ThunderSTORM: Methodology and Algorithms (update)

Research · October 2015

CITATIONS
0

READS
2,373

5 authors, including:

Guy M Hagen
University of Colorado Colorado Springs
95 PUBLICATIONS **2,368** CITATIONS
SEE PROFILE

Martin Ovesny
Charles University in Prague
21 PUBLICATIONS **1,176** CITATIONS
SEE PROFILE

Zdenek Svindrych
Geisel School of Medicine at Dartmouth
72 PUBLICATIONS **1,616** CITATIONS
SEE PROFILE

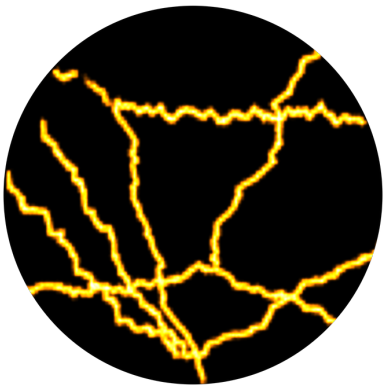
Pavel Křížek
Charles University in Prague
34 PUBLICATIONS **1,422** CITATIONS
SEE PROFILE

Charles University in Prague
First Faculty of Medicine
Institute of Cellular Biology and Pathology

ThunderSTORM: a comprehensive ImageJ plugin for PALM and STORM data analysis and super-resolution imaging

Methodology and Algorithms

Version 1.3+



Martin Ovesný, Pavel Křížek, Josef Borkovec, Zdeněk Švindrych, and Guy M. Hagen

About this text

This is a supplementary note for ThunderSTORM, an open source, modular plug-in for ImageJ. ThunderSTORM provides tools for automated processing, analysis, and visualization of data acquired by single molecule localization microscopy methods such as PALM or STORM. The plug-in, source code, and associated documentation are hosted at Google Code (https://code.google.com/p/thunder-storm/). ThunderSTORM is distributed under the terms of the GNU GPLv3 license.

Acknowledgments

This work was supported by the Czech Science Foundation [P304/09/1047, P205/12/P392, P302/12/G157, 14-15272P]; by European Union Funds for Regional Development [OPPK CZ.2.16/3.1.00/24010]; and by Charles University in Prague [Prvouk/1LF/1, UNCE 204022].

Contents

Notation

Scalars, vectors, and matrices

x	scalar
$\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^\top$	column vector with n entries, $i = 1, 2, \ldots, n$
$X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m]$	$m \times n$ matrix
$X(x, y)$	element value of the matrix X at the position given by the column x and row y

General mathematical symbols, variables and operations

$\lfloor \cdot \rfloor$	round down to integer ("floor")
$\lceil \cdot \rceil$	round up to integer ("ceiling")
$f(a \mid b)$	function of the variable a given the parameters b
$a \bmod b$	remainder after division of a by b
$\arg\min_x f(x)$	the value of x that yields the minimum of $f(x)$
$\arg\max_x f(x)$	the value of x that yields the maximum of $f(x)$
$\ln(x)$	natural logarithm of x
$\exp(x) = \mathrm{e}^x$	exponential function of x
$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \mathrm{e}^{-t^2} \, \mathrm{d}t$	error function of x
$\operatorname{atan2}(y, x) = 2 \arctan \frac{\sqrt{x^2+y^2}-x}{y}$	two-argument arctangent function

Introduction

ThunderSTORM is an open-source, interactive, and modular plug-in for ImageJ [1] designed for automated processing, analysis, and visualization of data acquired by single molecule localization microscopy (SMLM) methods such as stochastic optical reconstruction microscopy (STORM) [2] or photoactivated localization microscopy (PALM) [3, 4]. The software can run on computers with different operating systems, supports the ImageJ macro language, and is also compatible with other ImageJ-based applications such as Fiji [5] or μManager [6].

ThunderSTORM can process data for both 2D and 3D SMLM imaging, including data with high spatial molecular density, which is known as the “crowded field" problem. The steps involved in SMLM data processing are shown in Figure 1. Several algorithms for each of the processing steps have been implemented so experienced users have many options to adapt the processing to their data. However, the default settings perform very well on many of the SMLM data sets we have experimented with. Here we describe the algorithmic implementation of the methods involved in processing the data.

ThunderSTORM is also capable of generating simulated SMLM data and of evaluation of the performance of localization algorithms based on the ground-truth positions of the molecules. This allows users to perform Monte Carlo simulations [7, 8] and to quantitatively evaluate the performance of the applied algorithms. Localization results, as well as the ground-truth positions of molecules, can be imported/exported to/from ThunderSTORM in a variety of data formats, allowing compatibility with other SMLM localization software.

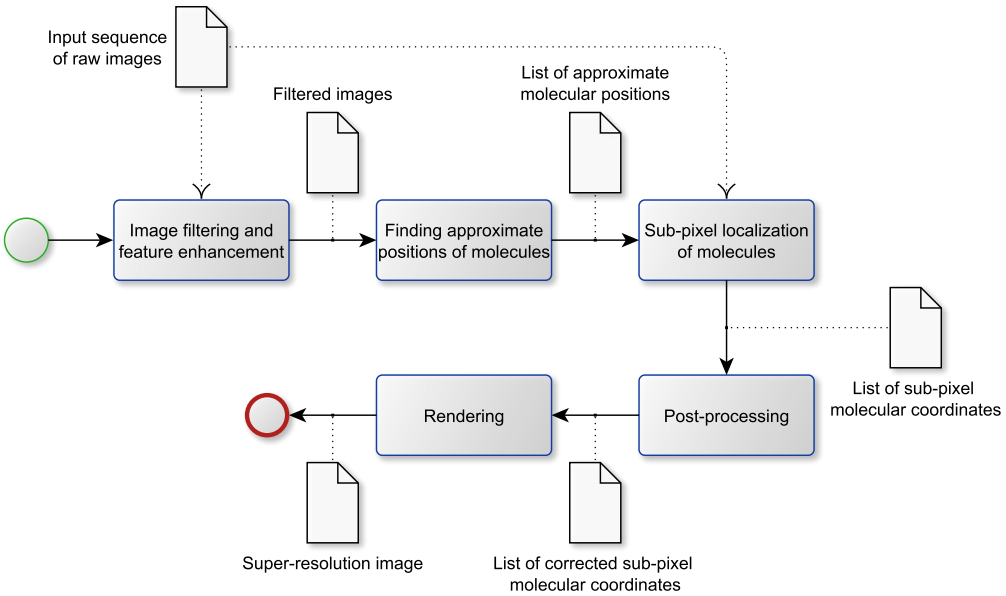


Figure 1: Data processing pipeline for single molecule super-resolution imaging.

Part I.
RAW DATA PROCESSING

1. Image filtering and feature enhancement

Image filtering and feature enhancement is one of the first steps in many image processing applications. For this purpose, ThunderSTORM provides several low-pass and band-pass convolution-based image filters. Users may also choose not to use a filter and leave the input data as it is for further processing.

Input: A raw image I of size $m \times n$ pixels.

Output: A filtered image F of size $m \times n$ pixels.

Convolution

A two-dimensional discrete convolution is defined as

$$(I * K)(x, y) = \sum_{-\infty}^{+\infty} \sum_{-\infty}^{+\infty} I(u, v) K(x - u, y - v) , \tag{1}$$

where I is an image of size $m \times n$, and K is a convolution kernel of arbitrary size. Values outside the domains of I and K are set to zero.

Convolution can be a time demanding operation when implemented according to the definition in Equation (1). To speed up the algorithms, we use a method known as convolution with separable kernels. This allows one to calculate the convolution of the image I with the kernel K as

$$F = I * K = (I * \boldsymbol{k}) * \boldsymbol{k}^\top \tag{2}$$

if the kernel K can be written as $K = \boldsymbol{k}\boldsymbol{k}^\top$. Here the kernel K is an $l \times l$ matrix and $\boldsymbol{k} = [k_1, k_2, \ldots, k_l]^\top$. Using the method of separable kernels reduces the time complexity of the convolution from $O(mnl^2)$ to $O(mnl)$.

1.1. Averaging filter

Applying an averaging filter (also known as mean or box filter) results in an image in which the value of each pixel is equal to the average intensity of pixels from the input image in a given neighborhood. Filtering can be defined as a convolution of the image with a kernel given by an $l \times l$ matrix containing equal entries that sum to one, thus

$$K_{\mathrm{AV}} = \frac{1}{l^2} \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} . \tag{3}$$

Kernel K_{AV} is separable and the vector $\boldsymbol{k}$ in Equation (2) can be composed from values $k_i = 1/l$, where $i = 1, \ldots, l$. The kernel size l is a user-specified parameter.

1.2. Gaussian filter

Convolution with a Gaussian kernel (also referred to as a Gaussian blur or as a Gaussian filter) is one of the most commonly used filters in image processing. The Gaussian kernel is formed by a matrix that contains the values of a rotationally symmetric Gaussian function

$$K_{\mathrm{G}}\left(x, y \mid \sigma\right) = a^2 \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \tag{4}$$

Kernel K_{G} of size $l \times l$ is separable and the vector $\boldsymbol{k}$ in Equation (2) can be composed from values $k_i\left(x \mid \sigma\right) = a \exp\left(-\frac{x^2}{2\sigma^2}\right)$, where $i = 1, \ldots, l$, $x = i-(l+1)/2$, $l = 1+2\lceil 3\sigma \rceil$, and a is a scaling factor such that $\sum_i k_i = 1$. Users need to input the standard deviation σ.

1.3. Lowered Gaussian filter

This filter was reported to perform well in the DAOPHOT [9] and DAOSTORM [10] algorithms. The convolution kernel is based on the Gaussian kernel in Equation (4) which has been lowered to have the sum of all its entries equal to zero,

$$K_{\mathrm{LG}}\left(x, y \mid \sigma\right) = K_{\mathrm{G}}\left(x, y \mid \sigma\right) - b\,, \tag{5}$$

where b is the mean value of all of the elements in K_{G}. The standard deviation σ is a user-specified parameter.

Although the kernel K_{LG} is not separable, the filtered image can be obtained by subtracting two images filtered with two separable kernels,

$$F = I * K_{\mathrm{LG}} = I * K_{\mathrm{G}} - I * K_{\mathrm{AV}}\,. \tag{6}$$

The lowered Gaussian is a band-pass filter. The sizes of both K_{G} and K_{AV} kernels are computed as $l = 1 + 2\lceil 3\sigma \rceil$.

1.4. Difference-of-Gaussians filter

The difference-of-Gaussians filter is a feature enhancement algorithm which acts as a band-pass filter. The convolution kernel is formed as the difference of two Gaussian kernels

$$K_{\mathrm{DoG}}\left(x, y \mid \sigma_1, \sigma_2\right) = K_{\mathrm{G1}} - K_{\mathrm{G2}}\,, \tag{7}$$

where the standard deviation $\sigma_2 > \sigma_1$, and the kernels are given by $K_{\mathrm{G1}} = K_{\mathrm{G}}\left(x, y \mid \sigma_1\right)$ and $K_{\mathrm{G2}} = K_{\mathrm{G}}\left(x, y \mid \sigma_2\right)$. The sizes for both kernels are computed as $l = 1 + 2\lceil 3\sigma_2 \rceil$. The standard deviations σ_1 and σ_2 are user-specified parameters.

Although the kernel K_{DoG} is not separable, the filtered image can be obtained by subtracting two images filtered with Gaussian kernels having different standard deviations,

$$F = I * K_{\mathrm{DoG}} = I * K_{\mathrm{G1}} - I * K_{\mathrm{G2}}\,. \tag{8}$$

1.5. Difference of averaging filters

The difference of averaging filters [11] is another type of a band-pass filter. The filtered image is obtained by subtracting images filtered with two averaging filters having different user-specified kernel sizes,

$$F = I * K_{\mathrm{AV1}} - I * K_{\mathrm{AV2}} \,. \tag{9}$$

Similar to the difference-of-Gaussians filter, the size of the kernel K_{AV2} has to be larger than K_{AV1}.

