

# Using DNA-PAINT and the Picasso Software to image Carbon Nanotube Structures

Marius Wiesner  
University of Osnabrück

October 16, 2024

# Contents

<b>1 Document Structure and Objectives</b>	<b>3</b>
<b>2 Theoretical Basics of Super Resolution Microscopy and the PAINT Method</b>	<b>4</b>
2.1 Light Diffraction and the Diffraction Limit in Microscopy . . . . .	4
2.2 The Point Spread Function (PSF) . . . . .	7
2.3 Super Resolution Mircoscopy and the PAINT Method . . . . .	9
2.4 Techniques for Localizing Single Molecules . . . . .	10
2.5 The Treatment of Noise in Single Particle Loaclization . . . . .	14
<b>3 DNA-PAINT</b>	<b>18</b>
<b>4 The Picasso Software Package</b>	<b>22</b>
4.1 Purpose and Structure of the Picasso Software . . . . .	22
4.2 Subprogram: Localize . . . . .	23
4.3 Subprogram: Filter . . . . .	27
4.4 Subprogram: Render . . . . .	28
4.5 The Picasso Library for Data Analysis . . . . .	29
<b>5 Application: Imaging Single Walled Carbon Nanotubes with DNA-PAINT and Analyzing Functionalisation Properties</b>	<b>31</b>

# 1 Document Structure and Objectives

This document has been created with two main objectives in mind.

Firstly, it is supposed to serve as an introduction to the basics of super resolution (SR) microscopy which enables imaging of structures beyond the diffraction limit. Specifically, it concentrates on the PAINT(= Point Accumulation for Imaging in Nanoscale Topography) method and a specific implementation of it, the DNA-PAINT method, which can be used to image nanoscale structures. The goal of this is to image single walled carbon nanotubes that are grafted with DNA strands.

In section 2 the document therefore starts by reviewing the basics of SR microscopy such as the diffraction limit, the point spread function and laying down the basic principle of the PAINT method. The chapter also goes into the treatment of noisy signals that are inevitably collected during the recording of image data.

It then focuses on the DNA-PAINT method, which uses DNA strands to implement the PAINT principle.

Secondly, this document shows how one can make use of the software environment package *Picasso* to generate images from camera image data obtained via the DNA-PAINT method and perform data analysis on that data.

Hence, after a short presentation on the structure of the Picasso software package in subsection 4.1, a more detailed look on its crucial components is given in chapters 4.2 to 4.4. These sections show how a high resolution image can be created from camera data all within the picasso environment.

Subsection 4.5 briefly describes the code library that accompanies the Picasso software package.

The last section of this document (section 5) shows how it is possible to perform data analysis on camera data acquired by the DNA-PAINT method using the python programming language (although in principle other programming languages can be used). This entails preprocessing raw camera image data to a usable format which can then be imported into a desired python environment using the Picasso python library.

For the data analysis part of section 5 this document is accompanied by two Jupyter Notebook files<sup>1</sup> which contain the necessary code for exporting/importing data and performing data analysis. This part mainly wants to show how information from camera image data can be extracted to gain information of the imaged structure. The data analysis part uses image data of single walled carbon nanotubes that were imaged with DNA-PAINT.

---

<sup>1</sup>Wiesner 2024a; Wiesner 2024b

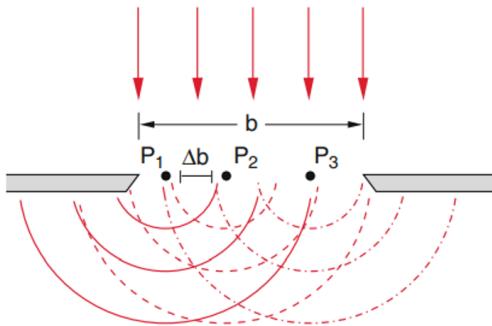
## 2 Theoretical Basics of Super Resolution Microscopy and the PAINT Method

### 2.1 Light Diffraction and the Diffraction Limit in Microscopy

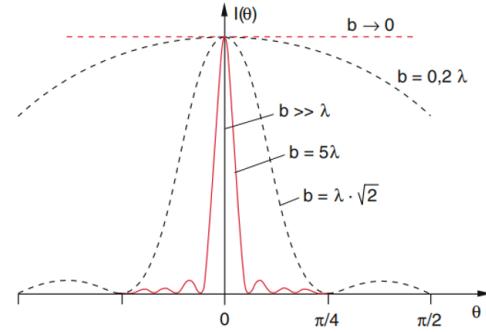
Classical microscopy techniques, that illuminate the whole structure at once, are physically limited in their resolution by the so called diffraction limit.

The term diffraction describes the phenomenon of interference patterns due to the wave like nature of light that emerge when light passes through narrow apertures. Classically this is explained with the Huygen-Fresnel principle and the principle of superposition.

The Huygen-Fresnel principle is illustrated in Figure 1a. It states that every point of a wavefront<sup>2</sup> is to be considered as a source of a secondary spherical wave.



(a) Huygen-Fresnel principle for a small slit with width  $b$  that is hit by a wavefront.



(b) Interference pattern that is generated by the superposition of the secondary waves in (a). The function shows intensity  $I$  against angle  $\theta$  perpendicular to the slit's axis.

Figure 1: Illustration of an interference pattern (b) that is generated by the secondary waves (a) according to the Huygen-Fresnel principle.

Source: (Demtröder 2006, (a) p. 322, (b) p.323).

The principle of superposition simply states that the Amplitudes (and thereby the intensities  $I \propto E_i$ ) of electromagnetic waves that occupy the same point in space must be added together according to

$$E(\vec{r}, t) = \sum_m A_m(\vec{r}, t) e^{i\phi_m}, \quad (1)$$

---

<sup>2</sup>In the following i will only consider diffraction in two dimensions, since the DNA-PAINT technique also images in two spacial coordinates, i.e. in a plane.

resulting in either constructive or destructive interference, depending on the individual waves' amplitude  $A_m$  and their respective phase  $\phi_m$  at time  $t$  at the position  $\vec{r}$ .<sup>3</sup> The result is an interference pattern as in Figure 1b.

Both principles together result in the phenomenon of light diffraction as illustrated in Figure 2.

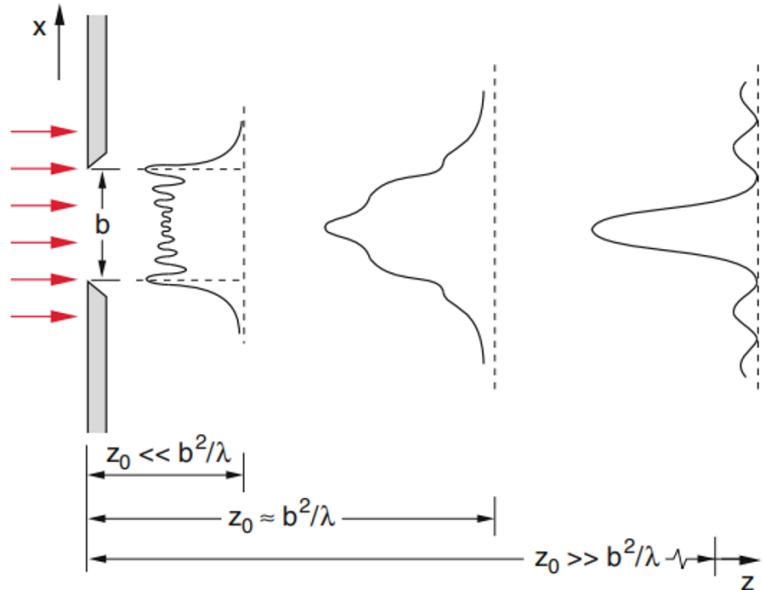


Figure 2: Diffraction patterns (Functional graphs of intensity  $I(\theta)$  as in Figure 1b) created by a slit opening as in Figure 1a. Pattern shape varies according to image distance  $z_0$ .

Source: (Demtröder 2006, p. 333). Edited by the author.

Since different light sources can be part of the same structure, as it is the case in structural microscopy, they can interfere with each other and deter the resolution quality due to an overlap of their individual diffraction patterns, as can be seen in Figure 3. The limit with which two point like sources of light can be resolved optically is given by the Abbe diffraction limit:

$$d = \frac{\lambda}{2 \cdot n \cdot \sin(\alpha)} = \frac{\lambda}{2 \cdot NA} \quad (2)$$

where  $d$  is the minimum distance needed for two points to be resolved independently. The refractive index of the medium is given by  $n$  and  $\theta$  is half the angle of the cone as it is shown in Figure 4. The wavelength of the light is given by  $\lambda$ .

---

<sup>3</sup>Demtröder 2006, p. 299.

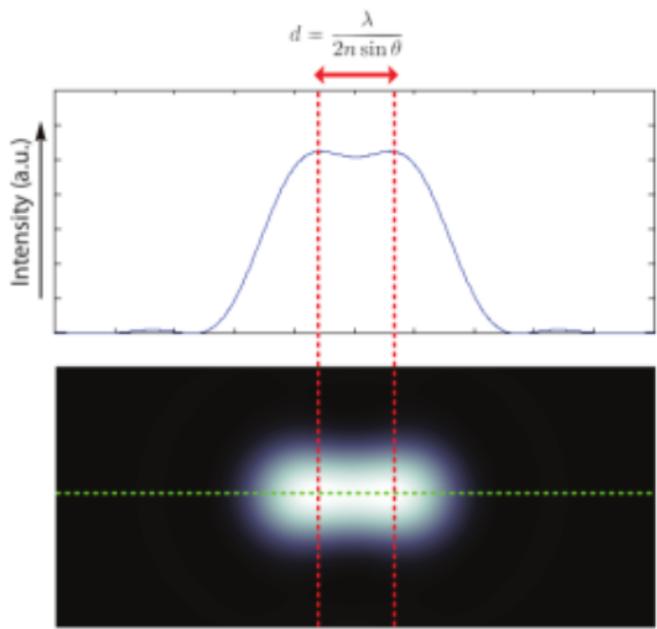


Figure 3: Overlap of two individual circular diffraction patterns generated by two point like light sources. The diffraction limit  $d$  gives the minimum distance to distinguish between the two light sources.

Source: (Scientific Volume Imaging 2024b).

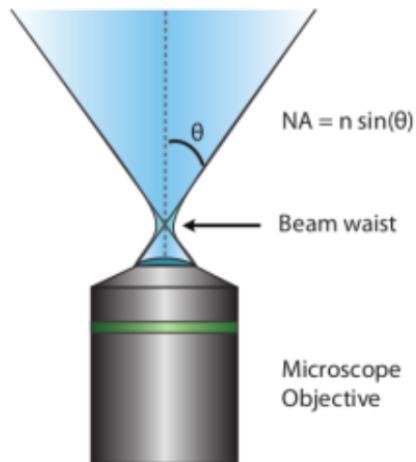


Figure 4: Illustration of the numerical aperture as a measure of how much light from the focus an objective lense can collect.

Source: (Scientific Volume Imaging 2024b).

