Histochem Cell Biol
DOI 10.1007/s00418-014-1211-4

ORIGINAL PAPER

SimpleSTORM: a fast, self-calibrating reconstruction algorithm for localization microscopy

Ullrich Köthe · Frank Herrmannsdörfer · Ilia Kats · Fred A. Hamprecht

Accepted: 5 March 2014

Abstract Although there are many reconstruction algorithms for localization microscopy, their use is hampered by the difficulty to adjust a possibly large number of parameters correctly. We propose SimpleSTORM, an algorithm that determines appropriate parameter settings directly from the data in an initial self-calibration phase. The algorithm is based on a carefully designed yet simple model of the image acquisition process which allows us to standardize each image such that the background has zero mean and unit variance. This standardization makes it possible to detect spots by a true statistical test (instead of hand-tuned thresholds) and to de-noise the images with an efficient matched filter. By reducing the strength of the matched filter, SimpleSTORM also performs reasonably on data with high-spot density, trading off localization accuracy for improved detection performance. Extensive validation experiments on the ISBI Localization Challenge Dataset, as well as real image reconstructions, demonstrate the good performance of our algorithm.

Keywords Localization microscopy · STORM reconstruction · Matched filter · Noise normalization · Self-calibration

Ullrich Köthe and Frank Herrmannsdörfer have contributed equally to this work.

U. Köthe (✉) · F. Herrmannsdörfer · I. Kats · F. A. Hamprecht
Multi-Dimensional Image Processing Group, University of Heidelberg, Speyerer Strasse 6, 69115 Heidelberg, Germany
e-mail: ullrich.koethe@iwr.uni-heidelberg.de

Introduction

Localization microscopy techniques such as STORM and PALM have become major tools for cell biology (Bates et al. 2008; Heilemann 2010; Henriques and Mhlanga 2009; Huang 2010). Despite differences in the experimental procedures, these techniques rest on a unified computational principle: Super-resolution images are constructed by means of subpixel-accurate spot detection in a sequence of images taken at optical resolution. Various algorithms have been proposed for this task, and many of them are available in open-source software. We will briefly review existing solutions in section "Related work" below.

While many of these algorithms are able to obtain good reconstructions in principle, their practical success requires a careful adjustment of various configuration settings. This parameter tuning may be difficult even for experts and raises the entry barrier for localization microscopy novices. To overcome this problem, SimpleSTORM was designed to determine all necessary parameters automatically from the raw image data during an initial *self-calibration phase* that precedes the actual reconstruction phase. It can thus produce good reconstructions with minimal user input, while still allowing optional configuration by experts for non-standard use cases.

SimpleSTORM is based on an accurate model of the image acquisition process. It assumes that Poisson-distributed photon counting noise is the dominant noise source. However, we explicitly account for the fact that photon counts are only observed after a number of transformations by amplification stages and digitization. Under the assumption that the combined transformation can be described by a linear equation, we can recover Poisson-distributed intensities by inverting this linear equation after robust estimation of its parameters (gain and offset). Furthermore, we assume

Published online: 11 April 2014

that the point spread function (PSF) is fixed throughout the image sequence and can be determined as a nonparametric or parametric (Gaussian) model. Finally, we assume that the background intensity varies at a much coarser scale than the PSF width and that less than half of the pixels in any sufficiently large window contain signal. Under these assumptions, the model parameters (gain and offset, PSF width, local background intensity) can be estimated automatically in a self-calibration and preprocessing phase.

In the reconstruction phase, the model parameters are used to transform each frame such that it can be considered as a sum of a background-free fluorescence signal and additive Gaussian noise with zero mean and unit variance. Fluorescence spots can thus be recognized by a simple statistical test: a pixel whose intensity is higher than three times the noise standard deviation contains signal with a probability of about 99.7 % (the threshold of the test can be adjusted to control the detection sensitivity of our algorithm). Since the PSF spreads over several pixels, whereas the noise of neighboring pixels is independent, it is even more unlikely that three adjacent pixels exceed the threshold just by chance. The combination of both criteria defines a reliable mask for spot detection. Pixels under (and near) the mask are finally convolved with a matched filter (i.e., a Gaussian filter corresponding to the PSF) for optimal noise reduction and interpolated to the desired resolution using a cubic spline. The coordinates of local intensity maxima after interpolation are reported as the detected spots.

Specifically, our algorithm proceeds in these steps:

1. Robust estimation of the gain and offset parameters
2. Estimation of the width of a Gaussian PSF via the Levenberg-Marquardt algorithm
3. Linear intensity transform into unit gain and zero offset to make the noise approximately Poisson distributed
4. Anscombe transform of the intensities to make the noise approximately normal
5. Dynamic background estimation and subtraction
6. Statistical test to determine the detection mask according to the specified sensitivity
7. Matched filtering with the PSF for optimal noise reduction
8. Cubic spline interpolation to specified subpixel accuracy and maxima detection

A typical result is shown in Fig. 1. On a standard laptop, our algorithm is able to process about 50 frames per second for a typical raw image size of 200 $\times$ 200 pixels. It performed favorably in the recent ISBI Localization Microscopy Challenge[1] that carefully tested more than 20 reconstruction algorithms. In particular, SimpleSTORM achieved high localization accuracy on high-density data, where a relatively large number of spots were simultaneously switched on in order to minimize total acquisition time. Software for our algorithm is freely available in an easy-to-use GUI program.[2]