1.6. Wavelet filter

The wavelet transform is commonly used in modern signal-processing applications. This type of wavelet filter applies the *à trous* algorithm [12, 13] which is an undecimated scheme in which the filter responses are up-sampled, thereby inserting holes (*trous* in French) between the filter coefficients. The output of each filter level, therefore, contains the same number of samples as the input.

The wavelet transform of the input data at level $j = 1, \ldots, N$ is computed as

$$F_j = V_{j-1} - V_j \,, \tag{10}$$

where

$$V_j = (V_{j-1} * \boldsymbol{k}_j) * \boldsymbol{k}_j^{\top} \,. \tag{11}$$

Here $V_0 = I$ is the input image, $\boldsymbol{k}_j$ is a filter kernel at the corresponding level, and l is the kernel size. The number of levels is given by N.

The expression in Equation (10) acts as a band-pass filter. Each wavelet level thus corresponds to a different range of spatial frequencies. The first level F_1 contains mainly the high spatial frequencies present in the input image including noise. Higher levels contain ranges of lower and lower image frequency components. In our case, the output of the wavelet filter is the second wavelet level F_2. The standard deviation of values obtained in the first wavelet level F_1 can be used as a molecular detection threshold in the filtered image [13], see Section 2.4.

The wavelet filter typically uses a convolution kernel based on normalized B-spline basis functions [12, 13]. In our case, the B-spline basis function of order $q \in \mathbb{N}^+$ in the variable $t \in \mathbb{R}$, written as $B_q(t)$, is defined recursively as follows. For $q = 1$,

$$B_1(t) = \begin{cases} 1 & \text{if } 0 \le t < 1 \,, \\ 0 & \text{otherwise} \,, \end{cases} \tag{12}$$

and for $q > 1$,

$$B_q(t) = \frac{t}{q-1} B_{q-1}(t) + \frac{q-t}{q-1} B_{q-1}(t-1) \,. \tag{13}$$

Here $B_q(t)$ has non-zero values only in the interval $0 \le t < q$.

The first wavelet level F_1 is computed using the convolution kernel $\boldsymbol{k}_1 = [k_{1,1}, \ldots, k_{1,l}]$, where $k_{1,i}(x \mid q, s) = a B_q\left(\frac{x}{s} + \frac{q}{2}\right)$, $i = 1, \ldots, l$, $x = i - (l+1)/2$, and $l = 2\left\lceil \frac{qs}{2} \right\rceil - 1$.

Here $s > 0$ defines a scaling factor of the variable x, and a is a scaling factor such that $\sum_i k_{1,i} = 1$. Kernels for higher wavelet levels are obtained by inserting zeros between the kernel coefficients. Users need to input the order q of the B-spline basis function and the scaling factor s.

In our implementation, the default kernel uses a B-spline basis function of the third order with a scaling factor $s = 2$. These settings yield kernels $\boldsymbol{k}_1 = \left[\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16}\right]^\top$ and $\boldsymbol{k}_2 = \left[\frac{1}{16}, 0, \frac{1}{4}, 0, \frac{3}{8}, 0, \frac{1}{4}, 0, \frac{1}{16}\right]^\top$, as suggested in [12, 13]. Examples of kernels based on B-spline basis functions are shown in Figure 2.

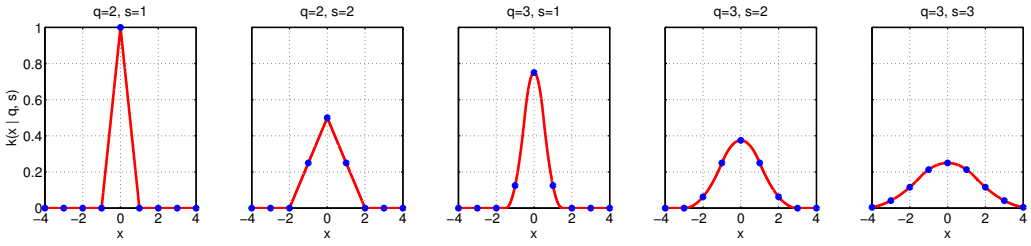


Figure 2: Examples of convolution kernels generated with B-spline basis functions. Blue points correspond to sampling of the B-spline basis function (in red).

1.7. Median filter

A median filter is a non-linear image filter often used to remove impulse noise (i.e., salt and pepper noise). The value of each pixel in the resulting image is equal to the median of the intensity values from the original image in a given neighborhood. If the neighborhood mask has an odd number of entries, the median is the middle element of all numerically sorted entries. For an even number of entries, the median is calculated as the mean of the two entries in the middle. Users need to input the kernel size and the type of neighborhood mask, which can be a cross (4-connected) or box (8-connected) pattern.

1.8. No filter

This selection leaves the input data as it is for further processing, i.e., $F = I$.

2. Finding approximate positions of molecules

Finding approximate positions of the molecules in the input images follows the image filtering and feature enhancement step described above. ThunderSTORM offers three algorithms for this purpose: detection of local intensity maxima, non-maximum suppression, or calculation of the centroid of connected components of segmented objects. All three methods need a user-specified threshold defining an intensity below which possible molecular detections will be rejected.

Input: A filtered image F of size $m \times n$ pixels.

Output: A list of coordinates $\{(\tilde{x}_p, \tilde{y}_p)\}_{p=1}^{\tilde{P}}$ with approximate molecular positions.

2.1. Detection of local intensity maxima

This method passes through each pixel of the filtered image F and determines if the pixel intensity $F(x, y)$ is greater than a user-specified threshold (Section 2.4) and at the same time greater than or equal to all values within a 4- or 8-connected neighborhood. If this condition holds, the point is accepted as a candidate molecule for further processing.

2.2. Non-maximum suppression

Our implementation of non-maximum suppression is based on morphological gray-scale dilation [14] defined by the formula

$$(F \oplus S)(x, y) = \max_{(x', y') \in D_S} \left[F(x - x', y - y') + S(x', y') \right] , \tag{14}$$

where F is the filtered image, S is a structuring element, and D_S is the domain of S. Gray-scale dilation is a more general approach than the method of detection of local intensity maxima described in Section 2.1, because the structuring element can have an arbitrary form.

We first perform a gray-scale dilation of the filtered image F using a structuring element S of size $l \times l$ with all entries set to zero, where $l = 2r + 1$, and r is a dilation radius given by the user. The approximate molecular positions are extracted by finding pixels in the image F that have intensities equal to corresponding pixels in the dilated image and also have an intensity greater than a user-specified threshold (Section 2.4). Essentially, a binary image is built with all non-maximum pixels set to zero and with all local maxima set to one.

2.3. Centroid of connected components

This algorithm is based on segmentation of molecules which is performed by applying a user-defined threshold (Section 2.4) on the filtered image F. Users also have the option to combine the thresholded image with the result of a watershed algorithm [14] applied

to the image F. This procedure was proposed in [13]. The position of each molecule is determined by computing the centroid of the relevant segmented object using

$$\tilde{x}_p = \frac{1}{c_p} \sum_{i=1}^{c_p} x_{i,p} \,, \quad \tilde{y}_p = \frac{1}{c_p} \sum_{i=1}^{c_p} y_{i,p} \,. \tag{15}$$

Here p indexes the objects, c_p is the number of elements in each object, and $x_{i,p}$ and $y_{i,p}$ are the integer pixel coordinates within each object. Segmented objects are determined as connected components by an algorithm based on a breadth-first search [15]. The watershed algorithm is based on the ImageJ "Find Maxima" function.

2.4. Threshold selection

When finding the approximate position of molecules, choosing the right threshold value is important for obtaining good results, because the threshold influences the number of missed molecules (false negatives) and the number of erroneous detections of non-existing molecules (false positive detections). ThunderSTORM uses a single-valued intensity threshold which is updated for every raw input image I and applied to the filtered image F. The threshold value can be specified by users as an expression combining mathematical functions and operators with variables based on the current raw or filtered image, see User's Guide Section 7. This is a powerful option, because users can specify the threshold value systematically for unknown input images, in which the global intensity may slowly fluctuate over time. Izeddin et al. [13] suggested to use the second wavelet level F_2 as the filtered image F and to choose the threshold value between 0.5 and 2 times the standard deviation of the intensity values from the first wavelet level F_1, see Section 1.6 for the definition of wavelet levels.

3. Sub-pixel 2D localization of molecules

Sub-pixel localization of single molecules with an accuracy below the diffraction limit is the basis of SMLM methods. ThunderSTORM supports the following localization methods: calculation of the centroid of the local neighborhood [16], the radial symmetry method [17], and fitting of point-spread function models by (weighted) least-squares methods or by maximum likelihood estimation [18, 19]. Users may also choose not to use any of the methods, thereby using the approximate positions of the molecules determined in the previous step. Note that the choice of method for sub-pixel localization of molecules is independent of the pre-processing methods described above.

Input: A raw image I of size $m \times n$ pixels and a list of coordinates $\{(\tilde{x}_p, \tilde{y}_p)\}_{p=1}^{\tilde{P}}$ with approximate molecular positions.

Output: A list of coordinates $\{(\hat{x}_p, \hat{y}_p)\}_{p=1}^{\hat{P}}$ with sub-pixel positions of the molecules.

Definition of the fitting region

Given the approximate position of a molecule $(\tilde{x}_p, \tilde{y}_p)$ and a user-specified fitting radius $r > 0$, we define $\mathcal{D} = \{-r, \ldots, r\} \times \{-r, \ldots, r\}$ as a set of integer (x, y) coordinates, and $\tilde{I}(x, y) = I(x + \tilde{x}_p, y + \tilde{y}_p)$ as intensity values of an $l \times l$ sub-image centered at the point $(\tilde{x}_p, \tilde{y}_p)$ of the raw input image I, where $l = 2r + 1$ is the size of the sub-image. The desired sub-pixel coordinates of the molecule are obtained as $\hat{x}_p = \hat{x}_0 + \tilde{x}_p$ and $\hat{y}_p = \hat{y}_0 + \tilde{y}_p$, where $\hat{x}_0$ and $\hat{y}_0$ define the sub-pixel refinements of the coordinates obtained by the methods described below.

3.1. No estimator

This option returns only the list of approximate molecular positions obtained by methods described in Section 2.

3.2. Centroid of local neighborhood

Calculation of the centroid in a local image neighborhood is a very fast method for sub-pixel molecule localization and is used in [16]. The algorithm does not estimate the intensity or imaged size of molecules, and is sensitive to noise. The main idea is simply to calculate the mean pixel positions weighted by the intensity of the image data. The sub-pixel refinement of the coordinates is calculated as

$$\hat{x}_0 = \frac{\sum_{x,y \in \mathcal{D}} x\tilde{I}(x,y)}{\sum_{x,y \in \mathcal{D}} \tilde{I}(x,y)}, \quad \hat{y}_0 = \frac{\sum_{x,y \in \mathcal{D}} y\tilde{I}(x,y)}{\sum_{x,y \in \mathcal{D}} \tilde{I}(x,y)}. \tag{16}$$

3.3. Radial symmetry

This algorithm finds the sub-pixel position of a molecule by determining the point with maximal radial symmetry in the data as described in [17]. The general idea is to find the origin of radial symmetry (i.e., the center of a molecule) as the point with the minimum

distance to gradient-oriented lines passing through all data points. The calculation of each molecular position is very fast due to an analytical solution, but the algorithm does not estimate the intensity or imaged size of a molecule. Radial symmetry is a robust feature in SMLM data, making the algorithm resistant to noise.