The product of  $n \cdot \sin(\alpha)$  is also referred to as the numerical aperture (NA), which measures the amount of light coming from the lens's focus point that the objective can collect.<sup>4</sup>

## 2.2 The Point Spread Function (PSF)

The specific diffraction pattern depends on the distance from the gap  $z_0$  that the diffracted image is viewed from as well as the size of the gap  $b$  that a wavefront of light is passing through as can be seen in Figure 1b. Diffraction is most noticeable when the size of the gap is similar to the wavelength of the wave ( $b \approx \lambda$ ). However, if the gap is much larger relative to the wavelength of the light ( $b \gg \lambda$ ), almost all of the intensity falls into the region  $\theta = 0^\circ$ , i.e. the light seems to pass through the slit in a straight line. Conversely, if the gap is very small relative to the wavelength ( $b \ll \lambda$ ) the light spreads uniformly behind it.<sup>5</sup>

With this in mind, one can understand how different microscopes can produce different diffraction patterns of an identical light source. In the microscope the diffraction of light can occur at the specimen plane due to interaction of the light with small particles of the structure, at the margins of the objective front lens or at edges of circular apertures at the objectives.<sup>6</sup>

Because of this, a single point of light never really appears as a point in the microscope image, but rather as a diffraction pattern containing a central disk having a finite diameter and being encircled by a fading series of rings. This basic diffraction pattern can be seen in Figure 5 and is referred to as an airy disc.<sup>7</sup>

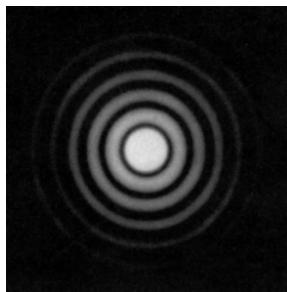


Figure 5: An airy disk diffraction pattern that is produced when a wavefront of coherent light passes through a small circular aperture.

Source: (Demtröder 2006, p. 324).

---

<sup>4</sup>Scientific Volume Imaging 2024b.

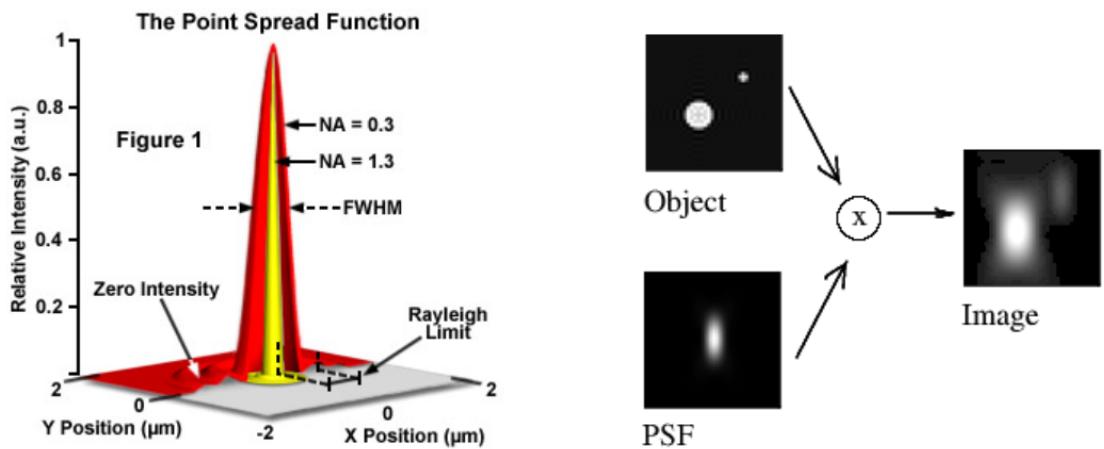
<sup>5</sup>Demtröder 2006, pp. 322f.

<sup>6</sup>National High Magnetic Field Laboratory 2015.

<sup>7</sup>Scientific Volume Imaging 2024a.

Each optical system therefore produces a different final, blurred image of a structure. This distortion is described by the system's Point Spread Function (PSF). The PSF describes what a single point like light source looks like in the final image of the optical system. It is illustrated in Figure 6a<sup>8</sup>. The distance from the PSF's maximum to the first minimum is referred to as the Rayleigh limit which in the case of two point like light sources is identical to the diffraction limit in Equation 2. Notice how the numerical Aperture (NA) influences the PSF.

According to the Heuygen-Fresnel principle (see 2.1) the light that is emitted by the point source spreads out spherically. After passing through the apertures of the optical system (i.e. the microscope) a diffraction image of the point source should emerge similar to that of the airy disk in Figure 5. However, due to effects like spherical abberation the diffraction pattern and therefore the PSF is often a distorted circle like that in Figure 6b.<sup>9</sup>



(a) Point Spread Function showing the intensity profile in the image plane of a point like light source.

(b) Linearity of the image formation process.

Figure 6: The Point Spread Function (PSF) (a) and its role in the image formation process (b).

Sources: (a)(Rottenfusser 2024), (b)(Scientific Volume Imaging 2024c)

It is important to note the importance of the PSF. Since light obeys the law of superposition according to Equation 1, every structure to be imaged can be thought of as a collection of point like light sources, each producing a distorted image according

---

<sup>8</sup>The PSF is of course three dimensional in theory but since the created image in florescence microscopy is a two dimensional one the corresponding PSF is a two dimensional cross section of the 3D PSF

<sup>9</sup>Rottenfusser 2024.

to the system's PSF. The resulting image is therefore the sum of the overlapping PSFs as illustrated in Figure 6b.<sup>10</sup>

### 2.3 Super Resolution Microscopy and the PAINT Method

Super resolution microscopy describes a collection of microscopy techniques that can image structures beyond the diffraction limit of conventional microscopes.<sup>11</sup> Although different approaches exist, we will focus on a technique labeled the "PAINT" method.

The acronym PAINT stands for "Point Accumulation for Imaging in Nanoscale Topography" and describes a microscopy super resolution technique for imaging nanoscale structures. It is localization based, meaning that instead of observing the whole structure that is supposed to be imaged at once, it images individual points of the structure over multiple camera frames. This is achieved with the use of fluorescent molecules by controlling them in such a way that only a subset of them<sup>12</sup> are detectable by the camera device at any given time, i.e. within a single camera frame.<sup>13</sup> The PAINT method is illustrated in Figure 7.

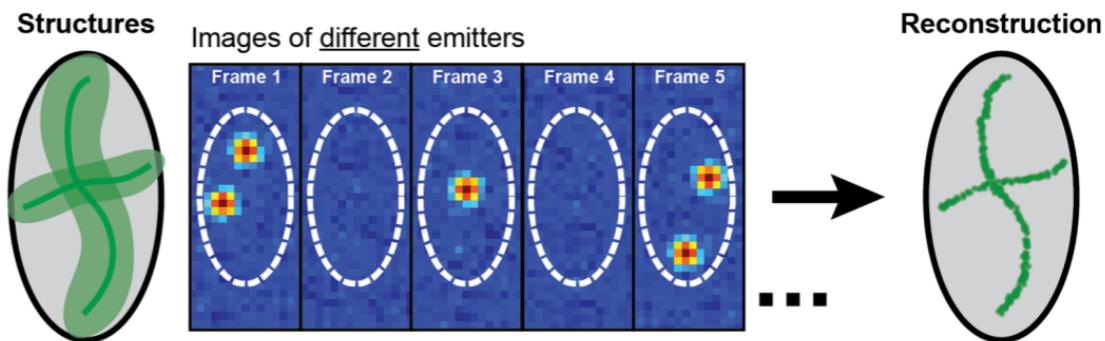


Figure 7: Illustration of how the PAINT method bypasses the diffraction limit (diffracted image of the structure indicated on the left in light green) by eliminating the point spread of individual spots.

Source: (Diezmann, Shechtman, and W. E. Moerner 2017, p. 2).

As described in subsection 2.1 the fundamental resolution limit of a microscope is given by the Abbe diffraction limit or the more abstractly defined Rayleigh limit as seen in Figure 6a, both quantified by Equation 2. Typical values for the numerical

<sup>10</sup>Scientific Volume Imaging 2024c.

<sup>11</sup>Diezmann, Shechtman, and W. E. Moerner 2017, pp. 3-4.

<sup>12</sup>In the following the diffracted image of an individual fluorescent molecule will simply be referred to as a "spot".

<sup>13</sup>Schnitzbauer et al. 2017b, pp. 1198-1199.

Aperture (NA) of modern high end microscope objectives range between  $1.3 - 1.5$  which, with a typical wavelength of  $500\text{nm}$  for visible light, yields a diffraction limit of  $d \approx 150 - 200\text{nm}$ .<sup>14</sup> However, since the PAINT method captures fluorescent molecule spots individually, the diffracted shape of an individual spot, i.e. its PSF, can be used to mathematically localize the molecule within the PSF. The PAINT method thereby effectively circumnavigates the diffraction limit by eliminating the point spread of a molecule individually and, only after that, superimposing the individual molecules to create an image of the whole structure.

## 2.4 Techniques for Localizing Single Molecules

Now that the basic idea of the PAINT method is clear, a basic description of the localization process of detected spots (see the exemplary frames in Figure 7) shall be given to broaden the readers understanding of the image formation process.

As already mentioned in the previous section, the PAINT method bypasses the diffraction limit by evaluating the PSF of individual spots and estimating the actual position of the fluorescent molecule within the PSF. The pixelated camera image of an individual spot, which represents the 2D cross section of the spots PSF, can be seen in Figure 8. The area containing the pixels of the PSF image is often referred to as the region of interest (ROI).<sup>15</sup>

The image that is captured by the camera is influenced by a multitude of different factors, e.g. the undistorted PSF of the optical system, the number of photons emitted by the fluorophores ( $N_{sig}$ ), the number of photons emitted by the background ( $N_{bg}$ ), detector properties, sample drift, etc.<sup>16,17</sup>

Mathematically this dependency can be described by an imaging function

$$H(\theta) = H(x, y, z, N_{sig}, N_{bg}, \dots), \quad (3)$$

which is dependent on all the parameters that influence the final image. These parameters are summarized in the imaging parameter  $\theta$ . The imaging function dictates the intensity on the image plane at each location  $(x_{image}, y_{image})$ , resulting in

$$I(x_{image}, y_{image}) = H(\theta), \quad (4)$$

where  $I$  is the intensity captured by the camera at the position  $(x_{image}, y_{image})$  of the image plane.

---

<sup>14</sup>Diezmann, Shechtman, and W. E. Moerner 2017, p. 4.

<sup>15</sup>Diezmann, Shechtman, and W. E. Moerner 2017.

<sup>16</sup>Diezmann, Shechtman, and W. E. Moerner 2017, p. 5.

<sup>17</sup>Deschout et al. 2014, pp. 260-264.