Related work

The classical reconstruction approach is based on direct fitting of the PSF to every spot candidate. A model of the PSF with adjustable parameters must be given. Usually, an isotropic Gaussian PSF like (6) or an anisotropic variant of it is sufficiently accurate (Thompson et al. 2002), although alternative models may be required under certain circumstances (Stallinga and Rieger 2012). The free parameters of the model are optimized by a nonlinear fitting algorithm (e.g., the Levenberg–Marquardt method) in order to minimize the least-squares residual between the fitted model and the intensities of a spot candidate. This method is the basis of the popular rapidSTORM software (Wolter et al. 2010), the ImageJ plugins Octane (Yu 2011), PeakFit (Herbert 2012) and GraspJ [offering GPU acceleration, Brede and Lakadamyali (2012)], as well as the Localization Microscopy MicroManager plugin [Stuurman (2012) and the Python localization microscopy environment (PYME, Baddeley (2012)].

An even simpler approach avoids iterative optimization by computing the centroid of each spot directly, either in terms of the intensity-weighted mean (Thompson et al. 2002) or the fluoroBancroft method (Andersson 2008). These methods were found to be only slightly less accurate then the least-squares fit when properly parametrized (Thompson et al. 2002; Hedde et al. 2009) and form the basis of fast reconstruction solutions such as the hardware-accelerated method of (Grüll et al. 2011) and the QuickPALM ImageJ plugin (Henriques et al. 2010).

Both fitting and direct methods have difficulties with the reconstruction of high-density image sequences, because these images violate the basic model assumption that neighboring spots do not overlap. Greedy procedures for fitting overlapping spots where proposed by (Egner et al. 2007), who adapt Högbom's classical CLEAN algorithm (Högbom 1974) to localization microscopy, the DAOSTORM algorithm from (Holden et al. 2011), an adaptation of DAOPHOT, a well-known algorithm from astronomy (Stetson et al. 1987), and the multi-emitter fitting algorithm of (Huang et al. 2011), who fit up to $N_{\max}$ overlapping PSFs simultaneously and select the most likely spot number by a statistical test.

[1] http://bigwww.epfl.ch/smlm/challenge/.

[2] https://github.com/ukoethe/simple-STORM.

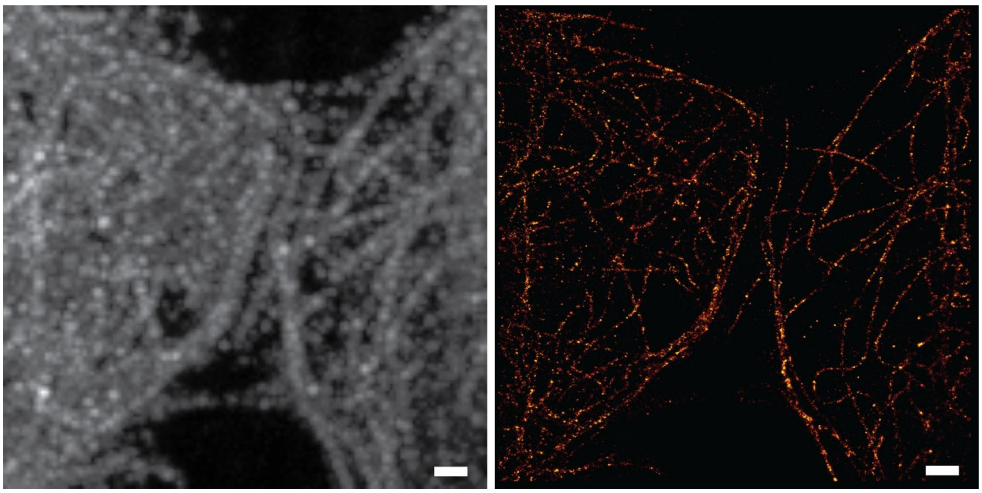

Fig. 1 Microtubuli, labeled with ATTO520 in HeLa-cells (raw data courtesy of Mike Heilemann). *Left* maximum projection of the raw data. *Right* reconstructed high-resolution image. *Scalebar* represents 1 μm

A more principled model for overlapping spots is provided by *compressed sensing* (Zhu et al. 2012). Here, a PSF candidate is initialized at all pixels of a fine grid (e.g., having one-eighth the pixel size of the original image). The fit is then performed under a strong *sparsity prior* which ensures that only the center pixels of true spots get non-zero activation. While this method is more accurate than DAOSTORM, it is also very expensive (about a hundred times slower). Recently, substantial improvements of the sparse reconstruction approach have been achieved by more sophisticated modeling and optimization methods (Kim et al. 2013; Min et al. 2013). However, it is as yet unclear if the improvements over simple algorithms like SimpleSTORM warrant the added complexity.

Proper noise modeling is a critical ingredient for reliable distinction between true spots and noise artifacts. Many authors simply use a generic additive noise model and select spots whose intensity exceeds the background by a certain multiple of the background's noise SD (Thompson et al. 2002; Grüll et al. 2011). Frequently, the image is preprocessed by averaging filters (Wolter et al. 2010; Huang et al. 2011), Gaussian filters (Křížek et al. 2011) or a wavelet transform (Izeddin et al. 2012) before thresholding. However, non-uniform background intensity and intensity-dependent background noise make the choice of an appropriate threshold very difficult. Heuristic solutions to this problem lead to algorithms with many adjustable parameters that are very hard to use.