The calculation starts by determining the intensity co-gradient

$$\nabla \tilde{I}_{x,y} = \left[\frac{\partial \tilde{I}\,(x,y)}{\partial u}, \frac{\partial \tilde{I}\,(x,y)}{\partial v} \right]^{\top} \tag{17}$$

for every integer point (x, y) from the set $\mathcal{D}$. Here uv coordinates are rotated by 45° from the xy coordinate system of the image, because partial derivatives are determined using the Roberts cross operator [14] as

$$\begin{aligned} \frac{\partial \tilde{I}\,(x,y)}{\partial u} &= \tilde{I}\,(x+1, y+1) - \tilde{I}\,(x,y)\ , \\ \frac{\partial \tilde{I}\,(x,y)}{\partial v} &= \tilde{I}\,(x, y+1) - \tilde{I}\,(x+1,y)\ . \end{aligned}$$

The computed co-gradient $\nabla \tilde{I}_{x,y}$ corresponds to the point $A_{x,y} = (x + 0.5, y + 0.5)$ and the slope of a gradient-oriented line passing through the point $A_{x,y}$ is, in a xy coordinate system, given by

$$s_{x,y} = \left(\frac{\partial I\,(x,y)}{\partial u} + \frac{\partial I\,(x,y)}{\partial v} \right) \left(\frac{\partial I\,(x,y)}{\partial u} - \frac{\partial I\,(x,y)}{\partial v} \right)^{-1} . \tag{18}$$

The origin of the radial symmetry $(\hat{x}_0, \hat{y}_0)$ can be determined as the point that minimizes the sum of weighed distances of all considered lines to that point. It can be shown [17] that the problem has an analytical solution. For simplicity, let index $j = x\,(2r+1) + y$, where $x, y \in \mathcal{D}$, thus $s_j = s_{x,y}$, $w_j = w_{x,y}$, and $b_j = b_{x,y}$. The analytical solution is given by the following equations

$$\hat{x}_0 = a^{-1} \left[\Big(\sum_{j \in \mathcal{D}} s_j w_j b_j \Big) \Big(\sum_{j \in \mathcal{D}} w_j \Big) - \Big(\sum_{j \in \mathcal{D}} w_j b_j \Big) \Big(\sum_{j \in \mathcal{D}} s_j w_j \Big) \right] , \tag{19}$$

$$\hat{y}_0 = a^{-1} \left[\Big(\sum_{j \in \mathcal{D}} s_j w_j b_j \Big) \Big(\sum_{j \in \mathcal{D}} s_j w_j \Big) - \Big(\sum_{j \in \mathcal{D}} w_j b_j \Big) \Big(\sum_{j \in \mathcal{D}} s_j^2 w_j \Big) \right] , \tag{20}$$

where $a = \left(\sum_{j \in \mathcal{D}} s_j w_j\right)^2 - \left(\sum_{j \in \mathcal{D}} s_j^2 w_j\right)\left(\sum_{j \in \mathcal{D}} w_j\right)$, $b_j = y - s_j x$, and $w_j = \frac{\tilde{w}_j}{s_j^2 + 1}$. The point-to-line distances are weighted by the factor $\tilde{w}_j = \tilde{w}_{x,y} = \frac{\left|\nabla \tilde{I}_{x,y}\right|^2}{d\left(A_{x,y}, C\right)}$, where $d(A_{x,y}, C)$ is the Euclidean distance of point $A_{x,y}$ to the centroid C. The centroid is computed analogously to Equation (16) from the gradient magnitudes $\left|\nabla \tilde{I}_{x,y}\right|$.

The implementation of the radial symmetry method is taken from [17], where the partial derivatives are additionally smoothed by a 3×3 averaging filter, see Section 1.1, in order to reduce noise and improve the accuracy of the results.

3.4. Fitting point-spread function models

The impulse response of a microscope to a point-like source is described by the point-spread function (PSF). Because a single molecule emitter can be treated as an incoherent point source in SMLM data, the result of fitting a PSF model to an image of a single molecule is an estimate of the molecular position, its imaged size, and its intensity. ThunderSTORM offers a choice between several PSF models and fitting procedures based on (weighted or unweighted) least-squares methods and maximum likelihood estimation [18, 19].

3.4.1. PSF models

It has been shown [20, 21] that the Gaussian function provides a very good approximation of the real PSF of a microscope. This is mainly due to pixelation effects and the presence of noise, which makes the difference between the Gaussian function and the real PSF negligible. The advantage of Gaussian PSF models are their simplicity, robustness, and computational efficiency.

Symmetric 2D Gaussian function

A common approximation of the real PSF is a symmetric two-dimensional Gaussian function given by the formula

$$\mathrm{PSF_G}\left(x, y \mid \boldsymbol{\theta}\right) = \frac{\theta_N}{2\pi\theta_\sigma^2} \exp\left(-\frac{(x-\theta_x)^2 + (y-\theta_y)^2}{2\theta_\sigma^2}\right) + \theta_b\,, \tag{21}$$

where $\mathrm{PSF_G}\left(x, y \mid \boldsymbol{\theta}\right)$ gives the expected photon count at the integer pixel position (x, y) for the parameters $\boldsymbol{\theta} = [\theta_x, \theta_y, \theta_\sigma, \theta_N, \theta_b]$. The entries of the vector $\boldsymbol{\theta}$ are as follows: θ_x and θ_y are the sub-pixel molecular coordinates, θ_σ is the imaged size of the molecule, θ_N corresponds to the total number of photons emitted by the molecule, and θ_b corresponds to the background offset.

Integrated form of a symmetric 2D Gaussian function

The integrated form of a symmetric two-dimensional Gaussian function can be used to help to take into account the discrete nature of pixels present in digital cameras [22, 11]. Assuming a uniform distribution of pixels with unit size, a single molecule intensity profile can be expressed as

$$\mathrm{PSF_{IG}}\left(x, y \mid \boldsymbol{\theta}\right) = \theta_N E_x E_y + \theta_b\,, \tag{22}$$

where $\mathrm{PSF_{IG}}\left(x, y \mid \boldsymbol{\theta}\right)$ gives the expected photon count at the integer pixel position (x, y) for the parameters $\boldsymbol{\theta} = [\theta_x, \theta_y, \theta_\sigma, \theta_N, \theta_b]$ and

$$E_x = \frac{1}{2}\,\mathrm{erf}\left(\frac{x-\theta_x+\frac{1}{2}}{\sqrt{2}\theta_\sigma}\right) - \frac{1}{2}\,\mathrm{erf}\left(\frac{x-\theta_x-\frac{1}{2}}{\sqrt{2}\theta_\sigma}\right), \tag{23}$$

$$E_y = \frac{1}{2}\,\mathrm{erf}\left(\frac{y-\theta_y+\frac{1}{2}}{\sqrt{2}\theta_\sigma}\right) - \frac{1}{2}\,\mathrm{erf}\left(\frac{y-\theta_y-\frac{1}{2}}{\sqrt{2}\theta_\sigma}\right). \tag{24}$$

The entries of the vector $\boldsymbol{\theta}$ are as follows: θ_x and θ_y are the sub-pixel molecular coordinates, θ_σ is the imaged size of the molecule, θ_N corresponds to the total number of photons emitted by the molecule, and θ_b corresponds to the background offset.

3.4.2. Least-squares methods

To approximate the data with a point-spread function, least-squares methods [18, 23, 19] are employed to minimize the sum of (weighted) squared residuals defined by

$$\chi^2\left(\boldsymbol{\theta} \mid \mathcal{D}\right) = \sum_{x,y\in\mathcal{D}} w\left(\tilde{I}\left(x,y\right) - \mathrm{PSF}\left(x,y \mid \boldsymbol{\theta}\right)\right)^2 . \tag{25}$$

Here the residual value for the (x,y) data point is defined as the difference between the observed image intensity $\tilde{I}\left(x,y\right)$ and the value approximated by the $\mathrm{PSF}\left(x,y \mid \boldsymbol{\theta}\right)$, where $\boldsymbol{\theta}$ are the PSF parameters. The residual value can be further weighted by $w = 1$, making all measurements equally significant, or weighted by $w = 1/\tilde{I}\left(x,y\right)$, which takes into account the uncertainty in the number of detected photons.

The search for parameters $\hat{\boldsymbol{\theta}}$ which minimize $\chi^2\left(\boldsymbol{\theta} \mid \mathcal{D}\right)$, leads to an optimization problem formulated as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \chi^2\left(\boldsymbol{\theta} \mid \mathcal{D}\right) , \tag{26}$$

which we solve by the Levenberg-Marquardt algorithm as implemented in the Apache Commons Math library [24]. The starting point for the optimization process is computed from the data as the difference between the maximum and the minimum intensity values for the molecular intensity θ_N, and as the minimum intensity value for the background offset θ_b. Users have to choose the starting point for the approximate molecular width θ_σ. The sub-pixel refinement of the coordinates is obtained as $\hat{x}_0 = \hat{\theta}_x$ and $\hat{y}_0 = \hat{\theta}_y$, where $\hat{\boldsymbol{\theta}} = \left[\hat{\theta}_x, \hat{\theta}_y, \ldots\right]$.

3.4.3. Maximum-likelihood estimation

This approach assumes that the number of photons collected by a single camera pixel follows the Poisson distribution. Thus, the probability of κ photons arriving at a camera pixel, where the expected number of photons is λ, is given by

$$p\left(\kappa \mid \lambda\right) = \frac{\lambda^\kappa \mathrm{e}^{-\lambda}}{\kappa!} . \tag{27}$$

Suppose that samples are drawn independently from the Poisson distribution, with the expected photon count $\lambda = \mathrm{PSF}\left(x,y \mid \boldsymbol{\theta}\right)$ given by the point-spread function model, and the observed photon count $\kappa = \tilde{I}\left(x,y\right)$ given by the image intensity expressed in photons. The likelihood [18, 19, 22, 11] of the parameters $\boldsymbol{\theta}$ can be modeled as

$$L(\boldsymbol{\theta} \mid \mathcal{D}) = \prod_{x,y\in\mathcal{D}} \frac{\mathrm{PSF}\left(x,y \mid \boldsymbol{\theta}\right)^{\tilde{I}(x,y)}\, \mathrm{e}^{-\mathrm{PSF}(x,y|\boldsymbol{\theta})}}{\tilde{I}\left(x,y\right)!} . \tag{28}$$

The maximum likelihood estimate of the parameters $\boldsymbol{\theta}$ is, by definition, the value that maximizes the likelihood $L(\boldsymbol{\theta} \mid \mathcal{D})$. Intuitively, the estimate $\hat{\boldsymbol{\theta}}$ corresponds to the value $\boldsymbol{\theta}$ that best agrees with the data. The maximization problem has the form

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \sum_{x,y \in \mathcal{D}} \left[\tilde{I}\left(x,y\right) \ln\left(\mathrm{PSF}\left(x,y \mid \boldsymbol{\theta}\right)\right) - \mathrm{PSF}\left(x,y \mid \boldsymbol{\theta}\right) \right], \tag{29}$$

which we solve by the Nelder-Mead method [25]. The starting point for the optimization process is computed from the data as the difference between the maximum and the minimum intensity values for the molecular intensity θ_N, and as the minimum intensity value for the background offset θ_b. Users have to choose the starting point for the approximate molecular width θ_σ. The sub-pixel refinement of the coordinates is obtained as $\hat{x}_0 = \hat{\theta}_x$ and $\hat{y}_0 = \hat{\theta}_y$, where $\hat{\boldsymbol{\theta}} = \left[\hat{\theta}_x, \hat{\theta}_y, \ldots\right]$.

3.4.4. Constraining parameters of PSF models

The Levenberg-Marquardt algorithm and the Nelder-Mead method used above search for values of the parameters $\boldsymbol{\theta}$ over an infinite interval. The optimization process can therefore converge to a solution with negative values which is impossible for variables corresponding to image intensity or to the standard deviation of a Gaussian PSF. We therefore limit the interval of possible values by transforming the relevant parameters and using $\mathrm{PSF}\left(x,y \mid \tilde{\boldsymbol{\theta}}\right)$ in Equations (26) and (29) instead of $\mathrm{PSF}\left(x,y \mid \boldsymbol{\theta}\right)$. The transformation for a 2D Gaussian PSF model is $\tilde{\boldsymbol{\theta}} = [\theta_x, \theta_y, \theta_\sigma^2, \theta_N^2, \theta_b^2]$. The optimization process is still unconstrained but will result in positive PSF parameters. Such a transformation also improves the stability of the fit.

