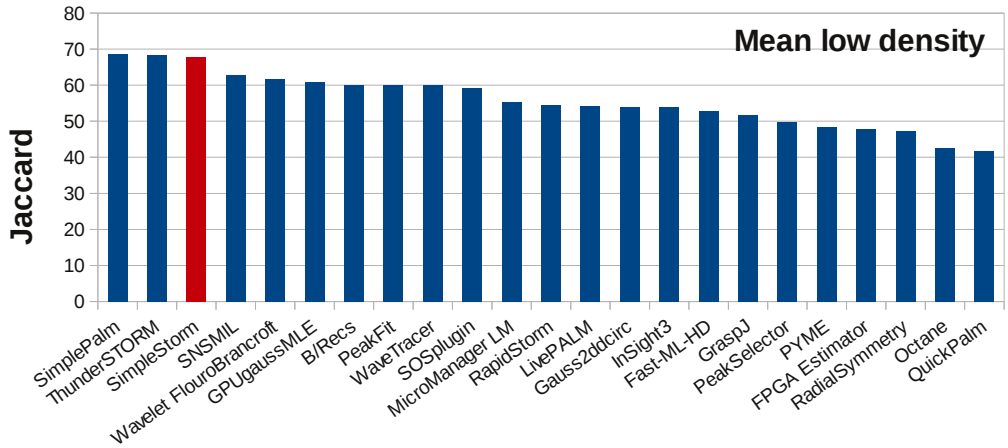


Figure 10.3: Results for low density data sets. The Jaccard indices are averaged over all three data sets. For this score higher is better.

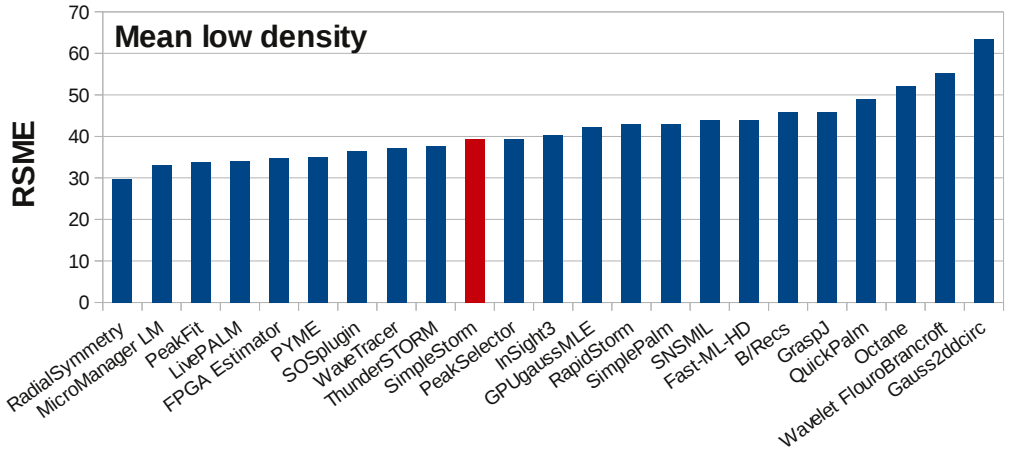


Figure 10.4: Results for low density data sets. Averaged RSME scores over all three datasets. For this score lower is better.

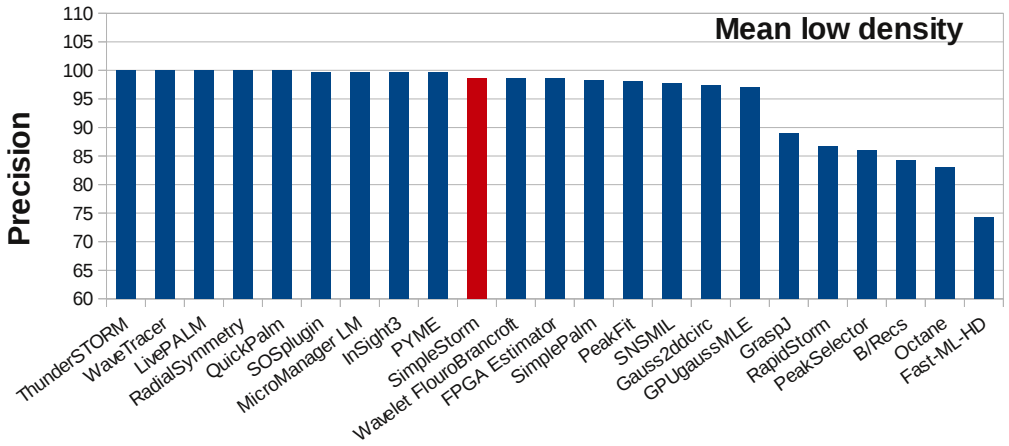


Figure 10.5: Results for low density data sets. Averaged Precision over all three datasets. For this score higher is better. The y axis is broken to show the differences better.