Since the actual image is not continuous but rather a discrete, pixelated image, as can be seen in Figure 8,  $I(x_{image}, y_{image})$  can be quantized into pixels  $p_k$  with each pixel containing a uniform intensity, so that

$$I(x_{image}, y_{image}) = I(p_k). \quad (5)$$

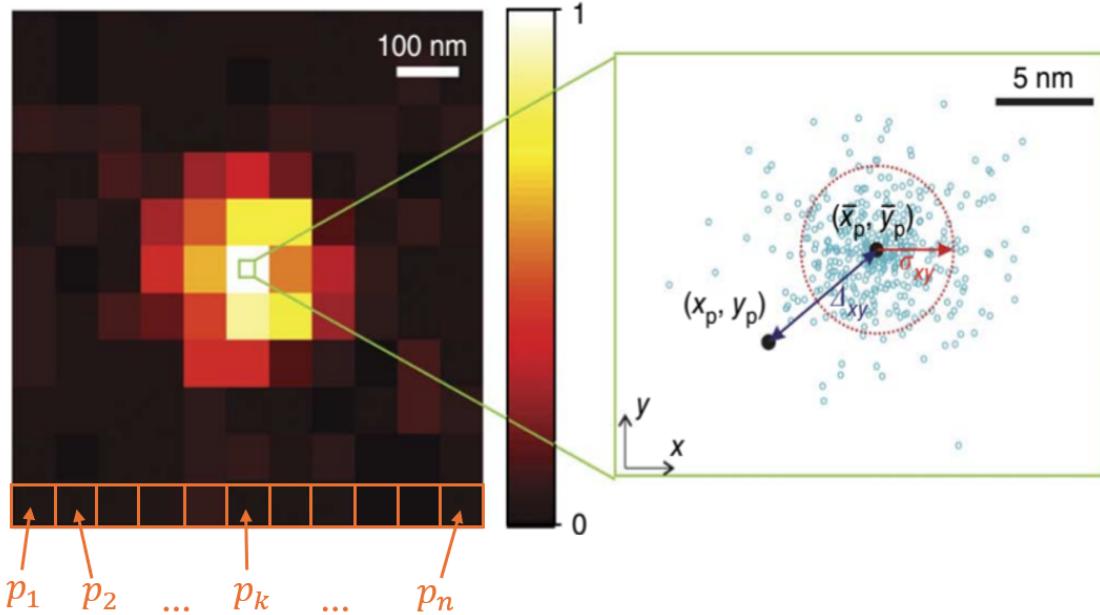


Figure 8: The pixelated PSF(or ROI) of an experimentally recorded image of a single fluorescent emitter at position  $(x_p, y_p)$ . Blue circles represent determined position estimates from different images of the same emitter. The localization precision is given by the standard deviation  $\sigma_{xy} = \sqrt{\sigma_x^2 + \sigma_y^2}$  and the localization accuracy is given by the estimates mean  $\Delta_{xy} = \sqrt{\Delta_x^2 + \Delta_y^2}$ .  
Source: (Deschout et al. 2014, p. 254).

When working with image data (i.e, frames that consist of individual blurred spots) a fitting algorithm is used on the intensity profile  $I(p_k)$  of a spot's ROI to determine an estimate  $(\hat{x}_0, \hat{y}_0)$  of the actual position  $(x_0, y_0)$  of the fluorescent molecule.

The most straight forward fitting algorithms use simple heuristics to come up with an estimate of the true position. Examples for this are the max-value estimator, which just determines the pixel with the highest intensity as the best estimate, or the weighted-average estimator, which picks the weighted average of all the pixels

intensities as the best estimate.<sup>18</sup> However, more sophisticated algorithms take an inverse approach. Instead of just using the image data to determine a best estimate, they try to optimize the parameters of a model function  $H_{model}(\theta)$  to fit the observed data  $I(p_k)$ . The most sophisticated model functions that can be used for  $H_{model}(\theta)$  are the Richards-Wolf model<sup>19</sup> and the Gibson-Lanni model<sup>20</sup>. But since these functions are very complex,  $H_{model}$  is often approximated with an Airy disc function that yields intensity profiles such as in Figure 5. Since the airy disc function

$$H_{model}^{\text{airyDisc}} = C * \left( \frac{J_1(kNA\rho)}{kNA\rho} \right)^2, \quad (6)$$

which shall not be explained further here<sup>21</sup>, is also complicated,  $H_{model}$  is often approximated with a gaussian distribution<sup>22,23</sup>

$$H_{model}^{\text{gaussian}} = \frac{N_{\text{sig}}}{2\pi\sigma^2} \cdot e^{-\frac{\rho^2}{2\sigma^2}} + b, \quad (7)$$

where  $N_{\text{signal}}$  is the number of photons from the molecule,  $\sigma$  is the standard deviation,  $\rho$  is the distance from the true position  $(x_0, y_0)$  and  $b$  is the number of photons coming from background and other noise sources. Figure 9 (B) shows a gaussian distribution for the x position as a blue curve.

With the choice of a modelling function  $H_{model}$  comes the question of how to optimize it to the image data  $I(p_k)$ . There are two common ways of optimization.

The first is a least-square (LS) optimization approach. This method simply minimizes the square error  $S$  between the models' intensity  $\mu_k(\theta)$  (i.e., the intensity at pixel  $k$  calculated by the gaussian distribution in Equation 7) and the observed intensities  $n_k$ <sup>24</sup>

$$S = \sum_{k=1}^N [n_k - \mu_k(\theta)]^2. \quad (8)$$

Alternatively one can use a maximum-likelihood estimator (MLE) to fit the gaussian distribution to the data. Maximum-likelihood estimation is rooted in the field of

---

<sup>18</sup>Diezmann, Shechtman, and W. E. Moerner 2017.

<sup>19</sup>Richards and Wolf 1959.

<sup>20</sup>Gibsen and Lanni 1992.

<sup>21</sup>For an explanation of the parameters see (Diezmann, Shechtman, and W. E. Moerner 2017, p. 7)

<sup>22</sup>Diezmann, Shechtman, and W. E. Moerner 2017, p. 7.

<sup>23</sup>Small and Stahlheber 2014, p. 268.

<sup>24</sup>Diezmann, Shechtman, and W. E. Moerner 2017, p. 6.

inferential statistics. The idea is to find the set of model function parameters  $\theta$  such that the generated model function has the highest probability (or *likelihood*) of generating the observed data. There are different approaches to this method, one being the log-likelihood function

$$L(\theta|n_k) = \sum_{k=1}^N [n_k \ln(\mu_k(\theta)) - \mu_k(\theta)] \quad (9)$$

as defined in (Diezmann, Shechtman, and W. E. Moerner 2017), which needs to be maximized to find the best estimate. A concise explanation of the likelihood principle is given in (Cousins 2020).

The MLE and LS often achieve similar results in accuracy and efficiency, although the MLE approach has shown to be superior to the LS approach.<sup>25</sup>

The choice of using the model function approach instead of using simple heuristics to come up with an estimator lies in the accuracy and efficiency of these estimators. Accuracy is equivalent to a small systematic error, meaning that measurements are not systematically pushed into a specific direction away from the true value. In the case of localization, accuracy therefore corresponds to small difference  $\Delta_{xy}$  between the actual position  $(x_p, y_p)$  and the mean of several estimated positions  $(\bar{x}_p, \bar{y}_p)$  of the same spot, as can be seen in Figure 8.

Efficency is defined as having a small random error, meaning that measurements are spread randomly but tightly around their mean. In the case of localization, this corresponds to a small deviation  $\sigma_{xy}$  from the mean estimated position  $(\bar{x}_p, \bar{y}_p)$ .<sup>26</sup>

Figure 9 compares the accuracy of a max-value estimator, a weighted-average estimator and a gaussian estimator optimized via the LS method. The simulated data was obtained by (Diezmann, Shechtman, and W. E. Moerner 2017).

---

<sup>25</sup>Diezmann, Shechtman, and W. E. Moerner 2017, section 2.

<sup>26</sup>Taylor 1997, pp. 94-97.

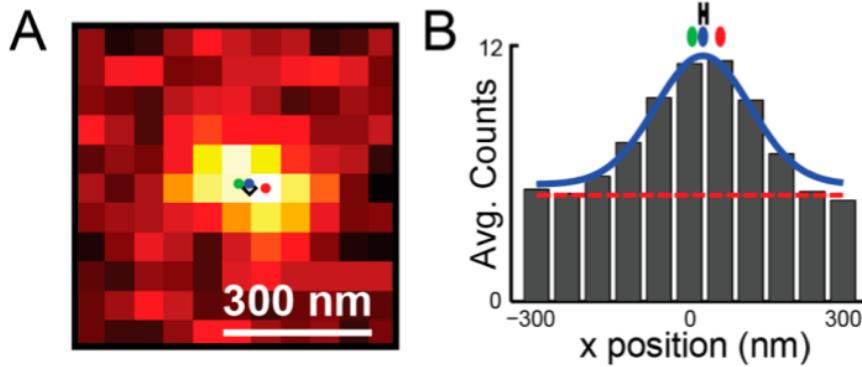


Figure 9: (A) Simulated image of a fluorescent molecule at the black diamond. simulated emitter produces airy disk signal with averaged 250 photons per pixel and is subject to an average of 5 signal photons per pixel. Green, blue, red dots: Average estimates after 10000 simulated frames of max-value, gaussian and weighted average estimators, respectively. (B) Image counts of A averaged along the x axis. Red dotted line shows the influence of background photons.

Source: (Diezmann, Shechtman, and W. E. Moerner 2017, p. 6).

The accuracy and efficiency of the three estimators after 10000 simulations can be seen in Table 1.

Method	Accuracy ( $\Delta_{xy}$ )	Efficiency ( $\sigma_{xy}$ )
Gaussian	+0.1nM	$\pm 10.0\text{nM}$
Max-value	-0.9nM	$\pm 34.0\text{nM}$
Weighted average	-20.0nM	$\pm 5.4\text{nM}$

Table 1: Values for the accuracy and efficiency of the estimators used on the simulated data of Figure 9.

Source: (Diezmann, Shechtman, and W. E. Moerner 2017, p. 6)

The gaussian estimator clearly yields the highest accuracy while also being the most efficient, which is why a gaussian estimator (with either an LS or MLE optimization) is commonly preferred to other estimators.

## 2.5 The Treatment of Noise in Single Particle Localization

The attentive reader might have read the preceding section and come up with the question: How can a simulated perfect emitter, like that in Figure 9, that is not subject to any noise, produce a signal that differs over a multitude of frames?

This phenomenon is called *photon noise*<sup>27</sup> and is due to the quantum nature of light or rather photons.<sup>28</sup> Since each individual photon arrives on the image plane in a probabilistic manner, the PSF only shows the probability distribution for an individual photon. Therefore, the image data  $I(p_k)$  can vary for different measurements of the same emitter. Since the position of each emitted photon can be viewed as a random variable that is independent of the other photons, the measured intensity of each pixel  $p_k$  follows a Poisson distribution as can be seen in Figure 10.  $N$  denotes the average rate of collected photons if the same emitter is measured multiple times. The standard deviation is given by  $\sigma = \sqrt{N}$ . This also means that a higher number of collected photons will decrease the relative standard deviation  $\sigma_r = \frac{\sigma}{N} = \frac{1}{\sqrt{N}}$ , thus reducing the effect of photon noise.

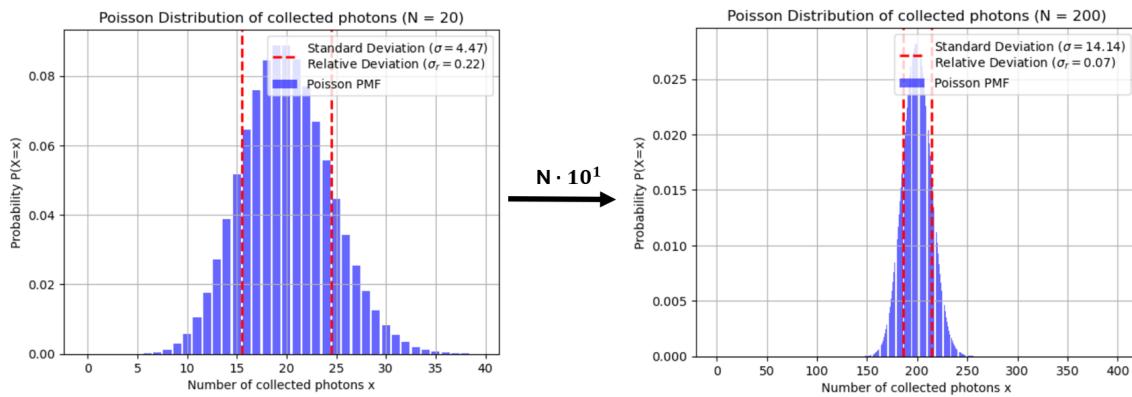


Figure 10: A Poisson distribution showing the probability of measuring a specific intensity for an individual pixel  $p_k$  when the average intensity is  $N$ . The right graph has an average that is one magnitude bigger than that of the left graph, resulting in a lower relative deviation.