3.4.5. Localization uncertainty

Let $\hat{\theta}_\sigma$ be the standard deviation of a fitted Gaussian PSF in nm, a is the backprojected pixel size in nm, $\hat{\theta}_N$ is estimate of the number of photons detected for a given molecule, and $\hat{b}$ is the background signal level in photons calculated as the standard deviation of the residuals between the raw data and the fitted PSF model. The uncertainty of least-squares or maximum-likelihood estimate of lateral position of a molecule is estimated as

$$\begin{aligned} (\Delta\hat{\theta}_{xy})^2\Big|_{\mathrm{LSQ}} &= \frac{g\hat{\theta}_{\sigma^2} + a^2/12}{\hat{\theta}_N}\left(\tfrac{16}{9} + 4\tau\right), \\ (\Delta\hat{\theta}_{xy})^2\Big|_{\mathrm{MLE}} &= \frac{g\hat{\theta}_{\sigma^2} + a^2/12}{\hat{\theta}_N}\left(1 + 4\tau + \sqrt{\tfrac{2\tau}{1+4\tau}}\right), \end{aligned} \tag{30}$$

respectively. Here

$$\tau = \frac{2\pi(\hat{b}^2 + r)(\hat{\theta}_\sigma^2 + a^2/12)}{a^2\hat{\theta}_N}.$$

The uncertainty for least-squares estimate is also known as the Thompson-Larson-Webb formula [20], which has been altered with the correction factor of $\frac{16}{9}$ as suggested by [19]. The uncertainty for maximum-likelihood was derived in [26]. Finally, the compensation for readout noise r and EM gain g has been added by following [27], who

suggested that when using EMCCD cameras, the correction factors should be set to $r = 0$, $g = 2$ and when using CCD or sCMOS cameras the readout noise in electron counts should be set to $r = g = 2$.

4. Sub-pixel 3D localization of molecules

ThunderSTORM supports 3D localization of molecules using the astigmatic imaging approach introduced by Huang et al. [28].

Input: A raw image I of size $m \times n$ pixels and a list of coordinates $\{(\tilde{x}_p, \tilde{y}_p)\}_{p=1}^{\tilde{P}}$ with approximate molecular positions.

Output: A list of coordinates $\{(\hat{x}_p, \hat{y}_p, \hat{z}_p)\}_{p=1}^{\hat{P}}$ with sub-pixel positions of the molecules.

4.1. The astigmatism method

3D SMLM imaging can be performed by introducing a weak cylindrical lens into the imaging path to create slight astigmatism in the image [28]. This results in images of molecules with different ellipticity depending on their axial position. When a molecule is in focus, its image appears round. If the molecule is slightly above or below the focal plane, its image appears ellipsoidal. Calibration of the imaging system is needed to determine the orientation of the imaged ellipsoid (the camera chip might not be aligned with cylindrical lens) and the relationships between the axial position and ellipticity of the imaged molecules.

4.1.1. PSF model

Rotated elliptical Gaussian function

A common PSF model for astigmatic 3D imaging is a rotated, elliptical Gaussian function given by the formula

$$\mathrm{PSF}_{\mathrm{EG}}\left(x, y \mid \boldsymbol{\theta}, \phi\right) = \frac{\theta_N}{2\pi\sigma_1\left(\theta_z\right)\sigma_2\left(\theta_z\right)} \exp\left(-\frac{x'^2}{2\left(\sigma_1\left(\theta_z\right)\right)^2} - \frac{y'^2}{2\left(\sigma_2\left(\theta_z\right)\right)^2}\right) + \theta_b\,, \quad (31)$$

where $\mathrm{PSF}_{\mathrm{EG}}\left(x, y \mid \boldsymbol{\theta}, \phi\right)$ gives the expected photon count at the integer pixel position (x, y) for the parameters $\boldsymbol{\theta} = [\theta_x, \theta_y, \theta_z, \theta_N, \theta_b]$, and

$$\begin{aligned} x' &= (x - \theta_x)\cos\phi - (y - \theta_y)\sin\phi\,, \\ y' &= (x - \theta_x)\sin\phi + (y - \theta_y)\cos\phi\,. \end{aligned}$$

The entries of the vector $\boldsymbol{\theta}$ are as follows: $\theta_x, \theta_y, \theta_z$ are the sub-pixel molecular coordinates, $\sigma_1\left(\theta_z\right)$ and $\sigma_2\left(\theta_z\right)$ are the imaged widths of the molecule along two perpendicular axes rotated by the angle ϕ with respect to xy coordinates, θ_N corresponds to the total number of photons emitted by the molecule, and θ_b is the background offset. The imaged widths $\sigma_1\left(\theta_z\right)$ and $\sigma_2\left(\theta_z\right)$ are modeled by a pair of defocusing curves described in Section 4.1.2.

4.1.2. Defocusing models

ThunderSTORM provides two models to describe the relationship between the axial position of a molecule and its imaged size. The first model is represented by a pair of second degree polynomials, the second model is introduced by Huang et al. [28].

Polynomial defocusing model

The default model used in ThunderSTORM describes the relationship between the axial position of a molecule and its imaged widths along two perpendicular axes by a pair of second degree polynomials

$$\sigma_1(z) = a_1(z-c_1)^2 + b_1, \tag{32}$$
$$\sigma_2(z) = a_2(z-c_2)^2 + b_2. \tag{33}$$

Here a_1, b_1, c_1 and a_2, b_2, c_2 are unknown parameters of the model which can be determined by the calibration process described in Section 4.1.3.

Huang's defocusing model

An alternative model of the defocusing curves is described by Huang et al. [28]. The relationship between the axial position of a molecule and its imaged widths along two perpendicular axes is given by

$$\sigma_1(z) = \frac{w_{01}}{2}\sqrt{1 + \left(\frac{z-c_1}{d_1}\right) + a_1\left(\frac{z-c_1}{d_1}\right)^2 + b_1\left(\frac{z-c_1}{d_1}\right)^4}, \tag{34}$$
$$\sigma_2(z) = \frac{w_{02}}{2}\sqrt{1 + \left(\frac{z-c_2}{d_2}\right) + a_2\left(\frac{z-c_2}{d_2}\right)^2 + b_2\left(\frac{z-c_2}{d_2}\right)^4}. \tag{35}$$

Here $w_{01}, a_1, b_1, c_1, d_1$ and $w_{02}, a_2, b_2, c_2, d_2$ are unknown parameters of the model which can be determined by the calibration process described in Section 4.1.3.

4.1.3. Calibration of the imaging system

Calibration is a procedure which determines the orientation angle ϕ of the imaged ellipsoids, and the relationship between the axial position z of the molecules and their imaged widths $\sigma_1(z), \sigma_2(z)$. The calibration is typically performed using a Z-stack of images of sub-diffraction fluorescent beads. We use a sparse sample with about 10 to 50 beads in the image and a Z-stack image sequence with an axial range of about 2 μm and a step size of 10 nm.

Determining the orientation angle

1. A sequence of images from a Z-stack is processed slice-by-slice using the methods described in Sections 1, 2, and 3.4. Images of the beads are fit independently using the elliptical Gaussian PSF given by Equation (31) with ϕ as a free parameter.

2. Results close to circular are discarded as the angle ϕ cannot be determined.

3. The final orientation angle is calculated as the circular mean of all remaining measurements

$$\phi = \frac{1}{4}\operatorname{atan2}\left(\frac{1}{n}\sum_{i=1}^{n}\sin\varphi_i,\ \frac{1}{n}\sum_{i=1}^{n}\cos\varphi_i\right), \tag{36}$$

where $\varphi_i = 4\left(\phi_i \bmod \frac{\pi}{2}\right)$ adjusts the fitted angles ϕ_i, and n is the number of measured beads.

Ellipticity as a function of an axial position

1. Using the approximate positions of the beads in multiple Z-planes and the orientation angle ϕ, both determined in the previous step, the images of the beads are fit again using the elliptical Gaussian PSF given by Equation (31), but with a fixed angle ϕ.

2. To estimate the coefficients for the defocusing model, see Section 4.1.2, we first fit the pair of defocusing curves for each bead separately using an iterative least-squares algorithm which automatically discards outliers.

3. From the fitted models, we determine a common focal plane of the beads as $\frac{c_1+c_2}{2}$ and shift the data along the z-axis such that all beads are positioned at the same focal plane.

4. The final coefficients are obtained by fitting the pair of defocusing curves to all shifted data points. The "zero" axial position is given by the intersection of the two polynomials.

4.1.4. Localization uncertainty

The lateral uncertainty is calculated same as in the 30, but τ differs because of the axial defocus (PSF spreads and is never focused in both planes simultaneously, thus the uncertainty is worse). This has been derived in [26] as

$$\tau = \frac{2\pi(\hat{b}^2 + r)(\hat{\theta}_{\sigma_1}\hat{\theta}_{\sigma_2}(1 + l^2/d^2) + a^2/12)}{a^2\hat{\theta}_N} .$$

Since the axial position is estimated from $\hat{\theta}_{\sigma_1}$ and $\hat{\theta}_{\sigma_2}$, the axial uncertainty is calculated from uncertainty of these parameters

$$\begin{aligned} (\Delta\hat{\theta}_{\sigma_j})^2\Big|_{\text{LSQ}} &= \frac{g\hat{\theta}_{\sigma_j^2}+a^2/12}{\hat{\theta}_N}(1+8\tau) , \\ (\Delta\hat{\theta}_{\sigma_j})^2\Big|_{\text{MLE}} &= \frac{g\hat{\theta}_{\sigma_j^2}+a^2/12}{\hat{\theta}_N}\left(1+8\tau+\sqrt{\frac{9\tau}{1+4\tau}}\right) , \end{aligned} \tag{37}$$

where j can be substituted to calculate uncertainty of $\hat{\theta}_{\sigma_1}$ and $\hat{\theta}_{\sigma_2}$. Then from error propagation follows

$$F^2 = \frac{4l^2\hat{\theta}_z^2}{(l^2 + d^2 + \hat{\theta}_z^2)^2} , \tag{38}$$

$$(\Delta F)^2 = (1 - F^2)\left[\left(\frac{\Delta\hat{\theta}_{\sigma_1}}{\hat{\theta}_{\sigma_1}}\right)^2 + \left(\frac{\Delta\hat{\theta}_{\sigma_2}}{\hat{\theta}_{\sigma_2}}\right)^2\right] , \tag{39}$$

$$(\Delta\hat{\theta}_z)^2 = (\Delta F)^2\frac{(l^2 + d^2 + \hat{\theta}_z^2)^4}{4l^2(l^2 + d^2 - \hat{\theta}_z^2)^2} , \tag{40}$$

where $2l$ is the distance between focal planes given by the astigmatic lens and the geometry of the setup and d is a measure of focal depth. These quantities are already known during the 3D fitting process as $l^2 \propto c_1 c_2$ and $d^2 \propto d_1 d_2$, where c_1, c_2, d_1, d_2 are parameters of the defocus curves as in Equations 34, 35.

5. The Crowded-field problem

High spatial densities of activated molecules can result in a "crowded field problem," in which single molecules are not adequately resolved. The reason for this is the limitation of the PSF models described above to fit only a single molecule. To help solve this problem, ThunderSTORM uses a multiple-emitter fitting analysis (MFA) approach similar to the algorithm described in [11].

5.1. Multiple-emitter fitting analysis

The multiple-emitter fitting analysis approach uses a PSF defined by a model

$$\mathrm{PSF}_N\left(x, y \mid \boldsymbol{\Theta}\right) = \sum_{i=1}^{N} \mathrm{PSF}\left(x, y \mid \boldsymbol{\theta}_i\right) , \tag{41}$$

where N is a number of molecules allowed in the fitting region, and $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N]$ are parameters describing position and shape of the imaged molecules modeled by the PSF.

The fitting of multiple-emitter models to the raw data proceeds according to the following algorithm. First, the algorithm fits PSF_1 (a single molecule model) with an initial molecular position obtained by one of the methods described in Section 2. The fitted PSF is subtracted from the raw data and the position of the maximum intensity value in the residual image is taken as an approximate position of a second molecule. The fitting is now repeated on the raw data with PSF_2 (a model containing two molecules) and with the initial positions estimated in the previous steps. The result of the fit is subtracted from the raw data to find an approximate position of a third molecule in the residual image. This routine is repeated until the maximum number of molecules allowed in the fitting region is reached. Note that initial positions of the molecules are further adjusted, during the multiple-emitter fitting analysis, by a "Push&Pull" process [11]. To find the optimal number of molecules, statistical tests are required, see Section 5.2.