Moreover, it has been shown in (Abraham et al. 2009) that a maximum likelihood fit based on a Poisson noise model outperforms the least-squares algorithm which implicitly assumes additive noise. A fast GPU-accelerated version of this method is described in (Smith et al. 2010). However, the Poisson model cannot be applied directly to the observed image intensities because they deviate from the recorded photon counts due to the action of amplification. Amplified intensities are no longer Poisson distributed. The amplification can be inverted when gain factor and offset are known, but this requirement is apparently overlooked sometimes [e.g., (Andersson 2008; Grüll et al. 2011; Brede and Lakadamyali 2012)]. A simple gain and offset estimation procedure using dedicated calibration images was proposed in (Lidke et al. 2005). A more convenient self-calibration algorithm was introduced in (Boulanger et al. 2010), who adaptively subdivide the given image sequence to determine regions of homogeneous intensity for gain estimation. This method is very similar in spirit to our approach in section "Noise normalization", but their algorithm is considerably more complicated.

Methods

Matched filters

It was shown by (Abraham et al. 2009) that a maximum likelihood fit outperforms least-squares fitting when the noise follows a Poisson distribution. On the other hand, it is well known that maximum likelihood and least-squares fitting are equivalent under additive Gaussian noise. Moreover, least-squares fitting can be replaced by *matched filtering* (Turin 1960) when the mean background intensity is zero. Under these conditions, these algorithms are mathematically equivalent (as we show below), but filtering is much more efficient, in particular when the matched filter is a Gaussian PSF. This motivates our desire to work on *standardized images*, and robust standardization of the given image sequence is at the heart of the SimpleSTORM algorithm.

We briefly recall the relationship between matched filtering and least-squares fitting. Let the image $s(x, y)$ contain just a single spot with unknown scaling μ_0 at an unknown location (x_0, y_0). We assume that the noise is additive Gaussian noise with mean zero and variance σ^2, i.e., our image model is

$$\begin{aligned} s(x,y) &= f(x,y) + n(x,y) \\ f(x,y) &= \mu_0 \,\mathrm{PSF}(x - x_0, y - y_0) \\ n(x,y) &\sim \mathcal{N}(0, \sigma^2) \end{aligned} \tag{1}$$

The least-squares algorithm determines the scaling μ_m and location x_m, y_m of a model $h(x, y)$ such that the least-squares residual

$$\mu_m^*, x_m^*, y_m^* = \underset{\mu_m, x_m, y_m}{\arg\min} \int_{\mathbb{R}^2} (s(x,y) - \mu_m h(x - x_m, y - y_m))^2 \, d\boldsymbol{x}$$

is minimized. After expanding the integrand, this is equal to

$$\begin{aligned} &\underset{\mu_m, x_m, y_m}{\arg\min} \int_{\mathbb{R}^2} s(x,y)^2 \, d\boldsymbol{x} + \mu_m^2 \int_{\mathbb{R}^2} h(x - x_m, y - y_m)^2 \, d\boldsymbol{x} \\ &\quad - 2\mu_m \int_{\mathbb{R}^2} f(x,y) h(x - x_m, y - y_m) \, d\boldsymbol{x} \\ &\quad - 2\mu_m \int_{\mathbb{R}^2} n(x,y) h(x - x_m, y - y_m) \, d\boldsymbol{x} \end{aligned}$$

The first integral is a constant (since the signal is fixed), the last one has zero expected value (since the noise is uncorrelated), and both terms can be dropped from the optimization. After reversing the sign, we obtain the equivalent optimization problem

$$\begin{aligned} &\underset{\mu_m, x_m, y_m}{\arg\max}\, 2\mu_m \int_{\mathbb{R}^2} f(x,y) h(x - x_m, y - y_m) \, d\boldsymbol{x} \\ &\quad - \mu_m^2 \int_{\mathbb{R}^2} h(x - x_m, y - y_m)^2 \, d\boldsymbol{x} \end{aligned}$$

Setting the derivative w.r.t. μ_m to zero gives an analytic solution for μ_m^*

$$\mu_m^* = \frac{\int_{\mathbb{R}^2} f(x,y) h(x - x_m, y - y_m) \, d\boldsymbol{x}}{\int_{\mathbb{R}^2} h(x - x_m, y - y_m)^2 \, d\boldsymbol{x}}$$

resulting in a reduced optimization problem for the location

$$x_m^*, y_m^* = \underset{x_m, y_m}{\arg\max} \frac{\left(\int_{\mathbb{R}^2} f(x,y) h(x - x_m, y - y_m) \, d\boldsymbol{x}\right)^2}{\int_{\mathbb{R}^2} h(x - x_m, y - y_m)^2 \, d\boldsymbol{x}}$$