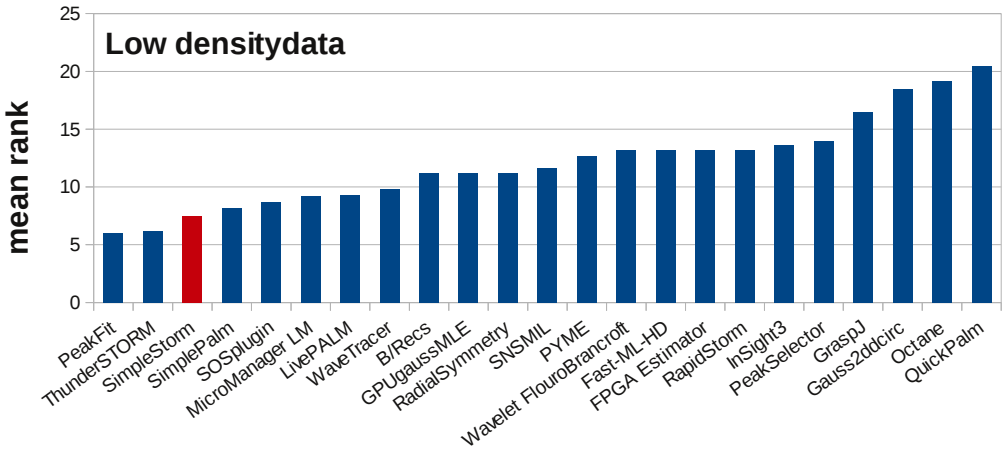


Figure 10.6: Averaged rank over Jaccard index and RSME score for all low density data sets. Lower ranks are better.

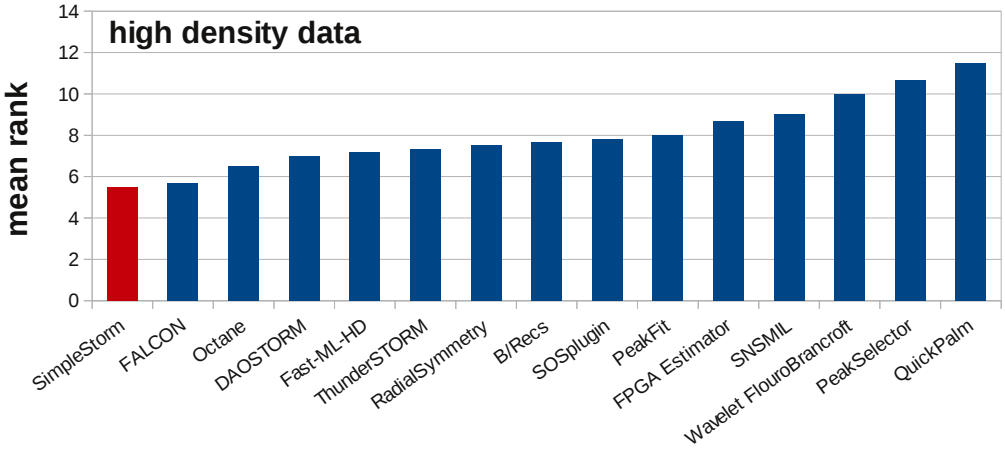


Figure 10.7: Averaged rank over Jaccard index and RSME score for all high density data sets. Lower ranks are better

11 Bibliography

Abbe, Ernst (1873), "Beiträge zur theorie des mikroskops und der mikroskopischen wahrnehmung." *Archiv für Mikroskopische Anatomie*, 9, 456.

Alex D. Herbert, Anthony M. Carr, Thomas Etheridge (2013), "Peakfit - single-molecule localisation software for imagej." URL `http://www.sussex.ac.uk/gdsc/intranet/microscopy/imagej/smlm_plugins`. Accessed on 17.05.2013.

Anscombe, F. J. (1948), "The Transformation of Poisson, Binomial and Negative-Binomial Data." *Biometrika*, 35, 246–254, URL `http://www.jstor.org/stable/2332343`.

Baddeley, David, MarkB. Cannell, and Christian Soeller (2011), "Three-dimensional sub-100 nm super-resolution imaging of biological samples using a phase ramp in the objective pupil." *Nano Research*, 4, 589–598, URL `http://dx.doi.org/10.1007/s12274-011-0115-z`.

Betzig, E., G. H. Patterson, R. Sougrat, O. W. Lindwasser, S. Olenych, J. S. Bonifacino, M. W. Davidson, J. Lippincott-Schwartz, and H. F. Hess (2006), "Imaging intracellular fluorescent proteins at nanometer resolution." *Science*, 313, 1642–1645.

Biomedical Imaging Group, Switzerland, EPFL (2013), "Localization microscopy isbi 2013 challenge." URL `http://bigwww.epfl.ch/smlm/challenge/`. Accessed on 04.04.2013.