Users can specify the size of the fitting region, the maximum number of molecules allowed in one fitting region, the type of PSF (models for obtaining both 2D or 3D coordinates are described in Sections 3.4.1 and 4.1.1), and the fitting method (least-squares methods or maximum likelihood estimation, see Sections 3.4.2 and 3.4.3). Optionally, users can constrain the multiple-emitter fitting algorithm such that all fitted molecules have the same intensity or an intensity in a given range. The background offset is constrained to the same intensity for all fitted molecules.

5.2. Model selection

Because a model with more parameters will always be able to fit the data at least as well as a model with fewer parameters [18, 23], statistical tests are required to determine whether the more complex model provides a significantly better fit of the underlying data. Statistical tests are usually based on pair-wise model comparison. Here a fit by PSF_1 is compared with a fit by PSF_2, the better of the two is compared with a fit by

PSF_3, etc. Pair-wise comparisons are based on an F-test [18, 23] or on a log-likelihood ratio test [18, 11] as described below.

5.2.1. F-test

An F-test [18, 23] arises in the case of fitting by least squares methods, see Section 3.4.2, when we need to compare significance of the fit between two models, where one model (the null model) is a special case of the other (the alternative model) for some choice of parameters. The F-test statistic computed from the data is given by the formula

$$F_{\chi^2}(\mathcal{D}) = \frac{\left(\chi^2(\boldsymbol{\Theta}_0 \mid \mathcal{D}) - \chi^2(\boldsymbol{\Theta}_1 \mid \mathcal{D})\right) / (v_1 - v_0)}{\chi^2(\boldsymbol{\Theta}_1 \mid \mathcal{D}) / (n - v_1)}, \tag{42}$$

where the sum of squared residuals $\chi^2(\boldsymbol{\Theta} \mid \mathcal{D})$ computed for a model with parameters $\boldsymbol{\Theta}$ is defined by Equation (25), vectors $\boldsymbol{\Theta}_0$ and $\boldsymbol{\Theta}_1$ are parameters of the null and alternative model, respectively, v_0 and v_1 (where $v_0 < v_1$) represent the number of free parameters of the null and alternative model, respectively, and $n = l^2$ is the number of data points within the fitting region $\mathcal{D}$, see Section 3.

Assuming the null hypothesis that the alternative model does not provide a significantly better fit than the null model, the F-test statistics computed in Equation (42) has an F-distribution with $(v_1 - v_0, n - v_1)$ degrees of freedom. The null hypothesis is rejected if $F_{\chi^2}(\mathcal{D})$ computed from the data is greater than the critical value of the $F_{v_1 - v_0, n - v_1}$ distribution for a user-specified p-value.

5.2.2. Log-likelihood ratio test

To compare between the fits of two models, in the case of fitting by the maximum likelihood estimation method described in Section 3.4.3, we use a model selection criteria based on a log-likelihood ratio test [18, 11]. Assuming that one model (the null model) is a special case of the other (the alternative model) for some choice of parameters, the log-likelihood ratio is given by the formula

$$\Lambda(\mathcal{D}) = -2 \ln \left[\frac{L(\boldsymbol{\Theta}_0 \mid \mathcal{D})}{L(\boldsymbol{\Theta}_1 \mid \mathcal{D})} \right], \tag{43}$$

where the likelihood $L(\boldsymbol{\Theta} \mid \mathcal{D})$ of parameters $\boldsymbol{\Theta}$ is defined by Equation (29), $\boldsymbol{\Theta}_0$ and $\boldsymbol{\Theta}_1$ are the parameters of the null and alternative model, respectively, and $\mathcal{D}$ is a fitting region defined in Section 3.

The probability distribution of the log-likelihood ratio computed in Equation (43), assuming the null hypothesis that the alternative model does not provide a significantly better fit than the null model, can be approximated by the χ^2 distribution with $v_1 - v_0$ degrees of freedom. This approximation is usually valid even for small sample sizes [18]. Here v_0 and v_1 (where $v_0 < v_1$) represent the number of free parameters of the null and alternative models, respectively. The null hypothesis is rejected if the log-likelihood ratio $\Lambda(\mathcal{D})$ computed from the data is greater than the critical value of the $\chi^2_{v_1 - v_0}$ distribution for some p-value specified by the users.

Part II.
POST-PROCESSING

6. Post-processing analysis

ThunderSTORM provides a variety of post-processing methods and tools which can be applied to the results obtained from previously processed SMLM data. The software includes methods for correcting the data by removing molecules with, e.g., poor localization accuracy or small local density, for merging molecules present in subsequent images, and for position correction of molecules caused by lateral drift of the sample and/or by axial stage scanning. Users can also define the order in which the different post-processing steps are applied. Localization results can be imported/exported to/from ThunderSTORM in a variety of data formats which allows full compatibility with other SMLM localization software.

Input: A list of coordinates with sub-pixel positions of the molecules.

Output: A corrected list of coordinates with sub-pixel positions of the molecules.

6.1. Removing molecules with poor localization

ThunderSTORM can remove unwanted molecules according to user-defined criteria, or display molecules in a user-specified region. The filtering criteria is formulated as an expression combining mathematical and logical functions and operators with parameters from the table of results obtained from previous data analysis. All molecules with parameters not accepted by the criteria are removed. Note that users can also display histograms of particular measured parameters and select thresholds to define the filtering criteria.

6.2. Local density filtering

Local density filtering effectively removes noise created by, e.g., isolated localizations. The method works by calculating the number of neighbors at a given radius (in 2D or 3D) for every localization event. Localization event is discarded if the number of neighbors is below a user-specified threshold.

6.3. Merging of reappearing molecules

In some SMLM experiments, a single photoactivated molecule may appear in several sequential images, then disappear for a while, appear again, and finally bleach completely. If this is the case, ThunderSTORM identifies such sequences of molecular locations and combines them into one single molecule. The new position, imaged size, and the background offset of the resulting signal is calculated as the mean value of the original data,

while the intensity and the background signal level (standard deviation of the residuals between the raw data and the fitted PSF model, see Section 3.4.5) is calculated as the sum. The localization uncertainty of the merged molecule is calculated from the new values. The distance within which molecules are merged together in the subsequent frames, as well as the allowed number of frames in which the molecule can disappear, are user-specified values. Users can also specify a maximum number of consecutive frames such that a repeating event is still considered a single molecule, see the timing diagram in Figure 3. The merging algorithm is based on finding nearest-neighbors between active molecules from the previous frame and the current frame.

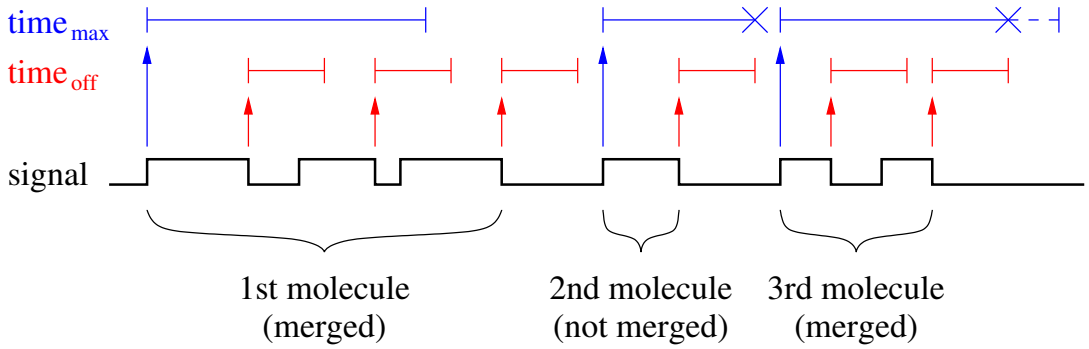


Figure 3: Timing diagram of the merging algorithm.

6.4. Removing duplicates

Repeated localizations of single molecules in one frame may occur due to overlapping fitting sub-regions when using the multiple-emitter analysis approach. To deal with the problem, ThunderSTORM uses a post-processing method suggested by Huang et al. [11]. Here molecules with a mutual distance smaller than their localization uncertainty are grouped together, and in each group, only the molecule with the smallest localization uncertainty is kept. The other molecules are rejected. ThunderSTORM generalizes this algorithm by allowing users to specify the distance threshold for grouping the molecules as a mathematical expression. A different distance threshold can be applied to individual molecules if the mathematical expression results in a vector. The same distance threshold is applied to all molecules if the result is a scalar.

6.5. Lateral drift correction

Long time-series data acquisitions usually suffer from sample drift. ThunderSTORM supports two methods for lateral drift correction. The first is based on tracking fiducial markers inserted into the sample, and the second on cross-correlation of similar structures in reconstructed super-resolution images. The trajectory of the relative sample drift can be saved to a file and applied later, possibly to a different dataset. For example, drift

estimated from a sub-region of the data can be applied to the whole dataset, or drift estimated from one channel can be applied to correct drift in another channel.

6.5.1. Fiducial markers

A common approach for correcting drift is performed by tracking fiducial markers present in the sample and then subtracting their relative motion from the molecular localizations. ThunderSTORM can identify fiducial markers automatically from the localization results as molecules that stay in the "on" state at one position for a substantial amount of time. Therefore, all localizations that arise from more than a user-specified number of frames are considered as fiducial markers and are used for the drift correction. Assigning molecular localizations in subsequent frames to a single fiducial marker is performed by the merging algorithm described in Section 6.3.

The sample drift is obtained by averaging the relative trajectories of all identified fiducial markers into a single trajectory. The sample drift at each frame $t = 1, \ldots, T$, is computed (for x and y) according to the formula

$$\overline{x}_t = \frac{1}{M} \sum_{i=1}^{M} (x_{i,t} - \theta_i) \ . \tag{44}$$

Here M is the number of fiducial markers, $x_{i,t}$ is the absolute position of the i-th marker at frame t, $i = 1, \ldots, M$, and θ_i is an unknown offset which has to be subtracted from the absolute marker position to obtain the relative position.

The offset θ_i is estimated by least squares minimization of the sum of squared differences between the relative marker positions and the relative sample drift, summed over all markers and frames. The optimization is defined by the formula

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}=[\theta_1,\ldots,\theta_M]}{\arg\min} \sum_{t=1}^{T} \sum_{i=1}^{M} \left((x_{i,t} - \theta_i) - \overline{x}_t\right)^2 \ , \tag{45}$$

where $\hat{\boldsymbol{\theta}} = \left[\hat{\theta}_1, \ldots, \hat{\theta}_M\right]$ are the values of the estimated offset for each of the markers.

In reality, some of the points $x_{i,t}$ may be missing because the markers might not be localized in some frames. If this is the case, the relative sample drift in Equation (44) is computed only from the corresponding number of fiducial markers. For the missing markers, the corresponding sum of squared differences in Equation (45) is set to zero.

The final drift trajectory is smoothed by robust locally weighted regression [29]. Users can specify the maximum merging distance and the minimum number of frames in which a molecule must appear to be considered as a fiducial marker, and the trajectory smoothing factor.

Note that analyzing samples with fiducial markers yields localizations of both the blinking fluorophores and the fiducial markers. This may slow down the merging algorithm used for automatic identification of the markers. For faster marker identification, the merging process can be limited to regions containing only the fiducial markers. The drift trajectory can then be saved to a file and applied later to the whole dataset.

6.5.2. Cross-correlation

ThunderSTORM also supports lateral drift correction using the method of Mlodzianoski et al. [30]. Here, the list of localized molecules is split into $n + 1$ batches based on the frame in which they appeared. Each batch is used to create one super-resolution image. The presumption of this method is that similar structures will appear in all reconstructed images. Cross-correlation methods are used to determine the shift between the first image and each of the subsequent images. This leads to n cross-correlation images, where the shift in the position caused by the drift corresponds to the relative position between the global intensity maximum peaks. The localized peaks are assigned to the central frame of each batch sequence and the drift for intermediate frames is determined by local regression using third degree polynomials. The original molecular coordinates are corrected for drift using the estimated values.