The numerator is the squared correlation function between $f(x, y)$ and $h(x-x_\mathrm{m}, y-y_\mathrm{m})$. Expanding it in terms of the Cauchy-Schwarz inequality yields

$$\begin{aligned} &\frac{\left(\int_{\mathbb{R}^2} f(x,y) h(x - x_m, y - y_m) \, d\boldsymbol{x}\right)^2}{\int_{\mathbb{R}^2} h(x - x_m, y - y_m)^2 \, d\boldsymbol{x}} \\ &\quad \le \frac{\int_{\mathbb{R}^2} f(x,y)^2 \, d\boldsymbol{x} \cdot \int_{\mathbb{R}^2} h(x - x_m, y - y_m)^2 \, d\boldsymbol{x}}{\int_{\mathbb{R}^2} h(x - x_m, y - y_m)^2 \, d\boldsymbol{x}} \\ &= \int_{\mathbb{R}^2} f(x,y)^2 \, d\boldsymbol{x} = \text{const.} \end{aligned}$$

and equality (i.e., the maximum possible value) is only achievable when $h(x - x_m, y - y_m) = \kappa f(x, y)$. The maximum is thus obtained for $x_m^* = x_0$ and $y_m^* = y_0$ as desired. Since κ can be chosen arbitrarily, $h(x, y)$ can be defined as a *matched filter*

$$h(x,y) = \mathrm{PSF}(x,y)$$

The optimal estimate of the spot location is therefore the point where the convolution

$$g(x,y) = \mathrm{PSF}(-x,-y) \times s(x,y) \tag{2}$$

assumes its maximum (recall that correlation is equivalent to convolution with the mirrored kernel). This point can be conveniently determined by spline interpolation of g, see section "Spot localization."

To use matched filters in localization microscopy, the following requirements must be met:

1. The PSF must be known. We address this in section "Estimation of the PSF."
2. The PSF should be uniform throughout the image. This condition may not be fulfilled when some spots are out of focus. However, this is not a major problem in practice, because the filter h degrades gracefully: although no longer the best possible filter, it still performs reasonably as long as the PSF is not too far off.
3. Each image should contain only one spot. This is clearly violated in practice and in fact undesirable. But this is no problem as long as the spot density is not too high. Since the PSF decreases quickly, the filter response at point (x, y) is not influenced by spots that are sufficiently far away. The detector suffers only a minor degradation at overlapping spots whose distance remains larger than the PSFs full width at half maximum (FWHM). The spot density can easily be controlled by the experimental setup.
4. The image's background intensity must be zero. We use a standard background subtraction procedure as described in section "Background subtraction."
5. The noise must be additive Gaussian. This is the most serious obstacle, because the noise variance is actually a function of the intensity and therefore not

additive. To rescue the matched filter approach, we transform the original image intensities so that the noise is approximately turned into additive Gaussian noise. Our *noise normalization* procedure is described in section "Noise normalization."

Noise normalization

Matched filtering is only optimal when the noise is additive and Gaussian distributed. However, localization microscopy is based on a photon counting process, which instead follows a Poisson distribution with intensity-dependent variance. The probability of observing k photons in pixel (x, y) is given by:

$$p(k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

where $\lambda = \lambda(x, y)$ is the expected count, i.e. true intensity (we dropped the dependency on (x, y) to improve readability). Moreover, these counts are not observed directly but are subjected to several amplification stages and discretized into a finite set of gray levels. If we assume linear amplification characteristics and neglect discretization effects,[3] the observed image gray levels k' depend on k according to the linear function

$$k' = a\,k + b \tag{3}$$

where a denotes the total gain factor and b the dark signal (offset). The noise in k' is no longer Poisson distributed. One can easily see this by recalling that both the mean and variance of a Poisson distribution are equal to λ. In contrast, mean and variance of k' are

$$E\left[k'\right] = a\,E[k] + b = a\,\lambda + b$$
$$\mathrm{Var}\left[k'\right] = a^2 \quad Var[k] = a^2\lambda$$

and these quantities are in general different. Consequently, it is *incorrect* to apply algorithms which rest on the assumption of Poisson or Gaussian noise (like matched filtering) directly to the observed image $k'(x, y)$. Fortunately, there is an easy way out: the *Anscombe transform* (Anscombe 1948)

$$q(x, y) = 2\sqrt{k(x, y) + \frac{3}{8}}$$

turns a Poisson-distributed signal $k(x, y)$ into an approximately Gaussian distributed one $q(x, y)$ with unit variance, regardless of the value of k (as long as $k \geq 4$). However, in order to apply the Anscombe transform, we need to know the coefficients a and b that map the observed gray values k' back into the Poisson-distributed counts k:

$$q(x, y) = 2\sqrt{\frac{k'(x, y) - b}{a} + \frac{3}{8}} \tag{4}$$

Determining a and b turns out to be tricky. The standard solution is to record dedicated calibration images where the mean and variance of k' can be computed easily (Lidke et al. 2005). Then, a linear regression through a set of pairs $\left(E[k'], \mathrm{Var}[k']\right)$ with different k' directly provides the desired coefficients. However, this approach is inconvenient when the camera is mounted on a microscope. Therefore, we seek to determine these coefficients from the raw localization images themselves by self-calibration.