This effect can also be seen in Figure 11, where a greater photon collection (achieved by a stronger irradiance of the probe or an increased exposure time (= *integration time*)) achieves better signal resolution, even when the relative magnitude of other noise sources stays the same. While maximizing the photon count therefore seems plausible, it has to be taken into account that the possible emission from a fluorophore might saturate and higher irradiance then only amplifies background noise, i.e. photons that do not originate from the molecule of interest.<sup>29,30</sup>

<sup>27</sup>Some papers on the topic also refer to photon noise as *shot noise*.

<sup>28</sup>Dobrucki and Kubitscheck 2017, p. 118.

<sup>29</sup>Diezmann, Shechtman, and W. E. Moerner 2017, p. 6.

<sup>30</sup>W. Moerner and Fromm 2003, Section B.

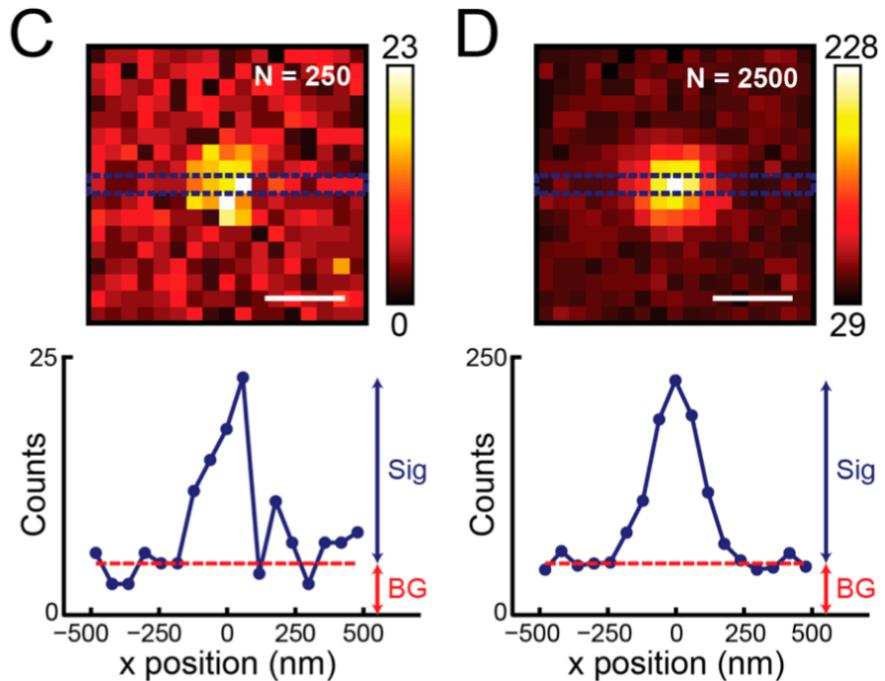


Figure 11: Simulated image of a perfect emitter that shows how the effects of photon noise are reduced when increasing the irradiance, such that the average collected photons rise from 250 (C) to 2500 (D). Graphs below show the intensities of the blue dotted area with relative Signal-to-Background ratio (SBR) staying constant.  
Source: (Diezmann, Shechtman, and W. E. Moerner 2017, p. 5)

There are two major sources of noise that originate in the camera detector and influence the PSF, namely *dark noise* and *read noise*. Since these result from currents of the camera measuring device, they shall not be discussed further here. For now, the reader should just be aware that the camera measuring device itself causes noise that can hinder the detection of spots.<sup>31</sup>

Although an increased exposure time can reduce the effects of noise<sup>32</sup>, with increased exposure time one generally has to take into account possible photobleaching effects of the fluorophores.<sup>33</sup> A more detailed explanation of these noise sources is given in Chapter 3 of (Dobrucki and Kubitscheck 2017).

Background noise photons also arise from the sample itself but are hard to suppress as discussed in (W. Moerner and Fromm 2003).

<sup>31</sup>Dobrucki and Kubitscheck 2017, pp. 114-118.

<sup>32</sup>Dobrucki and Kubitscheck 2017, pp. 114-118.

<sup>33</sup>Demchenko 2020.

- In summary it can be said that all noise sources can be put into three categories:
- (1) Photon noise (or shot noise) which is due to the quantum nature of light and can be reduced by increasing the collected photons of a fluorophore (either by increased irradiance or longer exposure time)
  - (2) Noise that originates from the camera hardware
  - (3) Noise that originates from light sources other than the observed specimen, i.e. background noise

The overall influence of noise sources can also be viewed quantitatively. For this purpose, (Basche, Ambrose, and W. Moerner 2020) developed a formula that takes into account the discussed noise sources which yields an estimate for the ratio between signal intensity  $I_s$  and the root mean square (RMS) of the overall noise  $I_{\text{noise}}^{\text{rms}}$ , called the Signal-to-noise ratio (SNR)

$$SNR = \frac{I_s}{I_{\text{noise}}^{\text{rms}}}. \quad (10)$$

A detailed explanation on how to use their formula to optimize the SNR is given in (W. Moerner and Fromm 2003).

### 3 DNA-PAINT

The previous sections described the basic principles and requirements of super resolution microscopy, particularly the PAINT method. They can be summarized as follows:<sup>34</sup>

The optical system must have a sufficient sensitivity (i.e, a sufficient signal-to-noise ratio (SNR)) to be able to detect single molecule spots.

The specimen must contain a concentration of fluorophores that is low enough such that the blurred images of single spots do not overlap in a given camera frame.

For the final image construction an algorithm is needed (preferably a gaussian estimator optimized via the LS or MLE method) to determine the position of single spots below the diffraction limit.

DNA-PAINT implements these principles by binding fluorescent molecules to single stranded DNA-strands called "imager strands". These imager strands can diffuse freely within an imaging solution around the structure that is supposed to be imaged. The imaging structure itself is functionalized with DNA strands referred to as "docking strands", that are grafted to the structure. These docking strands are complementary to the imager strands, meaning that they can transiently bind with each other.<sup>35</sup> See (G. Manoharan et al. 2023) and (Gririraj Manoharan 2021) for further information on the process of functionalizing Carbon Nano Tubes with different molecules. If you want to know how to capture images with the DNA-PAINT method yourself in an experiment, a detailed explanation is given in (Schnitzbauer et al. 2017b).

Figure 12 illustrates this basic idea of DNA-PAINT. The main principle for imaging a structure (in the following: a Carbon Nano Tube) is that for an imager strand to be detected by the camera device it needs to be attached to a docking strand.

Freely diffusing imager strands diffuse over many pixels within a single frame, which is why the camera cannot collect enough photons from a fixed spot to differentiate the signal to the noise. Temporarily bound imager strands are fixed at the same place for the binding duration, allowing the camera to collect enough photons to detect a blinking spot<sup>36</sup>.

---

<sup>34</sup>Diezmann, Shechtman, and W. E. Moerner 2017, p. 3.

<sup>35</sup>Schnitzbauer et al. 2017b.

<sup>36</sup>I will often refer to the detected spots as a "localization" since such a detected blinking spot represents the localization of a docking strand attached to the structure. The Picasso software refers to these spots as localizations as well.

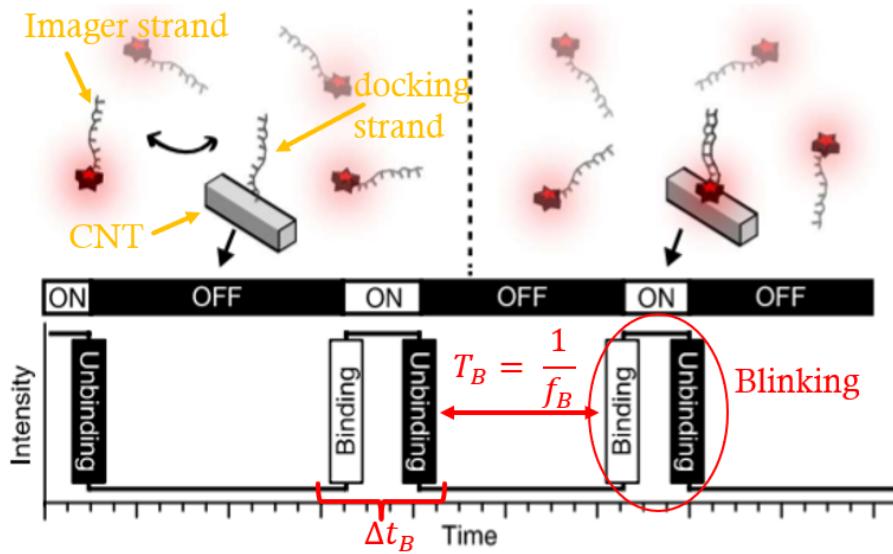


Figure 12: Transient binding of complementary DNA strands creating detectable blinking spots (= localizations).

Source: (Schnitzbauer et al. 2017b, p. 1199). Edited by the author.

Notice that  $\Delta t_B$  refers to the binding duration of a DNA strand pair. Accordingly,  $\frac{1}{f_B}$  represents the refractive period  $T_B$  with  $f_B$  being the binding frequency, i.e. the number of bindings that occur at an individual docking strand per unit of time.

The binding duration  $\Delta t_B$  depends on the stability of the DNA duplex and can be influenced by things like strand length, temperature, etc. The binding frequency  $f_B$  depends on the influx rate of imager strands to a docking strand and can be controlled by adjusting the concentration of imager strands.<sup>37</sup>

One of the big advantages of this technique is the avoidance of unwanted photo-bleaching effects.<sup>38</sup> This is due to the huge supply of imager strands for an individual docking strand. If one imager strand's conjugated fluorescent dye is affected by photobleaching it has practically no effect for the overall spot detection since there are many more imager strands the docking strand is coupling with over the duration of the experiment.

Figure 13 shows an example of a collected frame generated by DNA-PAINT image data.

<sup>37</sup>Schnitzbauer et al. 2017b.

<sup>38</sup>Photobleaching is a series of different photochemical reactions in which light-induced damage or modification of the fluorophore occurs, weakening the fluorescent capabilities of the fluorophores.