Carlas S Smith, Bernd Rieger, Nikolai Joseph and Keith A Lidke (2010), "Fast, single-molecule localization that achieves theoretically minimum uncertainty." URL `http://www.nature.com/nmeth/journal/v7/n5/full/nmeth.1449.html`.

E. M. M. Manders, J. A. Aten, F. J. Verbeek (1993), "Measurement of co-localization of objects in dual-colour confocal images journal of microscopy." *Journal of Microscopy*, 169, 375–382.

Grüll, Frederik, Manfred Kirchgessner, Rainer Kaufmann, Michael Hausmann, and Udo Kebschull (2011), "Accelerating image analysis for localization microscopy with fp-gas." *In Field Programmable Logic and Applications*. Doi: 10.1109/FPL.2011.11.

Heilemann, Mike, Sebastian van de Linde, Mark Schüttpelz, Robert Kasper, Britta Seefeldt, Anindita Mukherjee, Philip Tinnefeld, and Markus Sauer (2008), "Subdiffraction-resolution fluorescence imaging with conventional fluorescent probes." *Angewandte Chemie International Edition*, 47, 6172–6176, URL `http://dx.doi.org/10.1002/anie.200802376`.

Hwang, Youngbae, Kim Hun-Sik, and In So Kweon (2012), "Difference-based image noise modeling using skellam distribution." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34, 1329–1341.

Jones, Eric, Travis Oliphant, Pearu Peterson, et al. (2001–), "SciPy: Open source scientific tools for Python." URL `http://www.scipy.org/`.

Kats, Ilia (2013), *Implementation and benchmarking of simpleSTORM, a self-calibrating single-molecule localization algorithm*. Laboratory report, University of Heidelberg.

Köthe, Ullrich (2007), "Reliable low-level image analysis." URL `http://hci.iwr.uni-heidelberg/Staff/ukoethe/papers/habil-koethe.pdf`. Habilitation Tesis.

Köthe, Ullrich (2011), "The vigra computer vision library." URL `http://hci.iwr.uni-heidelberg.de/vigra`. Accessed on 31.04.2013.

Malkusch, Sebastian, Ulrike Endesfelder, J. Mondry, M. Gellé ri, P.J. Verveer, and Mike Heilemann (2011), "Coordinate-based colocalization analysis of single-molecule localization microscopy data." *Histochem Cell Biology*.

Newberry, Michael (1998), "Pixel response effects on ccd camera gain calibration." URL $http://www.mirametrics.com/tech_note_cdgain.htm$. Tech note.

R Core Team (2012), *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, URL `http://www.R-project.org/`. ISBN 3-900051-07-0.

Rodgers, Joseph Lee and W. Alan Nicewander (Feb., 1988), "Thirteen ways to look at the correlation coefficient." *The American Statistician*, 42, pp. 59–66, URL `http://www.jstor.org/stable/2685263`.

Rousseeuw, Peter J. and Katrien van Driessen (1999), "A Fast Algorithm for the Minimum Covariance Determinant Estimator." *Technometrics*, 41, 212–223, URL `http://www.jstor.org/stable/1270566`.

Rust, X. Zhuang, M.; M. Bates (2006), "Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (storm)." *Nature Methods*, 3, 793–796.

Schleicher, Joachim (2011), *Image Processing for Super-Resolution Localization Microscopy Utilizing an FPGA Accelerator*. Master's thesis, Heidelberg University.

Seamus J Holden, Achillefs N Kapanidis, Stephan Uphoff (2011), "Daostorm: an algorithm for high- density super-resolution microscopy." URL http://www.nature.com/nmeth/journal/v8/n4/full/nmeth0411-279.html.

Starr, Rebecca, Shane Stahlheber, and Alex Small (2012), "Fast maximum likelihood algorithm for localization of fluorescent molecules: erratum." *Opt. Lett.*, 37, 1967–1967, URL http://ol.osa.org/abstract.cfm?URI=ol-37-11-1967.

WOLTER, S. and M. SAUER (2012), "Follow-up to paper by s. wolter, m. schÃ¼ttpelz, m. tscherepanow, s. van de linde, m. heilemann and m. sauer, entitled real-time computation of subdiffraction-resolution fluorescence images." *Journal of Microscopy*, 245, 109–109, URL http://dx.doi.org/10.1111/j.1365-2818.2011.03562.x.

Wolter, Steve, Ulrike Endesfelder, Sebastian van de Linde, Mike Heilemann, and Markus Sauer (2011), "Measuring localization performance of super-resolution algorithms on very active samples." *Opt. Express*, 19, 7020–7033, URL http://www.opticsexpress.org/abstract.cfm?URI=oe-19-8-7020.

Erklärung:

Ich versichere, dass ich diese Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den 10. Juni 2013 .....