Among a large number of ideas we tried, the following turned out to be the most stable. Consider a pixel (x, y) whose true intensity is constant over time, i.e., the pixel shows background or a bead, but no blinking molecule. Then, we can easily compute its average intensity and variance over time and obtain a point for the linear regression. The difficulty is that we do not know which pixels have this property. The following consideration shows a way to identify them: Whenever the true intensity is not constant, the apparent variance is larger than it would otherwise be, because

$$\mathrm{Var}\left[f(t) + n(t)\right] = \mathrm{Var}\left[f(t)\right] + \mathrm{Var}[n(t)]$$

where $f(t)$ is a time-dependent signal and $n(t)$ denotes noise which is uncorrelated with the temporal behavior of f. When $f(t) = \text{const}$, its variance is zero, resulting in the minimal possible value of $\mathrm{Var}\left[f(t) + n(t)\right]$. Otherwise, the total variance increases. This leads to the following algorithm

1. Select n image locations at random.
2. Compute mean and variance of the corresponding pixels over the first T frames of the sequence ($T = 200$ works well in practice, but the value can be adjusted). Create the scatter plot of the resulting mean/variance pairs.
3. Use the RANSAC algorithm (Fischler and Bolles 1981) to compute the lower leaning line of the scatter plot: Repeat $k = 10{,}000$ times:
   (a) Select two points at random and compute the line through these points.
   (b) Determine the number of inliers of this line, i.e., the number of points whose distance from the line is at most a twentieth of the range between the minimal and the maximal value.
   (c) Keep the line with the maximum number of inliers as the best estimate of the lower leaning line.

[3] This is possible because the discretization noise is typically much smaller than the noise from other sources.

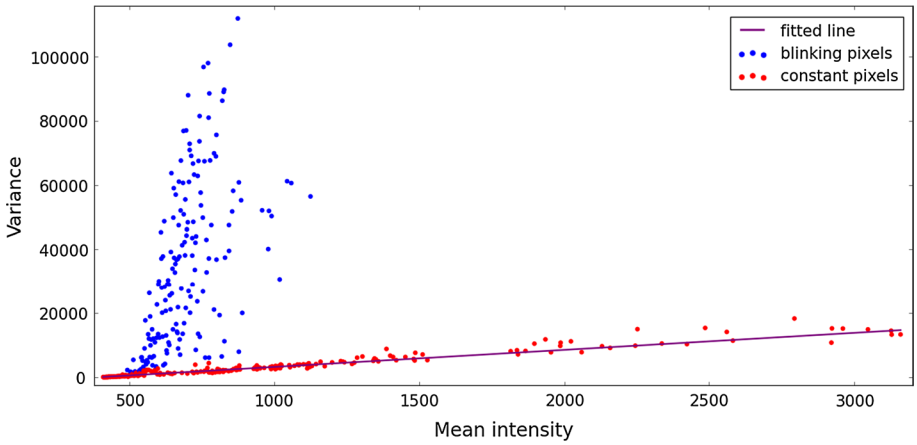

Fig. 2 RANSAC-fitting of a lower leaning line to a scatter plot showing variance versus mean intensity for randomly selected pixels. Slope and intercept of this line determine the correction coefficients in Eq. (4)

4. The coefficients a and b now correspond to the slope and intercept of the lower leaning line, because this line contains precisely the points whose intensity was constant over time. All points not near the lower leaning line are "contaminated" by intensity variations and therefore ignored.

Figure 2 shows an example of the fit according to this algorithm. It can be seen that it does indeed detect the lower leaning line of the scatter plot, which corresponds to the pixels with constant intensity.

Occasionally, the range of intensities on the lower leaning line is rather small (e.g., when no beads are present), resulting in noisy gain estimates. This can be fixed by iterative post-optimization: We use the estimated parameters to transform the data according to (4) and compute an intensity histogram for each of the first 200 frames. These histograms contain mixture distributions with one mixture component for the background and one for the rest. We fit a Gaussian to the background mixture component in each histogram and compute the average variance of these Gaussians. When this value is not within 5 % of unity, the gain factor is multiplied with the measured variance and the procedure repeated. This algorithms lead to stable gain estimates within a few iterations.

Background subtraction

Once, the noise model is known, we transform the observed images $k'_t(x, y)$ into noise-normalized ones $q_t(x, y)$ (where t indicates time) according to section "Noise normalization." The next step is the estimation of the background intensity. It is based on the standard assumption that the background varies much slower then the actual signal both spatially and over time. We split the dataset into non-overlapping blocks of size $\rho^2 \times \tau$, where ρ is the block size in the two spatial directions, and τ is the block size along the time direction. Default values of $\rho = 30$ and $\tau = 20$ work well in all our experiments. If necessary, the user can adjust these settings. This is easy because he or she can determine how fast the background varies by simple visual inspection of the data.

In each block, the median of the gray values s is computed. The median is preferable to the mean because it is more stable when the block contains non-background pixels (blinking spots and beads): these pixels lead to a significant upward bias in the mean, whereas the median increases only marginally. The median values are placed on the grid points defined by the centers of the blocks and interpolated to the original image resolution (both in spatial and time direction) using a Catmull-Rom spline which ensures smooth (i.e., differentiable) interpolation and thus avoids blocking artifacts in the background estimate $\beta_t(x, y)$. After subtracting β_t from the noise-normalized signal q_t, we obtain the signal s_t which has unit variance in all pixels and zero mean in the background:

$$s_t(x, y) = q_t(x, y) - \beta_t(x, y) \tag{5}$$

We call the resulting image s_t the *standardized* image because it now conforms to the requirements of the matched filter method. Figure 3 shows examples.