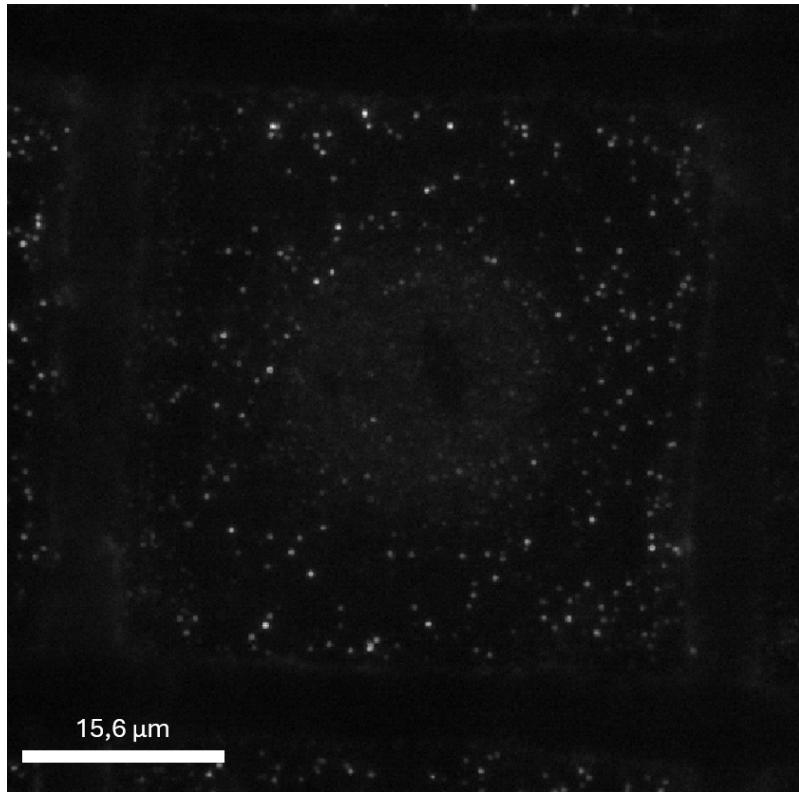


Figure 13: Camera frame showing blinking spots of Carbon Nano Tubes functionalized via the DNA-PAINT method. The whole sample grid consists of  $480 \times 480$  pixels with a pixelsize of 130nm.

After localizing the spots in the individual frames (usually one DNA-PAINT measurement consists of  $10^4 - 10^5$  frames), the spots undergo a fitting algorithm to determine the position estimates and the fitted spots are superimposed to create the final image (as already discussed in subsection 2.3). An exemplary, final DNA-PAINT image can be seen in Figure 14. It shows a rendered DNA-PAINT image of a bundle of amino acid chains with a width of around 100 nM each. It can be clearly seen how this method can create a clear structural image beyond the diffraction limit of  $\approx 200$ nM when the blinking spots of all collected camera frames are superimposed. The figure also contains some spots that are isolated from the actual structure (i.e., the amino acid chains). This is due to imager strands binding to the coverslip. However the effect of such unwanted binding can usually be considered negligible, as shown for example in (Gririraj Manoharan 2021, p. 107).

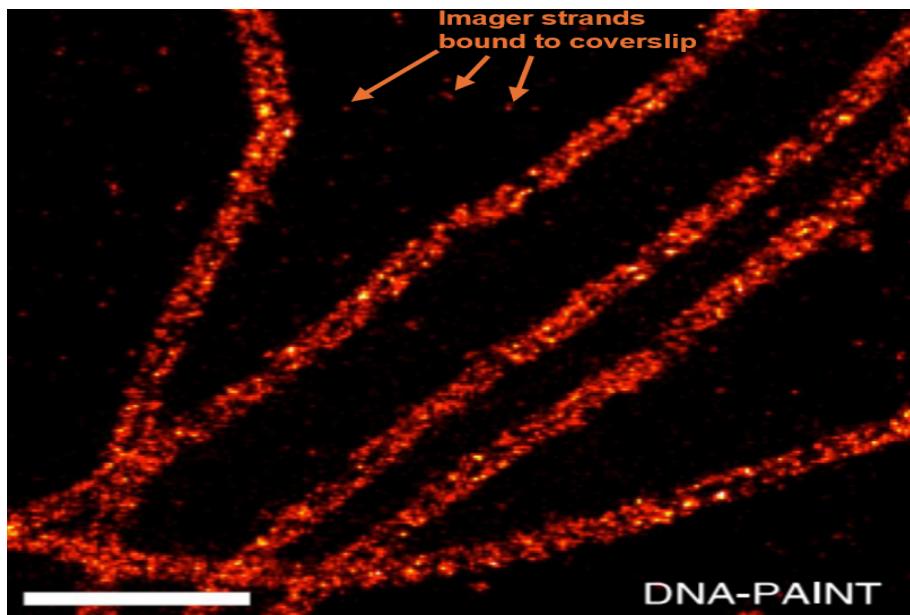


Figure 14: DNA-PAINT image of the amino acid chain  $\alpha$ -tubulin (Scale bar: 500nm).

Source: (Schnitzbauer et al. 2017b, p. 1199). Edited by the author.

It is also relevant to note that fluorescent quenching has been shown in (Eftrik 2002) to be distance dependent, i.e. dependent on the length of the DNA strands. The term 'Quenching' refers to phenomena that weaken the fluorescent intensity of fluorophores without destroying them, i.e. without bleaching.(Eftrik 2002) In this context this is due to energy transfer of fluorophores to the Nanotubes. See (Eftrik 2002) for more information on this. However, experiments that try to quantify this dependence empirically , like the one done by (Gririraj Manoharan 2021, p. 109 f.), seemingly show that strand lengths commonly used in DNA-PAINT do not effect the intensity of the fluorophores noticeably. More experiments are needed to conclude whether strand length significantly effects a DNA-PAINT image.

## 4 The Picasso Software Package

### 4.1 Purpose and Structure of the Picasso Software

The *Picasso* software environment was created specifically to work with raw image data that was obtained by the DNA-PAINT method. It is structured into multiple subprograms and files to generate structural images from raw camera image data. That way, all image creating efforts (except data acquisition, i.e. performing the DNA-PAINT method) can be handled by the Picasso software environment. It also provides interfaces such as the *Filter-program* for viewing image statistics and the *picassosr* library for more sofisticated data analysis.

The Picasso environment can be downloaded from

<https://github.com/jungmannlab/picasso>.

Follow the instructions in the read-me file (readme.rst) to install the Picasso environment on your machine. There you will find two ways of installing Picasso.

The easiest way is to use the installer executable which installs the stand-alone program of Picasso (this contains all the subprograms) with desktop shortcuts for each subprogram.

Alternatively you can install Picasso as a python package which allows to use Picasso's internal routines in custom Python programs. The easiest way to achieve this is by having an anaconda distribution on your machine and then follow the steps of the read-me file to create an anaconda environment that supports the picasso python library called *picassosr*<sup>39</sup>. If you use this way of installation there will unfortunately be no desktop shortcuts for all the subprograms. You can still run them from the command line interface of your machine. Figure 15 shows how to do that on a windows operating system.

Since this document also deals with data analysis aspects of the imaged data (which means programming is necessary) in section 5, it is advised to go with the second installation route.

As already mentioned the Picasso environment consists of multiple subprograms, each with an individual purpose. An overview of all the subprograms is provided in Figure 16.

---

<sup>39</sup>Note: You should create a new anaconda environment for installing the *picassosr* package. To see why see subsection 4.5 or (Wiesner 2024b).

```

Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\mariu>cd anaconda3

C:\Users\mariu\anaconda3>cd Scripts

C:\Users\mariu\anaconda3\Scripts>activate picasso

(picasso) C:\Users\mariu\anaconda3\Scripts>picasso localize

(picasso) C:\Users\mariu\anaconda3\Scripts>

```

Figure 15: Starting the picasso programs (here: Localize) from windows' command line interface.

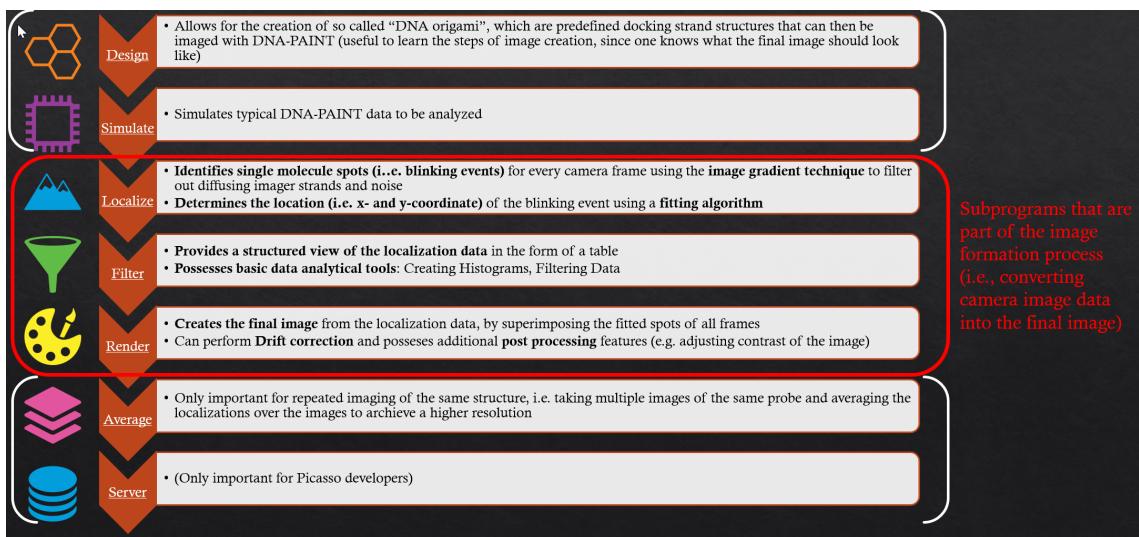


Figure 16: Subprograms of the Picasso software environment. Subprograms relevant for image formation (and discussed in this document) are marked in red.

The environment consists of 7 subprograms. For an in depth explanation of each program check the original picasso publication (Schnitzbauer et al. 2017b) or alternatively the official picasso documentation (Schnitzbauer et al. 2017a). The next three sections focus on the three subprograms that are relevant for the image creation: Localize, Filter, Render.

## 4.2 Subprogram: Localize

The *Localize* program accepts raw camera image data<sup>40</sup> obtained by the DNA-PAINT Method and is able to localize blinking events for individual frames with

---

<sup>40</sup>The accepted filetypes are listed in (Schnitzbauer et al. 2017a)

sub camera pixel precession. This is done in two steps, localization of spots and their respective fitting. Figure 17 shows the interface of the Localize program.

The localization step detects the fluorescent imager strands attached to docking strands in each frame. The spots are detected by the program's *net Gradient* algorithm. The net Gradient  $G_{\text{net}}$  in Equation 11 is a quantity calculated for a square area (= box) around points with the highest photon counts in their local neighbourhood (= local maximum). The box represents the ROI of a spot as discussed in subsection 2.4 and its size can be set before running the localization step. The program accepts a minimum net Gradient and filters out spots that fall below that minimum.

$$G_{\text{net}} = \sum_{\text{box}} \vec{g}_k * \vec{u}_k \quad (11)$$

The net Gradient makes use of the central difference, which is a function that assigns each pixel  $p_k$  of the ROI the difference between its intensity and the intensity of the middle of the ROI.  $G_{\text{net}}$  is then calculated by multiplying the central difference gradient  $\vec{g}_k$  of the intensity at pixel  $p_k$  with the pixel's unit vector in the direction to the center  $\vec{u}_k$ . Since  $\vec{g}_k$  is a 2d vector field, the multiplication with  $\vec{u}_k$  yields a value that measures how strongly the intensity grows towards the center of the ROI.<sup>41</sup> It is essentially a contrast measure that should be proportional to the signal-to-noise ratio (SNR).

Apart from filtering out low contrast spots (which are harder to fit), the net Gradient can also be used to filter out isolated spots (See section 5 and (Wiesner 2024a), section: "Filtering out isolated spots by further analyzing Photon count and net Gradient" for more information on this.)