In our implementation, super-resolution images are created by the average shifted histograms method described in Section 8.3, cross-correlation images are computed by Fast Fourier Transform methods as implemented in ImageJ, and the location of global intensity maximum peaks is determined with sub-pixel precision using the radial symmetry method described in Section 3.3. The number of batches n and the magnification of super-resolution images is defined by users. For better stability of the solution, intensity maximum peaks are first localized in cross-correlation images computed from reconstructed images with a magnification of one. The peak position is refined afterwards using cross-correlation images computed from super-resolution images with a user specified magnification. The final position of the peak is obtained as a local intensity maximum in close proximity to the peak obtained at lower magnification.

6.6. Z-stage scanning

To extend the axial field of view in 3D SMLM experiments, data can be acquired in multiple z-stage positions [31]. However, the astigmatism method described in Section 4.1 can estimate only the relative axial positions of molecules with respect to the focal plane. To find the absolute z coordinates, the correct offset must be added to the axial positions of each molecule. The new z coordinate can be determined as

$$z_{\text{abs}} = z_0 + z_{\text{rel}} + z_{\text{step}} \left(\left\lfloor \frac{f-1}{n_{\text{im}}} \right\rfloor \bmod n_{\text{pos}} \right) , \tag{46}$$

where z_0 is the initial z-stage offset, z_{rel} is the estimated molecular position relative to the focal plane ($z_{\text{rel}} = 0$ for 2D data), z_{step} is the step between two positions of the Z-stage, f is the image frame in which a given molecule appeared, n_{im} is the number of images taken in each z-stage position, and n_{pos} is the number of z-stage positions. All z variables are in nm.

7. Co-localization

Co-localization is used to measure spatial overlap between two (or more) different fluorescent labels, each having a separate emission wavelength. ThunderSTORM provides a module for analysis of dual-color super-resolution images which applies coordinate based co-localization [32].

Input: Two list of coordinates with sub-pixel positions of localized molecules.

Output: CBC value for every molecule.

7.1. Coordinate based co-localization

Traditional co-localization analysis of conventional, dual-color fluorescence microscopy images suffers from a limited spatial resolution and chromatic errors [33]. Coordinate based co-localization (CBC) of dual-color super-resolution images provides much better approach as it allows to determine co-localization on a molecular level, it overcomes the dynamic range of a camera, and it is not sensitive to cross talks [32].

Calculation of the CBC value around a given molecule A_i, according to [32], starts by determining two distributions of distances

$$D_{A_i,A}(r) = \frac{N_{A_i,A}(r)}{N_{A_i,A}(R_{\max})} \frac{R_{\max}^2}{r^2}, \tag{47}$$

$$D_{A_i,B}(r) = \frac{N_{A_i,B}(r)}{N_{A_i,B}(R_{\max})} \frac{R_{\max}^2}{r^2}. \tag{48}$$

Here $N_{A_i,A}(r)$, resp. $N_{A_i,B}(r)$, is the number of localized molecules in channel A, resp. B, within the distance $r = 0, \ldots, R_{\max}$ around A_i. These distributions are corrected for the area given by r and normalized by the number of localizations within the largest observed distance $R_{\max}$.

Having these two distributions of distances, Sperman's rank correlation coefficient $S_{A_i} = \langle D_{A_i,A}, D_{A_i,B} \rangle$ is calculated. The co-localization value is determined for every single molecule according to

$$C_{A_i} = S_{A_i} \exp\left(-\frac{E_{A_i,B}}{R_{\max}}\right), \tag{49}$$

where $E_{A_i,B}$ is a distance from localization A_i to the nearest neighbor localization in channel B.

In ThunderSTORM, the input data for channels A and B are provided as a table of results and a ground-truth table, respectively. Computed values with the co-localization coefficient C_{A_i}, with the distance to the nearest neighbor $E_{A_i,B}$, and with the number of neighbors $N_{A_i,B}(r)$ within the radius $r = 0, \ldots, R_{\max}$, are displayed in the table of results as new columns.

8. Visualization methods

Visualization (rendering) of SMLM data involves creation of a new super-resolution image based on the coordinates of the localized molecules. ThunderSTORM offers several methods, all of which support both two-dimensional visualization and slice-by-slice three-dimensional visualization. In the three-dimensional case, each image slice contains a visualization of molecules with axial positions in the user-specified range. The desired magnification ratio of the new super-resolution image to the original image is user-specified.

Input: A list of coordinates with sub-pixel positions of the molecules.

Output: A super-resolution image with user-specified size and magnification.

8.1. Scatter plot

Scatter plot visualization [34] is the simplest method, and does not usually provide high quality results. A simple binary image is created with pixel intensity values set to one at locations corresponding to molecular positions. All other pixel intensity values are set to zero. This method is fast but does not reflect the density of molecules.

8.2. Histogram

Histograms are used to estimate the density of data by counting the number of observations that fall into each of the bins. In our case, a two-dimensional histogram of molecular positions is created with the bin size corresponding to the pixel size of the final super-resolution image [34]. Thus, for every localized molecule, the bin value (i.e., the image brightness) at the corresponding molecular positions is incremented by one.

The histogram visualization optionally supports "jittering" [8]. When enabled, a random number drawn from the normal distribution, with a standard deviation equal to the computed (or user-specified) localization uncertainty, is added to the coordinates of every molecular position before creating the histogram. This step is applied multiple times and all generated histograms are averaged together. As the number of jitters increases, the final image approaches the result of the Gaussian rendering described in Section 8.4. For a small number of jitters, the histogram visualization is much faster than the Gaussian rendering but the resulting images may appear noisy.

8.3. Averaged shifted histograms

This visualization algorithm uses a density estimation approach based on averaged shifted histograms [35]. In the one-dimensional case, this method works by averaging n histograms with the same bin width w, but with the origin of each histogram shifted by $\frac{w}{n}$ from the previous histogram. In the multidimensional case, there are n^d multidimensional histograms averaged in total, i.e., for n shifts in each of the d dimensions. In our implementation, the width of the histogram bin is determined as $w = na$,

where a is the pixel size of the super-resolution image. The number of shifts n in the lateral and axial directions can be specified independently.

8.4. Gaussian rendering

This method draws a normalized symmetric 2D or 3D Gaussian function integrated over the voxel volume for every localized molecule, with a standard deviation equal to the computed, or user-specified localization uncertainty. The visualized molecules are added sequentially to the final super-resolution images. The contribution of one molecule to the voxel intensity at the integer position (x, y, z) is computed as

$$v\left(x, y, z \mid \boldsymbol{\theta}_p\right) = E_x E_y E_z \,, \tag{50}$$

where p indexes the molecules, and the parameters $\boldsymbol{\theta}_p = [\hat{x}_p, \hat{y}_p, \hat{z}_p, \hat{\sigma}_{xy,p}, \hat{\sigma}_{z,p}]$. Here $\hat{x}_p, \hat{y}_p, \hat{z}_p$ is the estimated position of a molecule, $\hat{\sigma}_{xy,p}$ is the corresponding lateral localization uncertainty, $\hat{\sigma}_{z,p}$ is the axial localization uncertainty,

$$E_x = \frac{1}{2}\operatorname{erf}\left(\frac{x - \hat{x} + \frac{1}{2}}{\sqrt{2}\hat{\sigma}_{xy}}\right) - \frac{1}{2}\operatorname{erf}\left(\frac{x - \hat{x} - \frac{1}{2}}{\sqrt{2}\hat{\sigma}_{xy}}\right), \tag{51}$$

$$E_y = \frac{1}{2}\operatorname{erf}\left(\frac{y - \hat{y} + \frac{1}{2}}{\sqrt{2}\hat{\sigma}_{xy}}\right) - \frac{1}{2}\operatorname{erf}\left(\frac{y - \hat{y} - \frac{1}{2}}{\sqrt{2}\hat{\sigma}_{xy}}\right), \tag{52}$$

$$E_z = \frac{1}{2}\operatorname{erf}\left(\frac{z - \hat{z} + \frac{\Delta_z}{2}}{\sqrt{2}\hat{\sigma}_z}\right) - \frac{1}{2}\operatorname{erf}\left(\frac{z - \hat{z} - \frac{\Delta_z}{2}}{\sqrt{2}\hat{\sigma}_z}\right), \tag{53}$$

and $\triangle_z$ is the size of a voxel in the axial direction. Contributions from one molecule are limited to an interval given by a circle with radius of $3\hat{\sigma}_{xy,p}$ around the molecule position in lateral dimension and by $3\hat{\sigma}_{z,p}$ in axial direction. For data visualization in the 2D case, $z = 0$ and the term $E_z = 1$.

8.5. Comparison of visualization methods

All visualization methods described above, except for the scatter plot, result in molecular density maps. Their time complexity is $O\left(N\right)$, where N is the number of molecules to visualize. However, the real speed of visualization depends on the number of histograms to average in the case of histogram based methods, or on the volume to render in the case of Gaussian rendering. The Gaussian rendering approach is slow but it can show the calculated localization accuracy for each molecule. The method based on averaging histograms with jittered molecular positions approximates the Gaussian rendering approach for a large number of jitters. This method is fast for small number of jitters but the resulting images appear very noisy. The proposed method based on the average shifted histogram approach provides similar results as Gaussian rendering with a constant standard deviation, therefore the result does not show the localization uncertainty for each molecule. However, the average shifted histogram approach is orders of magnitude faster than Gaussian rendering. For more information about other visualization methods see Baddeley et al. [34].

The super-resolution SMLM images shown in Figures 4 and 5 were created from simulated data generated by ThunderSTORM, and from real SMLM data publicly available from the Single-Molecule Localization Microscopy challenge website [36], respectively.

Simulated data were created using the module "Generator of simulated data" within ThunderSTORM, see Section 9. The raw image sequence consists of 1000 images of size 256×256 pixels with a pixel size of 80 nm/pixel. The spatial density was set to 0.5 molecules/μm^2 and was modulated by a Siemens-star pattern (available on the ThunderSTORM website). The Siemens star pattern is commonly used to test the resolution of optical instruments. Other camera and data generator settings were kept at default values. About eighty thousand molecules were generated and localized. The results of different visualization methods with different parameter settings are shown in Figure 4.

For real data, we used the image sequence "Tubulin AF647" [36]. The raw image sequence consists of 9990 frames of size 128×128 pixels with a pixel size of 100 nm/pixel. Data were processed by ThunderSTORM with the default settings. About sixty thousand molecules were localized and drift correction was applied. Super-resolution images visualized by different methods are shown in Figure 5.