Estimation of the PSF

In order to apply the matched filter method, an accurate estimate of the PSF is required. We investigated two approaches to PSF estimation, a nonparametric and a parametric one. In both cases, the PSF is estimated by averaging over many spots *after* image standardization. The nonparametric method determines the PSF in the form of an optimal *Wiener filter* which is based on the *magnitude spectrum* of the signal. Let

$$S_t = \mathcal{F}[s_t]$$

the Fourier transform of standardized frame s_t. Then the average power spectrum over T frames is

$$P = \frac{1}{T} \sum_{t=1}^{T} |S_t|^2$$

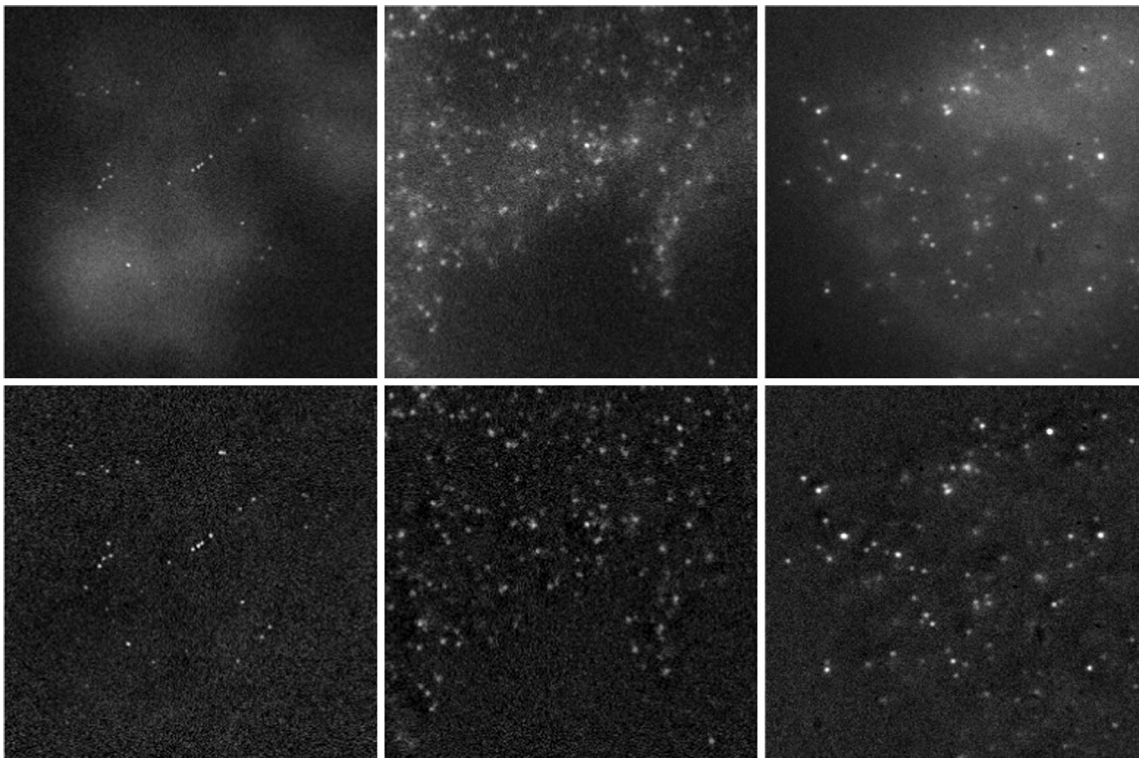

Fig. 3 *Top* original images (from *left* frame 50 of the artificial Tubulin2 dataset, actin conjugated to mEos2, and tubulin labeld with Cy5, see section "Results" for details on the data). *Bottom* the same frames after standardization

where $|S_t|$ is the point-wise magnitude of the complex frequency response. The Wiener filter is then defined as

$$W = \left(\frac{P - P_{\text{noise}}}{P}\right)_+$$

where P_{noise} is the expected power spectrum of the noise and $(.)_+ = \max(0, .)$ is the hinge function that truncates negative values at zero (negative values can occur because we estimate W from a finite sample). Since the noise (after standardization) is additive Gaussian noise with zero mean and unit variance, its expected power spectrum is simply $P_{\text{noise}} = 1$. The matched filter can now be computed by multiplication of each frame's Fourier transform with the Wiener filter, followed by inverse Fourier transform

$$\hat{s}_t = \mathcal{F}^{-1}[S_t \cdot W]$$

In contrast, the parametric method assumes that the PSF is shaped like a Gaussian, which is a very good approximation for typical microscopes. Under this assumption, the self-calibration only needs to determine a single parameter, the PSF scale σ_{PSF}. Clearly, the variance of a single parameter estimate is much smaller than the variance of an entire nonparametric PSF estimate when the same number of samples is used. The Gaussian model is therefore preferable when it conforms to the actual PSF shape, whereas the Wiener filter should be applied otherwise. Due to image standardization, our estimate σ_{PSF} is related to the PSF size σ^*_{PSF} of the raw data by the relation $\sigma_{\text{PSF}} = \sqrt{2}\,\sigma^*_{\text{PSF}}$.