After the localization, the detected spots are *fitted*, meaning an algorithm is run to determine an estimate of the true position of the fluorescent molecule, as discussed in detail in subsection 2.4. The program currently implements three fitting algorithms:

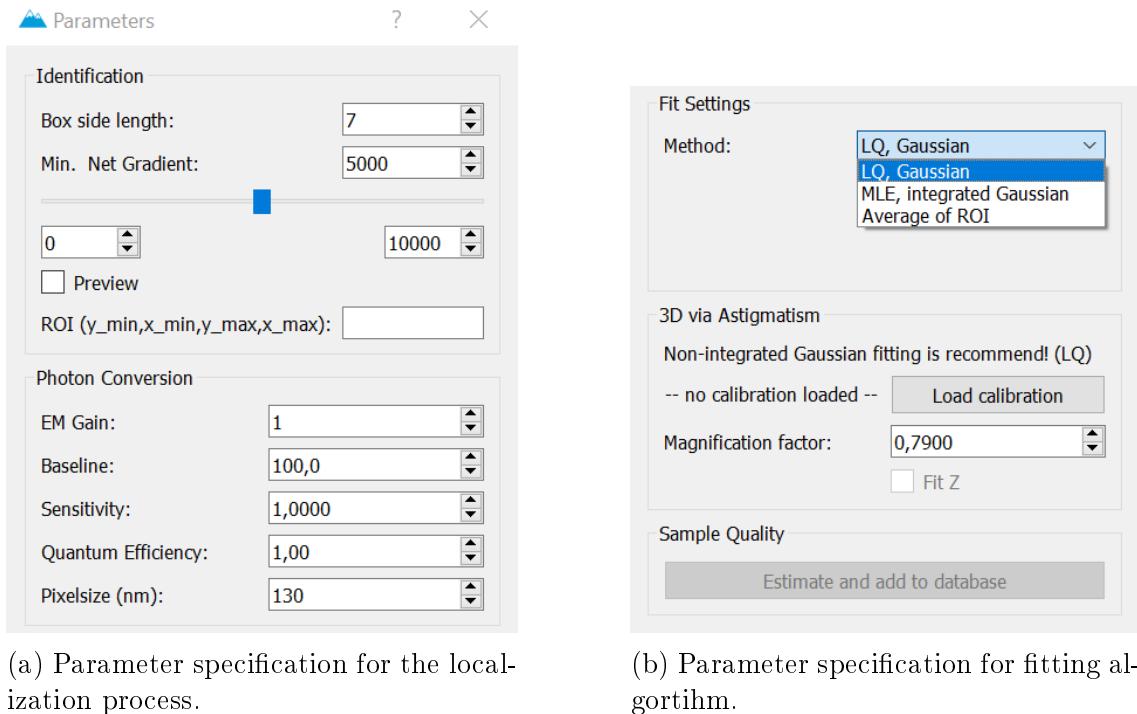
- MLE, Gaussian
- LS, Gaussian
- Average of ROI

All of these algorithms have been explained in detail in subsection 2.4. It has also been shown that a gaussian estimator approach should be chosen when possible, where a MLE optimization is usually to be preferred to a LS optimization. Figure 9

---

<sup>41</sup>Schnitzbauer et al. 2017b.

and Table 1 show that the average estimator approach is very inaccurate and inefficient compared to a gaussian approach. However, if computational time is of great concern, it can represent an alternative.



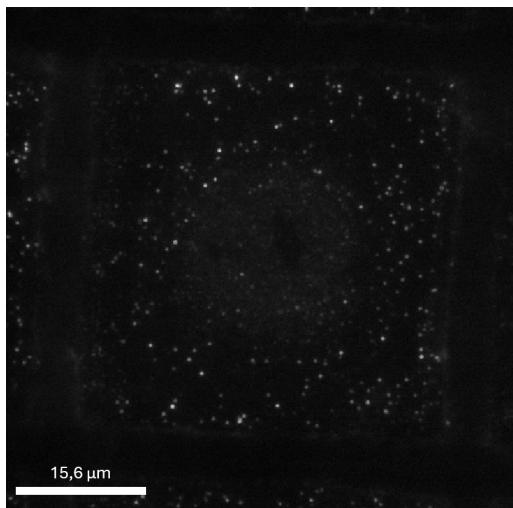
(a) Parameter specification for the localization process.

(b) Parameter specification for fitting algorithm.

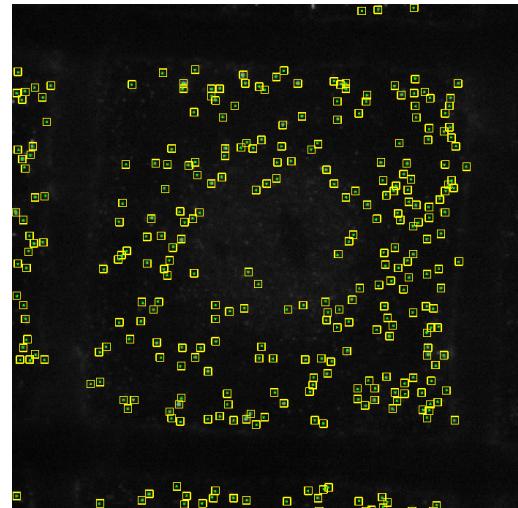
Figure 17: Interface of the Localize program.

Figure 18 shows the frames before (a) and after (b) the localization and fitting process. After running the spot localization and the fitting algorithm the localizations can be saved to a .hdf5 file accompanied by a .yaml file<sup>42</sup>, which holds information on the localization specifications (such as image plane size, pixel size, number of frames, etc.) and is generated automatically by the Localize program. The .yaml file can also be adjusted manually in a simple Texteditor. An example for such a .yaml file can be seen in Figure 19 which was generated by camera data of single walled carbon nano tubes that were imaged with the DNA-PAINT method. The localized and fitted data, i.e. the hdf5 file, can then be used for filtering/data analysis in either the *Render* program or a python environment and for the final image creation in the *Render* program.

<sup>42</sup>Which does not substantially differ from a .txt file.



(a) Image data before running spotting and fitting algorithm.



(b) Image data after running spotting and fitting algorithm. The yellow boxes represent the ROI of a spot. The fitted estimator position is indicated by a green asterisk.

Figure 18: A single frame before (a) and after (b) localizing and fitting in the Localize program.

```

Byte Order: <
Data Type: int16
Frames: 40000
Height: 480
Width: 480
---
Box Size: 7
Fit method: LQ, Gaussian
Generated by: Picasso Localize
Min. Net Gradient: 8000
PixelSize: 130
ROI: null
baseline: 100.0
gain: 1
qe: 1.0
sensitivity: 1.0
---
Filtered by: Marius Wiesner

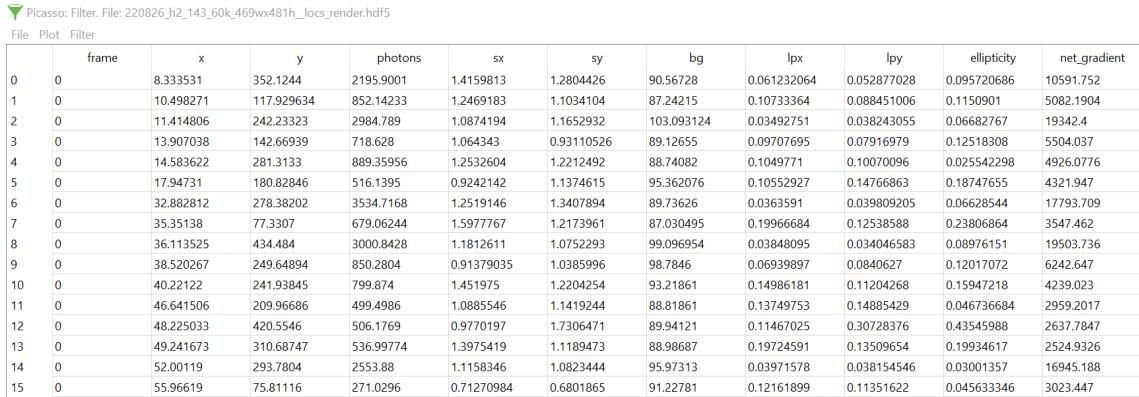
```

Figure 19: A yaml file for storing localization specifications based on localization data of carbon nano tubes.

Source: Generated from carbon nano tubes' DNA-PAINT data by the author.

## 4.3 Subprogram: Filter

The *Filter* program offers a way to filter and analyze the localization data on a very basic level. It also provides a very convenient way of visualizing the localization data in tabular form. Figure 20 provides an example for the carbon nano tubes' localization data (i.e., the .hdf5 file) that was opened with the Filter program.



The screenshot shows a software window titled "Picasso: Filter, File: 220826\_h2\_143\_60k\_469wx481h\_locs\_render.hdf5". The menu bar includes "File", "Plot", and "Filter". The main area displays a table with 16 rows and 11 columns. The columns are labeled: frame, x, y, photons, sx, sy, bg, lpx, lpy, ellipticity, and net\_gradient. The data represents localization points across 16 frames, with each row containing specific coordinates, photon counts, and statistical parameters for each localization.

	frame	x	y	photons	sx	sy	bg	lpx	lpy	ellipticity	net_gradient
0	0	8.333531	352.1244	2195.9001	1.4159813	1.2804426	90.56728	0.061232064	0.052877028	0.095720686	10591.752
1	0	10.498271	117.929634	852.14233	1.2469183	1.1034104	87.24215	0.10733364	0.088451006	0.1150901	5082.1904
2	0	11.414806	242.23323	2984.789	1.0874194	1.1652932	103.093124	0.03492751	0.038243055	0.06682767	19342.4
3	0	13.907038	142.66939	718.628	1.064343	0.93110526	89.12655	0.09707695	0.07916979	0.12518308	5504.037
4	0	14.583622	281.3133	889.35956	1.2532604	1.2212492	88.74082	0.1049771	0.10070096	0.025542298	4926.0776
5	0	17.94731	180.82846	516.1395	0.9242142	1.1374615	95.362076	0.10552927	0.14766863	0.18747655	4321.947
6	0	32.8882812	278.38202	3534.7168	1.2519146	1.3407894	89.73626	0.0363591	0.039809205	0.06628544	17793.709
7	0	35.35138	77.3307	679.06244	1.5977767	1.2173961	87.030495	0.19966684	0.12538588	0.23806864	3547.462
8	0	36.113525	434.484	3000.8428	1.1812611	1.0752293	99.096954	0.03848095	0.034046583	0.08976151	19503.736
9	0	38.520267	249.64894	850.2804	0.91379035	1.0385996	98.7846	0.06939897	0.0840627	0.12017072	6242.647
10	0	40.22122	241.93845	799.874	1.451975	1.2204254	93.21861	0.14986181	0.11204268	0.15947218	4239.023
11	0	46.641506	209.96686	499.4986	1.0885546	1.1419244	88.81861	0.13749753	0.14885429	0.046736684	2959.2017
12	0	48.225033	420.5546	506.1769	0.9770197	1.7306471	89.94121	0.11467025	0.30728376	0.43545988	2637.7847
13	0	49.241673	310.68747	536.99774	1.3975419	1.1189473	88.98687	0.19724591	0.13509654	0.19934617	2524.9326
14	0	52.00119	293.7804	2553.88	1.1158346	1.0823444	95.97313	0.03971578	0.038154546	0.03001357	16945.188
15	0	55.96619	75.81116	271.0296	0.6801865	91.22781	0.12161899	0.11351622	0.045633346	3023.447	

Figure 20: The Filtering program describing the localization data in tabular form.  
Source: Captured in the Filter program by the author.