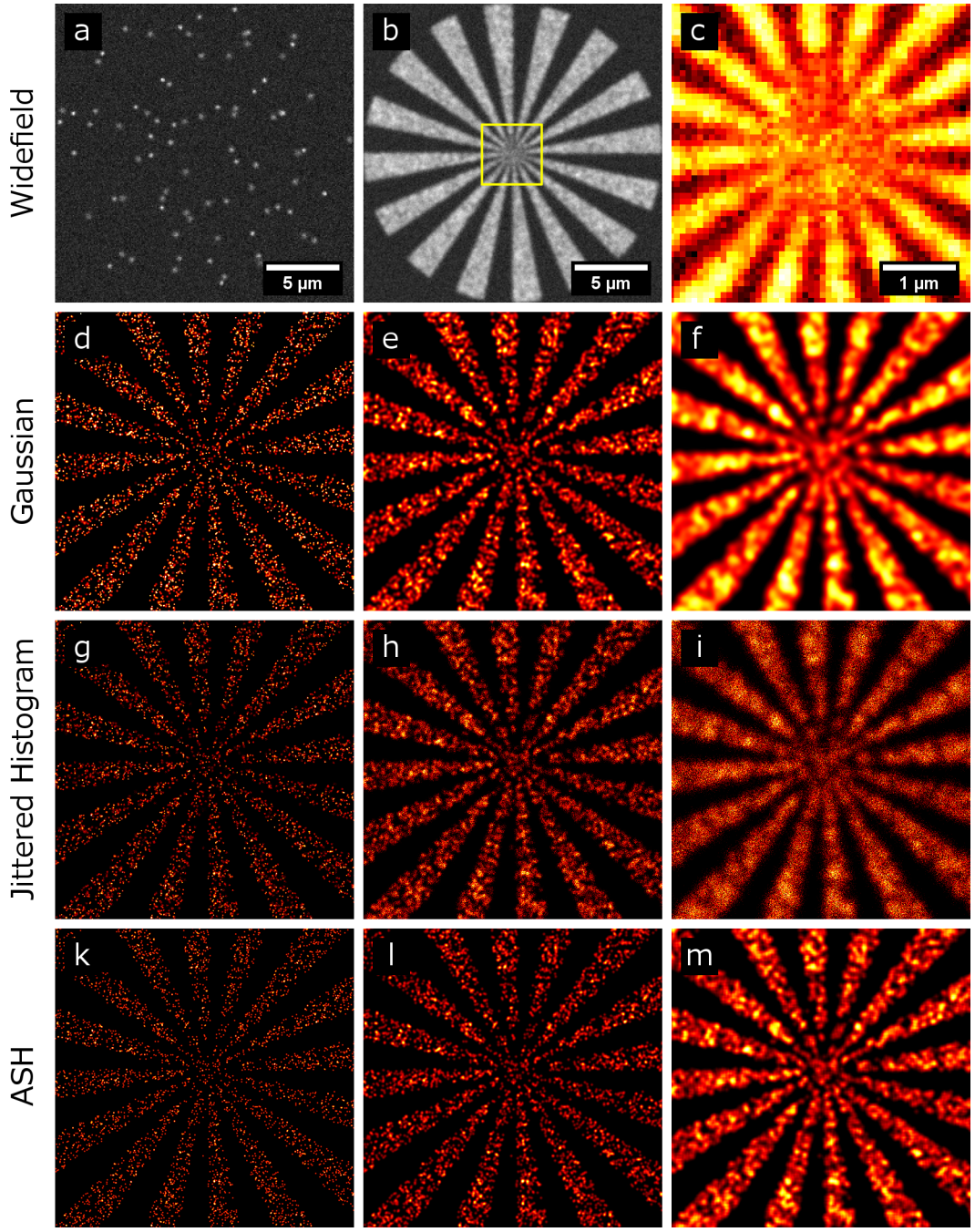


Figure 4: Comparison of visualization methods on simulated data. Widefield images: a) a single frame of raw data, b) average intensity time projection, c) zoomed region indicated in b). d-m) Super-resolution SMLM images of the region indicated in b) magnified 10×. Gaussian rendering with lateral uncertainty set to: d) computed localization uncertainty, e) 20 nm, f) 50 nm. Jittered Histogram with number of averages set to 100 and with lateral uncertainty set to: g) computed localization uncertainty, h) 20 nm, i) 50 nm. Average Shifted Histogram with the number of lateral shifts set to: k) 2, l) 4, m) 8.

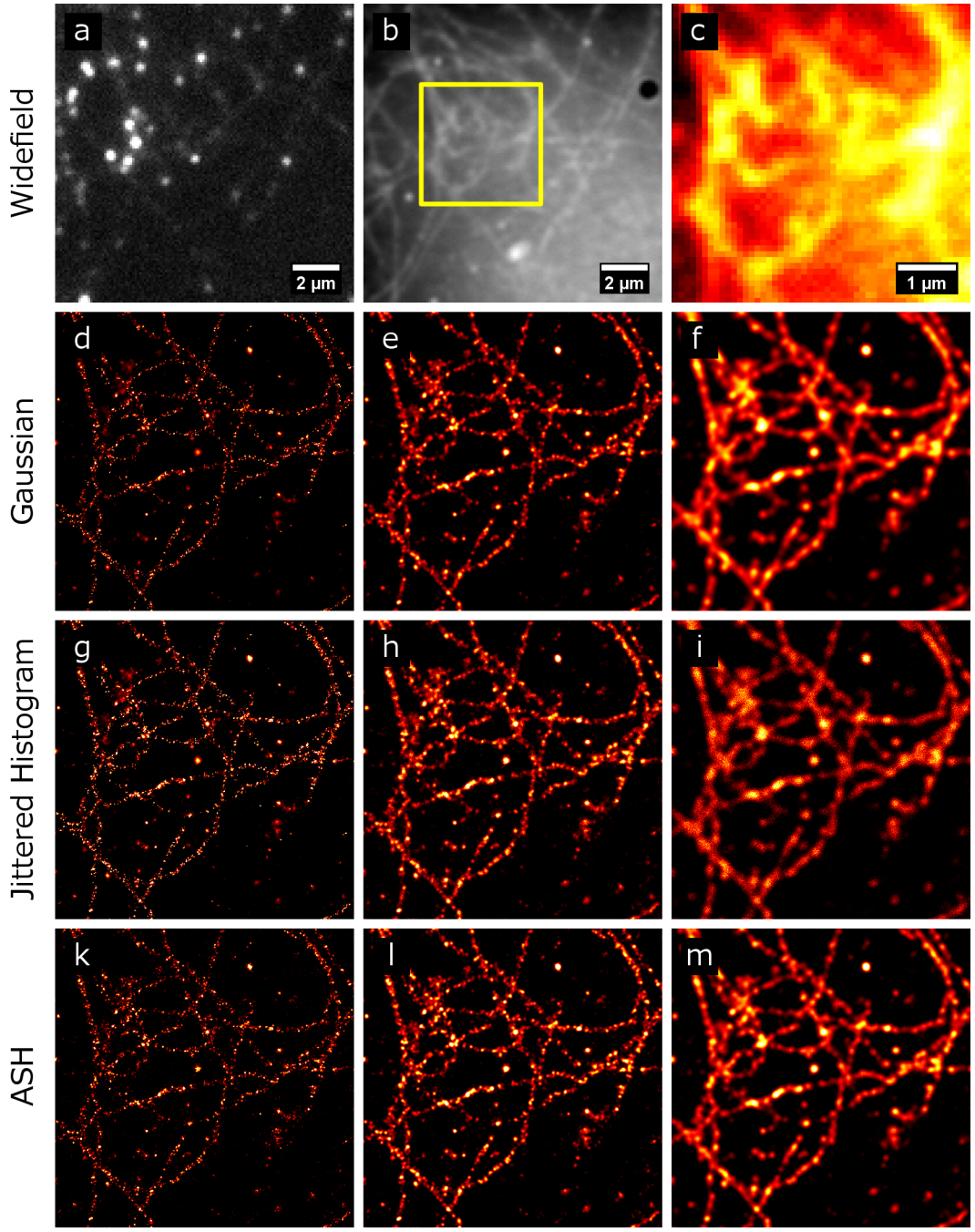


Figure 5: Comparison of visualization methods on Tubulin AF647 [36]. Widefield images: a) a single frame of raw data, b) average intensity time projection, c) zoomed region indicated in b). d-m) Super-resolution SMLM images of the region indicated in b) magnified 10×. Gaussian rendering with lateral uncertainty set to: d) computed localization uncertainty, e) 20 nm, f) 50 nm. Jittered Histogram with number of averages set to 100 and with lateral uncertainty set to: g) computed localization uncertainty, h) 20 nm, i) 50 nm. Average Shifted Histogram with the number of lateral shifts set to: k) 2, l) 4, m) 8.

Part III.
SIMULATION ENGINE

9. Generating simulated data

ThunderSTORM is capable of generating a sequence of SMLM-like images in which the ground-truth positions of the molecules are known. This allows users to perform Monte Carlo simulations [7, 8] and to quantitatively evaluate the performance of applied localization algorithms by calculating, e.g., the Jaccard index or F_1 score, see Section 10. In addition to the image size and sequence length, users can specify the intensity, imaged size, and spatial density of the generated molecules. The resulting images can be subjected to sample drift. Noise in the generated images simulates the behavior of CCD or EMCCD cameras.

9.1. Image formation

For each frame, we first create an ideal, noise free, SMLM-like image to simulate the expected number of photons detected in each camera pixel. Image formation starts by creating a list of molecules with FWHM and intensity chosen randomly in user-specified ranges, and with random positions of molecules given by a user-specified spatial density, see Section 9.2. Users can also specify any of the implemented PSF models, see Section 3.4.1, including 3D models described in Section 4, to create the simulated images of molecules. The generated molecules are added sequentially to the final image similarly as in the Gaussian rendering method described in Section 8.4. A user-specified offset is added to the generated image sequence to simulate photon background. Alternatively, a gray-scale image, in which each pixel value is normalized to the interval $[0, 1]$, can be used as a weighting factor of the offset level in different parts of the generated images to simulate an irregular background as might be encountered in real samples.

In order to simulate the photon counting process in the generated images, each pixel value expressed in photons is modified by a Poisson-distributed random number. The data generator can optionally simulate EM gain of EMCCD cameras. In this case we use a stochastic model described in [37], where the electron multiplication is modeled by the Gamma distribution $\Gamma(k, g)$. The shape k is given by the number of photons detected in the simulated CCD register and the scale g is given by the user-specified value of gain.

Finally, the signal in the camera register is digitized by converting the photons to digital counts. The CCD sensitivity (in photons per A/D count), and the camera digitizer offset (in A/D counts) are user-specified in the camera setup, as well as the camera pixel size (in nanometers) as projected to the sample plane.

9.2. Fixed or spatially varying density of molecules

Users can specify a fixed or spatially varying density of simulated molecules in the generated images. The variability is achieved by a user-supplied gray-scale mask M,

in which each pixel value is normalized to the interval $[0, 1]$ and used to represent the weighting factor, see Figure 6. The average number of molecules at a given integer pixel position (x, y) is then computed as

$$\overline{d}\,(x, y) = a^2 M\,(x, y)\, d_{\max}\,, \tag{54}$$

where a is the camera pixel size as projected to the sample plane, and $d_{\max}$ is the maximum spatial density of molecules per unit area as specified by users.

The procedure for generating molecular positions follows the spatial Poisson point process [18]. Thus for each value $\overline{d}\,(x, y)$, a random number of events (molecules) is created in the pixel (x, y) using a Poisson random number generator. Molecular centers are placed uniformly and randomly within that pixel.

Note that the mask M should be at least the same size as the desired super-resolution image in order to preserve high resolution in the final reconstruction. The coordinates of the molecular centers are down-scaled appropriately.

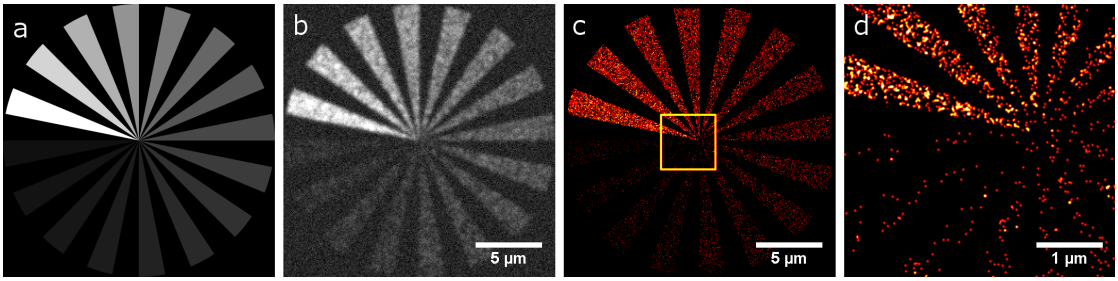


Figure 6: SMLM data generated with variable spatial density of molecules. a) gray-scale mask, b) average intensity time projection of raw data, c) SMLM data reconstruction, d) zoomed region indicated in c).

9.3. Additional sample drift

The generated molecular positions in the image sequence can be subjected to a lateral sample drift. Users need to specify the speed and direction of the drift, which is constant throughout the image sequence.

10. Performance evaluation

The performance of localization algorithms and post-processing methods can be evaluated by comparing the obtained molecular positions with the ground-truth positions. ThunderSTORM provides a tool for computing statistical measures related to the number of correctly detected molecules (TP, true positive detections), to the number of erroneous detections of non-existent molecules (FP, false positive detections), and to the number of missed molecules (FN, false negatives).