The Gaussian fit can be performed both in the spatial and in the Fourier domain. The Fourier domain approach starts in the same way as in the nonparametric case, i.e., we compute the average power spectrum of the noise-normalized signal. To simplify subsequent computations, we cut out sufficiently large *square* ROIs from the original frames and compute their average power spectrum P'. Now, instead of using the power spectrum directly to define a Wiener filter, we compute the average magnitude spectrum $\sqrt{P'}$ and use it to fit a Gaussian function. Since the Fourier transform of a Gaussian PSF and the corresponding magnitude spectrum are again rotationally symmetric Gaussian functions, the model is

$$g(r|w_1, w_2, w_3) = w_1 \exp\left(-\frac{r^2}{2w_2^2}\right) + w_3 \tag{6}$$

where $r = \sqrt{u^2 + v^2}$ is the distance of the point from the origin. The parameters w_i are chosen by means of nonlinear least-squares optimization (using the Levenberg–Marquardt algorithm) such that the squared difference between the model and the power spectrum is minimized

$$w_1, w_2, w_3 = \arg\min_{w_i} \sum_{u,v} \left[g(r|w_1, w_2, w_3) - \sqrt{P'(r)}\right]^2$$

Parameters w_1 and w_3 account for the signal and noise intensities respectively, whereas the desired PSF scale in the spatial domain can be obtained from the parameter w_2 by the simple relation

$$\sigma_{\mathrm{PSF}} = \frac{W}{2\pi w_2}$$

where W is the width of the ROI used to compute the magnitude spectrum (recall that we select squared ROIs to simplify matters). The fit in the spatial domain is performed similarly, but the simple averaging via the average power spectrum is not possible here. Instead, we fit the PSF independently to a number of easy-to-detect spots, and then define σ_{PSF} as the median of the parameter w_2 of the individual estimates. (Thus, we also use the standard fitting algorithm of (Thompson et al. 2002), but only to estimate σ_{PSF}.)

Finally, the result of matched filtering is

$$\hat{g}_t = s_t \star \mathrm{gauss}_{\sigma_{\mathrm{PSF}}}$$

where $\star$ denotes convolution. Experimental results are reported in section "Validation datasets".

Spot detection

Spot detection is performed in each frame after image standardization. Since the background now contains only Gaussian additive noise with zero mean and unit variance, a standard statistical test can be used to detect pixels whose intensity is unlikely to be background. For a given p value, corresponding to the probability of false positives, the intensity threshold is

$$t \geq \sqrt{2}\,\mathrm{erfi}(2p-1) \tag{7}$$

where erfi(.) is the inverse error function. That is, a normalized pixel with intensity at least t has a probability of at most p to represent background. For example, p values of 1 and 0.1 % correspond to thresholds 2.3 and 3.1 and indicate that one gets one false positive on average per 10×10 and 30×30 window, respectively. Note that these thresholds are independent of the image content. In contrast to many existing algorithms, there is no need for manual threshold adjustment or complicated threshold selection heuristics in SimpleSTORM.

The false positive rate can be further reduced by noticing that true spots always cover several pixels, whereas the noise in neighboring pixels is uncorrelated. Therefore, the probability that multiple adjacent pixels are above threshold *simultaneously* is low for background pixels, but high for true spots. In practice, we accept a spot if at least three connected pixels exceed the threshold for the chosen p value. The probability of false positives in this setting can be approximated by a binomial distribution with parameter p. For example, for $p = 1$ %, the probability that three or more background pixels are above threshold in a given 3×3 window is below 0.01 %, for $p = 0.1$ % the resulting probability is $\approx 10^{-5}$ %. The result of this step is a set of spot masks, i.e., connected regions above threshold that contain at least three pixels.

Spot localization

In order to perform spot localization, we first subject the normalized image to the matched filter described in section "Matched filters", i.e., we filter with a Gaussian whose size σ_{PSF} has been determined according to section "Estimation of the PSF." We do not apply this filter before spot *detection* because this would introduce complicated correlation between the noise of neighboring pixels, making the distinction between signal and noise more difficult.

The theory of matched filtering suggests that each local maximum of the filtered image corresponds to a location of best match between the data and the spot model. This location needs to be determined to subpixel accuracy, and a residual error of 1/5 to 1/10 of a pixel is typically achievable. The simplest possibility to do so is via cubic spline interpolation of the filtered image to the desired resolution. The upsampling ratio can be chosen by the user and is typically between 8 and 16. To save time, the interpolation is only performed in sufficiently large rectangles around each spot mask (i.e., the bounding rectangle plus four pixels in every direction). Each local maximum of the interpolated image which is covered by one of the spot masks is returned as a detected spot.

Important spot properties such as signal-to-noise ratio and anisotropy can be readily obtained from the properties of the interpolated image. Let g be the intensity of the interpolated image at the local maximum position, and g_{xx}, g_{xy}, g_{yy} the corresponding second derivatives (these derivatives can be easily computed analytically from the spline representation). Then the following relations can be derived from basic properties of Gaussian functions. Since the noise has unit variance before matched filtering, the (unfiltered) signal-to-noise ratio is simply given by

$$\mathrm{SNR} = 2g$$

(the factor of 2 accounts for the smoothing effect of the matched filter). The SD of the 2D localization error Δx after matched filtering, defined in units of the pixel spacing, is then

$$\mathrm{StdDev}[\Delta \boldsymbol{x}] = \frac{1}{\sqrt{\pi}\, g} \tag{8}$$

This is of the expected form, because g is proportional to $\sqrt{N}$ (the square root of the photon count) due to the action of the Anscombe transform.