There are 11 columns describing the localizations (plus an additional column on the far left counting the localizations). Each row represents a localization within a camera frame that was spotted and fitted by the Localization program.

The first column represents the camera frame in which the localization was spotted. The coordinates of the position estimator are saved in the *x* and *y* coordinates. The *photons* column describes the number of photons collected from a spot, while *sx* and *sy* describe the point spread in x and y direction, respectively. The precision of the sub-camera pixel localization (in camera pixels), i.e. the standard deviation of the position estimate (which in the case of a gaussian estimator corresponds to the standard deviation of the gaussian distribution), is given with *lpx* and *lpy*. The *ellipticity* describes the asymmetry of the PSF (relative to a symmetric airy signal). Finally, the *net Gradient* is represented in the last column.

It is also possible to filter the data by specifying min. and max. values for the columns (e.g. a minimum and maximum photon count) and to create one- and two-dimensional histograms of individual columns. However, for more sophisticated filtering and analysis of the data the python library needs to be leveraged which will be explained in section 5.

## 4.4 Subprogram: Render

The program for creating the final image of the structure from the localized and fitted data is *Render*.

The Render program takes, just like the Filter program, localization data (i.e. .hdf5 files accompanied by a .yaml file<sup>43</sup>) as input to create a final image of the localization data. Pictures of the program's interface can be seen in Figure 21. The program creates the image by constructing a two dimensional grid of the x and y coordinates of the position estimators of all frames. An example of this can be seen in Figure 22. Additionally, it creates a contrast that is proportional to the photon count of each localization, i.e. the intensity. Things like contrast colours, contrast range and pixel blurring can be adjusted in the program's settings. It is also possible to zoom in and out to explore specific regions of interest.

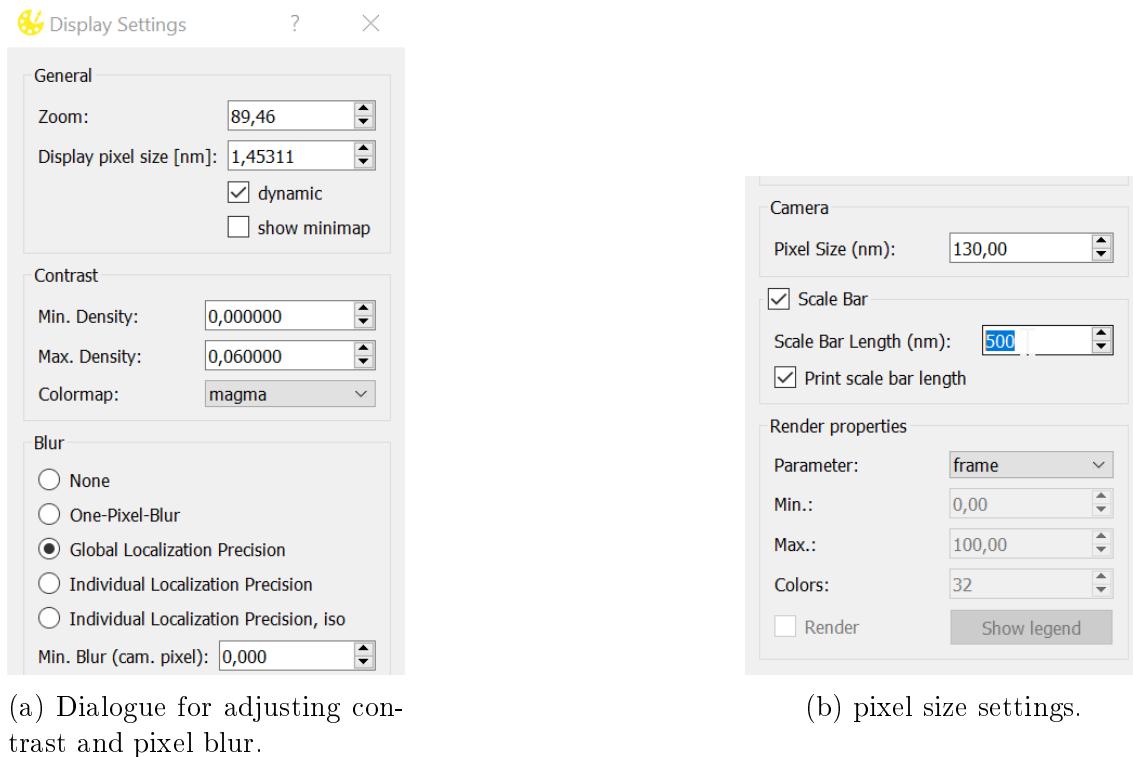


Figure 21: Interface of the Render program.

---

<sup>43</sup>Note that every .hdf5 file needs to have a corresponding .yaml file with the same name. Otherwise Picasso will throw an error.

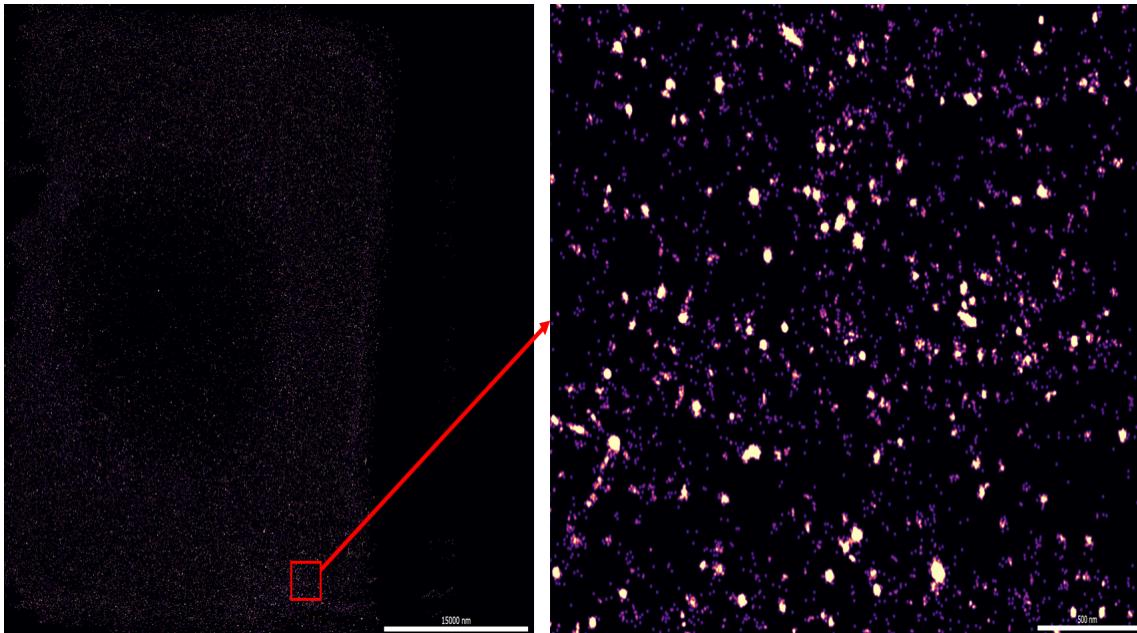


Figure 22: Rendered image of a grid with single walled carbon nanotubes and imaged via the DNA-PAINT method and a zoomed in area. Scale bar on left image: 1500nm. Scale bar on zoomed in image: 500nm.

#### 4.5 The Picasso Library for Data Analysis

As previously mentioned the Picasso software comes with a python library called *picassosr*. This library contains an inout-output-module (called *io*) for importing localized and fitted data, i.e. .hdf5 files, into your custom python script, where it can be analyzed and filtered using either the *postprocess* module or other python libraries (e.g. Pandas, NumPy, etc.). Using the input-output-module, the data can be exported into a .hdf5 file afterwards to render the final image via the *Render* program.

If you want to work with the *picassosr* package you should have used the second installation route by installing the package into a new anaconda environment (see subsection 4.1). The reason for using a new environment is that the *picassosr* package is dependent on older versions of numpy, matplotlib, etc. and other libraries (such as NumPy) may cause compatibility issues. For that reason one should use a different environment (i.e. a standard environment with all up to date packages) to analyze and filter the localization data (see the (Wiesner 2024b) notebook for that part in combination with section 5).

The Jupyter Notebook "WorkingWithThePicassoLibrary.ipynb"<sup>44</sup>, that is attached to this document, provides the code and more in depth explanation to work with the picasso python library.

---

<sup>44</sup>Wiesner 2024a.

## 5 Application: Imaging Single Walled Carbon Nanotubes with DNA-PAINT and Analyzing Functionalisation Properties

To demonstrate how the DNA-PAINT imaging method combined with the Picasso software can be used to create images of nano structures and how to analyze the corresponding data, the attached Jupyter Notebook "WorkingWithPicassoLocalizationData.ipynb" (Wiesner 2024a) was created. It contains the code for filtering and analyzing localization data in a python script and the detailed explanation of it. In this chapter, a brief overlook over the notebooks results is given. The reader is highly encouraged to view the notebooks himself. The Notebooks code can be used for any localization data, but was written specifically for the localization data of carbon nanotubes.

It is possible to extensively filter the localized and fitted data to various parameters. For example, one can filter out isolated spots that would deter the structural image. The basic idea for identifying isolated spots is as follows: A relatively small net Gradient for a relatively hight photon count indicates places of high functionalization density, since the neighbouring space of the spot must be bright. Vice versa, a relatively big net Gradient for a relatively low photon count indicates spots that are rather isolated.

Isolated spots are unlikely to be caused by a docking strand that is part of the CNT structure or they at least distort the final image of the overall structure. Highly isolated spots (i.e. spots with high net Gradient while having a low photon count) should therfore be omitted in order to create a better structural image.

Figure 23 shows the distribution of the spots' net gradient graphed with their corrsponding photon count and a linaer regression line. To omit isolated spots the distribution of the regression residuals is calculated and a multiple of the standard deviation of the residuals is used to set the omission boundary.

Filtering procedures like this (more are given in the notebook<sup>45</sup>) seem to sharpen the resolution of the nanotube structure. Figure 24 shows a rendered image that was created in the notebook with python code. The image was created from the same data as the images in Figure 22, with the difference of being filtered.

The data can also be use to obtain information about the functionalization properties of the nanotubes. For example, the average distance between the position estimate (i.e., the functionalization density) for the data of Figure 24 computes to approximately 8nm (See the notebook for the computation).

---

<sup>45</sup>Wiesner 2024a.