Localized molecular positions and ground-truth coordinates can be imported/exported to/from ThunderSTORM in various data formats, thus the performance can also be evaluated for other SMLM localization software.

10.1. Counting localized and missed molecules

The process of performance evaluation starts by pairing the localized molecules with the closest molecule in the ground-truth data, see Figure 7. The numbers of correctly and incorrectly identified molecules are counted as follows. If the distance between the paired molecules is smaller than a user-specified radius, then the localization is counted as a TP detection and the localized molecule is associated with the ground-truth position. If the distance is greater than or equal to that radius, then the localization is counted as a FP detection. Ground-truth molecules which were not associated with the localized molecules are counted as FNs. With a growing density of molecules it becomes more important how the algorithm performs the matching. To solve the problem of finding the correct matching between localized molecules and the ground-truth data, the Gale-Shapley algorithm [38] is used. KD-trees [15] are employed for an effective implementation.

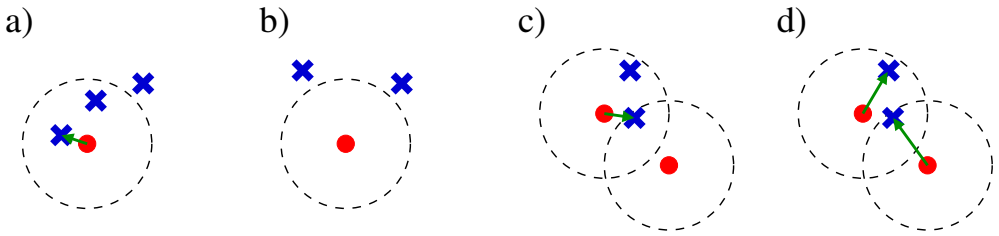


Figure 7: Counting localized and missed molecules. Red dot - ground-truth position of a molecule, blue cross - localized molecule, green arrow - association of a localized molecule with ground-truth position, dashed circle - detection tolerance radius. a) 1 TP + 2 FP, b) 1 FN + 2 FP, c-d) example of a situation, where c) greedy approach fails by finding 1 TP + 1 FP + 1 FN, and where d) Gale-Shapley algorithm finds a correct solution with 2 TP.

10.2. Precision and recall

Statistical measures related to the number of correctly or incorrectly detected molecules, or missed molecules, are the recall r (also called sensitivity) and the precision p (also called positive predictive value) [39, 7, 8]. Their definitions are given by

$$\begin{aligned} r &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}\,, \\ p &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}\,. \end{aligned}$$

Recall measures the fraction of correctly identified molecules, and precision measures the portion of correctly identified molecules in the set of all localizations. The theoretical optimum is achieved for values of recall and precision both equal to 1.0.

10.3. F1 score

For purposes of comparison between multiple algorithms, it is convenient to combine precision and recall into a single measure of performance with some trade-off between both values. A traditional method for this applies the F_1 score [39, 8] defined by

$$F_1 = \frac{2pr}{p+r}\,.$$

Values of the F_1 score close to zero indicate both bad recall and precision while values approaching 1.0 signify a good ratio between recall and precision.

10.4. Jaccard index

Another measure suitable for comparing similarity and diversity of sets of samples is the Jaccard index [39] defined by the formula

$$J = \frac{|\mathcal{A}\cap\mathcal{B}|}{|\mathcal{A}\cup\mathcal{B}|}\,.$$

Here $\mathcal{A}$ is the set of ground-truth molecular positions, $\mathcal{B}$ is the set of all molecular positions localized by processing the data, intersection $|\mathcal{A}\cap\mathcal{B}| = \mathrm{TP}$ gives the number of true positive detections, union $|\mathcal{A}\cup\mathcal{B}| = \mathrm{TP}+\mathrm{FP}+\mathrm{FN}$, and $|\cdot|$ denotes the size of the set. The Jaccard index ranges from zero to one and a theoretical optimum is achieved for values of the Jaccard index equal to 1.0.

10.5. RMS distance

For all molecules identified as true positives, we also calculate the root-mean square distance between the ground-truth positions of the molecules and their localizations.

References

[1] Rasband, W. S. ImageJ. U. S. National Institutes of Health, Bethesda, Maryland, USA. Available at http://imagej.nih.gov/ij/, (1997–2012).

[2] Rust, M. J., Bates, M., and Zhuang, X. Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). *Nature Methods* **3**(10), 793–796 (2006).

[3] Betzig, E., Patterson, G. H., Sougrat, R., Lindwasser, O. W., Olenych, S., Bonifacino, J. S., Davidson, M. W., Lippincott-Schwartz, J., and Hess, H. F. Imaging intracellular fluorescent proteins at nanometer resolution. *Science* **313**(5792), 1642–1645 (2006).

[4] Hess, S. T., Girirajan, T. P. K., and Mason, M. D. Ultra-high resolution imaging by fluorescence photoactivation localization microscopy. *Biophysical Journal* **91**(11), 4258–4272 (2006).

[5] Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J.-Y., White, D. J., Hartenstein, V., Eliceiri, K., Tomancak, P., and Cardona, A. Fiji: an open-source platform for biological-image analysis. *Nature Methods* **9**(7), 676–682 (2012).

[6] Edelstein, A., Amodaj, N., Hoover, K., Vale, R., and Stuurman, N. Computer control of microscopes using μManager. In *Current Protocols in Molecular Biology*, 14.20.1–14.20.17. John Wiley & Sons, Inc. (2010).

[7] Wolter, S., Schüttpelz, M., Tscherepanow, M., van de Linde, S., Heilemann, M., and Sauer, M. Real-time computation of subdiffraction-resolution fluorescence images. *Journal of Microscopy* **237**(1), 12–22 (2010).

[8] Křížek, P., Raška, I., and Hagen, G. M. Minimizing detection errors in single molecule localization microscopy. *Optics Express* **19**(4), 3226–35 (2011).

[9] Stetson, P. B. DAOPHOT - A computer program for crowded-field stellar photometry. *Publications of the Astronomical Society of the Pacific* **99**, 191 (1987).

[10] Holden, S. J., Uphoff, S., and Kapanidis, A. N. DAOSTORM: an algorithm for high- density super-resolution microscopy. *Nature Methods* **8**(4), 279–80 (2011).

[11] Huang, F., Schwartz, S. L., Byars, J. M., and Lidke, K. A. Simultaneous multiple-emitter fitting for single molecule super-resolution imaging. *Biomedical Optics Express* **2**(5), 1377–93 (2011).

[12] Starck, J.-L. and Murtagh, F. *Astronomical Image and Data Analysis*. Springer-Verlag, (2002).

[13] Izeddin, I., Boulanger, J., Racine, V., Specht, C. G., Kechkar, A., Nair, D., Triller, A., Choquet, D., Dahan, M., and Sibarita, J. B. Wavelet analysis for single molecule localization microscopy. *Optics Express* **20**(3), 2081–95 (2012).

[14] Šonka, M., Hlaváč, V., and Boyle, R. *Image Processing, Analysis, and Machine Vision*. Cengage Learning, 3rd edition edition, (2007).

[15] Knuth, D. E. *The Art Of Computer Programming*, volume 1. Addison-Wesley, Boston, 3rd edition, (1997).

[16] Henriques, R., Lelek, M., Fornasiero, E. F., Valtorta, F., Zimmer, C., and Mhlanga, M. M. QuickPALM: 3D real-time photoactivation nanoscopy image processing in ImageJ. *Nature Methods* **7**(5), 339–340 (2010).

[17] Parthasarathy, R. Rapid, accurate particle tracking by calculation of radial symmetry centers. *Nature Methods* **9**(7), 724–6 (2012).

[18] Kendall, M. and Stuart, A. *The Advanced Theory of Statistics*. London: Charles Griffin, (1979).

[19] Mortensen, K. I., Churchman, L. S., Spudich, J. A., and Flyvbjerg, H. Optimized localization analysis for single-molecule tracking and super-resolution microscopy. *Nature Methods* **7**(5), 377–381 (2010).

[20] Thompson, R. E., Larson, D. R., and Webb, W. W. Precise nanometer localization analysis for individual fluorescent probes. *Biophysical Journal* **82**(5), 2775–83 (2002).

[21] Stallinga, S. and Rieger, B. Accuracy of the Gaussian point spread function model in 2D localization microscopy. *Optics Express* **18**(24), 24461–24476 (2010).

[22] Smith, C. S., Joseph, N., Rieger, B., and Lidke, K. A. Fast, single-molecule localization that achieves theoretically minimum uncertainty. *Nature Methods* **7**(5), 373–5 (2010).

[23] Bevington, P. R. and Robinson, D. K. *Data reduction and error analysis for the physical science*. McGraw-Hill Higher Education. McGraw-Hill, (2003).

[24] Commons-Math. The Apache Commons Mathematics Library; version 3.2. Available at http://commons.apache.org/math/, (2013).

[25] O'Neill, R. Algorithm AS 47–function minimization using a simplex procedure. *Applied Statistics* **20**, 338–45 (1971).

[26] Rieger, B. and Stallinga, S. The lateral and axial localization uncertainty in super-resolution light microscopy. *Chemphyschem: a European journal of chemical physics and physical chemistry* **15**(4), 664–70 March (2014).

[27] Quan, T., Zeng, S., and Huang, Z.-L. Localization capability and limitation of electron-multiplying charge-coupled, scientific complementary metal-oxide semiconductor, and charge-coupled devices for superresolution imaging. *Journal of Biomedical Optics* **15**(6), 066005 (2010).

[28] Huang, B., Wang, W., Bates, M., and Zhuang, X. Three-dimensional super-resolution imaging by stochastic optical reconstruction microscopy. *Science* **319**(5864), 810–3 (2008).

[29] Cleveland, W. S. Robust Locally Weighted Regression and Smoothing Scatterplots. *Journal of the American Statistical Association* **74**(368), 829–836 (1979).

[30] Mlodzianoski, M. J., Schreiner, J. M., Callahan, S. P., Smolková, K., Dlasková, A., Santorová, J., Ježek, P., and Bewersdorf, J. Sample drift correction in 3D fluorescence photoactivation localization microscopy. *Optics Express* **19**(16), 15009–19 (2011).

[31] Huang, B., Jones, S. A., Brandenburg, B., and Zhuang, X. Whole-cell 3D STORM reveals interactions between cellular structures with nanometer-scale resolution. *Nature Methods* **5**(12), 1047–52 (2008).

[32] Malkusch, S., Endesfelder, U., Mondry, J., Gelléri, M., Verveer, P., and Heilemann. Coordinate-based colocalization analysis of single-molecule localization microscopy data. *Histochemistry and Cell Biology* **137**(1), 1–10 (2012). Springer-Verlag.

[33] Ronneberger, O., Baddeley, D., Scheipl, F., Verveer, P., Burkhardt, H., Cremer, C., Fahrmeir, L., Cremer, T., and Joffe, B. Spatial quantitative analysis of fluorescently labeled nuclear structures: problems, methods, pitfalls. *Chromosome Research* **16**(3), 523–562 (2008). Springer Netherlands.

[34] Baddeley, D., Cannell, M. B., and Soeller, C. Visualization of localization microscopy data. *Microscopy and Microanalysis* **16**(1), 64–72 (2010).

[35] Scott, D. W. Averaged shifted histograms: effective nonparametric density estimators in several dimensions. *The Annals of Statistics* **13**(3), pp. 1024–1040 (1985).

[36] Benchmarking of Single-Molecule Localization Microscopy Software. Biomedical Imaging Group, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. Available at http://bigwww.epfl.ch/smlm/, (2013).

[37] Hirsch, M., Wareham, R. J., Martin-Fernandez, M. L., Hobson, M. P., and Rolfe, D. J. A stochastic model for electron multiplication charge-coupled devices–from theory to practice. *PLoS ONE* **8**(1), e53671 (2013).

[38] Gale, D. and Shapley, L. S. College admissions and the stability of marriage. *The American Mathematical Monthly* **69**(1), 9–15 (1962).

[39] Tan, P.-N., Steinbach, M., and Kumar, V. *Introduction to Data Mining.* Addison-Wesley Longman Publishing Co., (2005).