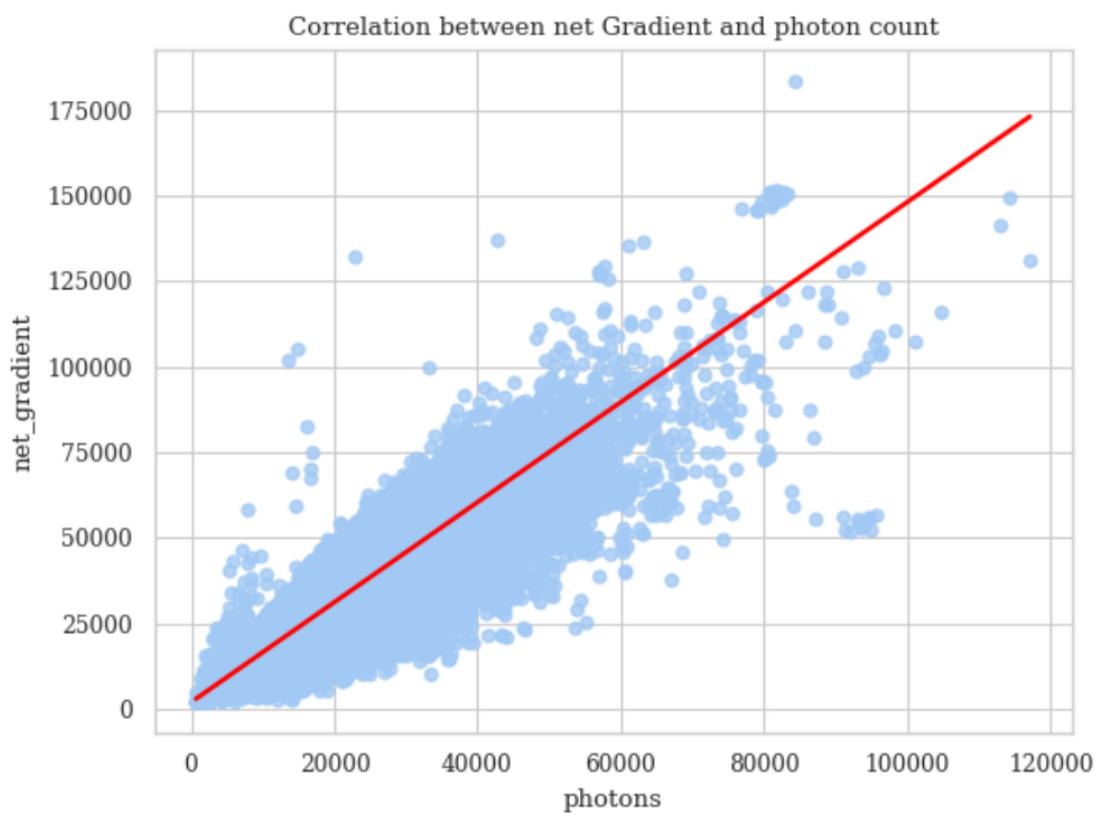


Figure 23: Graph showing the correlation between net Gradient and photon count. Each blue circle represents a position estimate. Isolated spots posses high net Gradient while having low photon count.

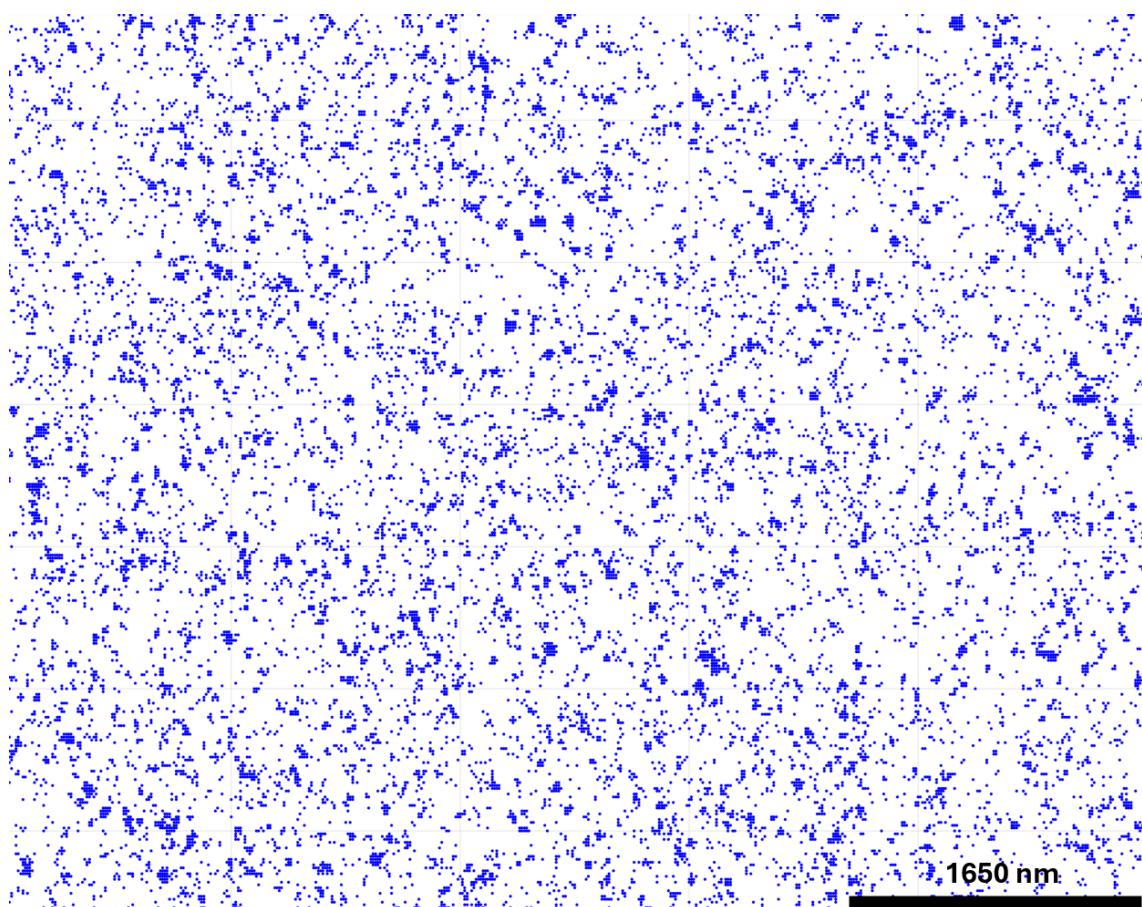


Figure 24: Rendered image after filtering process. The linear structure of the carbon nanotubes can be observed.

## References

- Basche, Th., W.P. Ambrose, and W.E. Moerner (2020). “Optical spectra and kinetics of single impurity molecules in a polymer: spectral diffusion and persistent spectral hole burning”. In: *Journal of the Optical Society of America* 9.5. URL: <https://opg.optica.org/josab/viewmedia.cfm?uri=josab-9-5-829&seq=0>.
- Cousins, Robert (2020). *What is the likelihood function, and how is it used in particle physics?* URL: [https://ep-news.web.cern.ch/what-likelihood-function-and-how-it-used-particle-physics%20\(last%20accessed%20on%20October%202024\)](https://ep-news.web.cern.ch/what-likelihood-function-and-how-it-used-particle-physics%20(last%20accessed%20on%20October%202024)).
- Demchenko, Alexander P. (2020). “Photobleaching of organic fluorophores: quantitative characterization, mechanisms, protection”. In: *Methods and Applications in Fluorescence* 8. DOI: 10.1088/2050-6120/ab7365. URL: <https://iopscience.iop.org/article/10.1088/2050-6120/ab7365>.
- Demtröder, Wolfgang (2006). *Experimentalphysik 2, Elektrizität und Optik (vierte, überarbeitete und erweiterte Auflage)*. Springer. ISBN: 3-540-44794-6.
- Deschout, Hendrik et al. (2014). “Precisely and accurately localizing single emitters in fluorescence microscopy”. In: *Nat Methods* 11, pp. 253–266. DOI: <https://doi.org/10.1038/nmeth.2843>. URL: <https://www.nature.com/articles/nmeth.2843>.
- Diezmann, Lexy von, Yoav Shechtman, and W. E. Moerner (2017). “Three-Dimensional Localization of Single Molecules for Super-Resolution Imaging and Single-Particle Tracking”. In: *Chemical Reviews* 117.11, pp. 7244–7275. DOI: 10.1021/acs.chemrev.6b00629. URL: <https://doi.org/10.1021/acs.chemrev.6b00629>.
- Dobrucki, Jurek W. and Ulrich Kubitscheck (2017). *Fluorescence Microscopy: From Principles to Biological Applications. Second Edition*. Wiley Verlag. ISBN: 9783527338375.
- Eftnik, Maurice R. (2002). “Fluorescence Quenching: Theory and Applications”. In: *Topics in Fluorescence Spectroscopy* 2. DOI: [https://doi.org/10.1007/0-306-47058-6\\_2](https://doi.org/10.1007/0-306-47058-6_2).

- Gibsen, S. Frisken and F. Lanni (1992). “Experimental test of an analytical model of aberration in an oil-immersion objective lens used in three-dimensional light microscopy”. In: *Journal of the Optical Society of America* 9. DOI: <https://doi.org/10.1098/rspa.1959.0200>.
- Manoharan, G. et al. (2023). “Click-Functionalization of Silanized Carbon Nanotubes: From Inorganic Heterostructures to Biosensing Nanohybrids”. In: *Molecules* 28, 2161. URL: <https://doi.org/10.3390/molecules28052161>.
- Manoharan, Gririraj (2021). “Click Functionalization of Carbon Nanotubes for Nano-Bio Applications”. PhD thesis. Department of Physics, University of Osnabrück.
- Moerner, W.E. and David P. Fromm (2003). “Methods of single-molecule fluorescence spectroscopy and microscopy”. In: *Review of Scientific Instruments* 74.8, pp. 3597–3919. DOI: <https://doi.org/10.1063/1.1589587>. URL: <https://pubs.aip.org/aip/rsi/article-abstract/74/8/3597/347480/Methods-of-single-molecule-fluorescence>.
- National High Magnetic Field Laboratory, Optical Microscopy Division of the (2015). *Diffraction of Light*. URL: <https://micro.magnet.fsu.edu/primer/lightandcolor/diffractionintro.html>.
- Richards, B. and E. Wolf (1959). “Electromagnetic diffraction in optical systems, II. Structure of the image field in an aplanatic system”. In: *Proceedings of the Royal Society A* 253.1274. DOI: <https://doi.org/10.1098/rspa.1959.0200>.
- Rottenfusser, Rudi (2024). *Education in Microscopy and Digital Imaging*. URL: <https://zeiss-campus.magnet.fsu.edu/articles/basics/psf.html>.
- Schnitzbauer, Joerg et al. (2017a). *Picasso documentation*. URL: <https://picassosr.readthedocs.io/en/latest/?badge=latest>.
- (2017b). “Super-resolution microscopy with DNA-PAINT”. In: *Nature Protocols* 12, pp. 1198–1228.

Scientific Volume Imaging, B.V. (2024a). *Airy Disc*. URL: <https://svi.nl/AiryDisk>.

- (2024b). *Diffraction limit*. URL: <https://svi.nl/DiffractionLimit>.
- (2024c). *Point Spread Function*. URL: [https://svi.nl/Point-Spread-Function-\(PSF\)](https://svi.nl/Point-Spread-Function-(PSF)).

Small, Alex and Shane Stahlheber (2014). “Fluorophore localization algorithms for super-resolution microscopy”. In: *Nat Methods* 11, pp. 267–279. DOI: <https://doi.org/10.1038/nmeth.2844>. URL: <https://www.nature.com/articles/nmeth.2844>.

Taylor, John R. (1997). *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements. Second Edition*. University Science Books. ISBN: 978-0-935702-75-0.

Wiesner, Marius (2024a). *WorkingWithPicassoLocalizationData.ipynb*. URL: [https://github.com/MariusWiesner/Data\\_Analysis\\_on\\_DNA\\_PAINT\\_image\\_data/tree/master](https://github.com/MariusWiesner/Data_Analysis_on_DNA_PAINT_image_data/tree/master).

- (2024b). *WorkingWithThePicassoLibrary.ipynb*. URL: [https://github.com/MariusWiesner/Data\\_Analysis\\_on\\_DNA\\_PAINT\\_image\\_data/tree/master](https://github.com/MariusWiesner/Data_Analysis_on_DNA_PAINT_image_data/tree/master).