The maximum and minimum radius of a potentially anisotropic spot can be computed from the eigenvalues of the second derivative matrix

$$\kappa_{1,2} = \frac{1}{2}\left(g_{xx} + g_{yy} \pm \sqrt{g_{xx}^2 + g_{yy}^2 + 4\,g_{xy}^2 - 2\,g_{xx}g_{yy}}\right)$$

by the expressions

$$s_{\max} = \sqrt{-g/\kappa_1 - \sigma_{\text{filter}}^2}$$
$$s_{\min} = \sqrt{-g/\kappa_2 - \sigma_{\text{filter}}^2}$$

(with $\sigma_{\text{filter}} = \sigma_{\text{PSF}}$ for matched filtering) and their ratio gives the spot anisotropy

$$\text{AI} = \frac{s_{\max}}{s_{\min}}$$

Spot radii and anisotropy can serve as additional selection criteria to remove undesired anisotropic spots, or as a means to determine spot depth in 3D localization microscopy with cylindrical lenses (however, depth estimation is not yet implemented in the SimpleSTORM software).

It should be noted that each spot mask can contain multiple local maxima. This is desirable because it allows us to detect overlapping spots, provided that the overlap is not too big. Specifically, when we skip the filtering step altogether and detect maxima directly in the interpolated standardized images, overlapping spots of equal intensity remain separable (i.e., give rise to distinct maxima) when the distance of their centers exceeds $2\sigma_{\text{PSF}}$. The localization error $\Delta x'$ then becomes

$$\text{StdDev}[\Delta \boldsymbol{x}'] = \frac{\pi\,\sigma_{\text{PSF}}^2}{\sqrt{3}\,g'}$$

where g' is the intensity at the location of the maximum and σ_{PSF} is taken in units of the pixel spacing. This possibility is an advantage of our method over reconstruction algorithms that explicitly fit the PSF to the image data: The fitting approach only works reliably when the spots do not overlap significantly, i.e., when the spot distance is about twice as big.

Results

Validation datasets

Validation data should be both realistic and accompanied with ground truth. A good compromise between these conflicting goals is achieved by the simulated image sequences that have been designed with great care for the ISBI Single-Molecule Localization Microscopy Challenge 2013 and

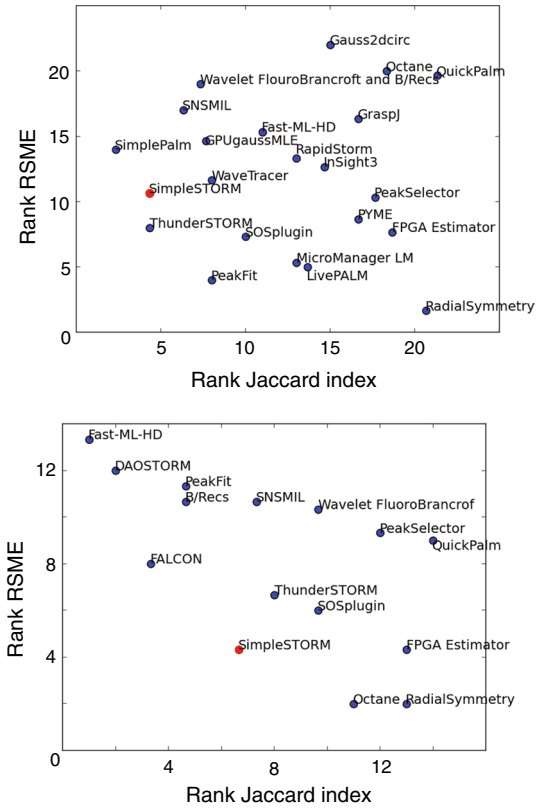

Fig. 4 Average algorithm rankings in the ISBI Localization Challenge (*top* low density, *bottom* high-density datasets). Smaller distance to the origin is better

can be downloaded freely.[4] The ground truth is available for four "training" datasets, and we used these for most of our quantitative experiments. Six additional datasets with undisclosed ground truth were used in the contest, where SimpleSTORM also performed well. Since the organizers did not yet publish the results officially, we compiled the ranking charts in Fig. 4 from the material handed out at the competition workshop. The charts place each participant according to the average ranking in two performance metrics (Jaccard index and RMSE, see below) in the "high-spot density" and "low-spot density" data categories.

Two of the training datasets show simulated tubulin tubes, and we will refer to them as "Tubulin 1" and "Tubulin 2". They consist of 2400 frames with a resolution of 256 px × 256 px (at 150 nm per pixel) and strong autofluorescent background. The Tubulin 2 dataset has a higher noise level than Tubulin 1. The other two datasets simulate a bundle of tubulin tubes. There is a long sequence with 12000 frames and almost no overlapping spots and a high-density set with 361 frames and severely overlapping spots. These

[4] http://bigwww.epfl.ch/smlm/datasets/index.html.

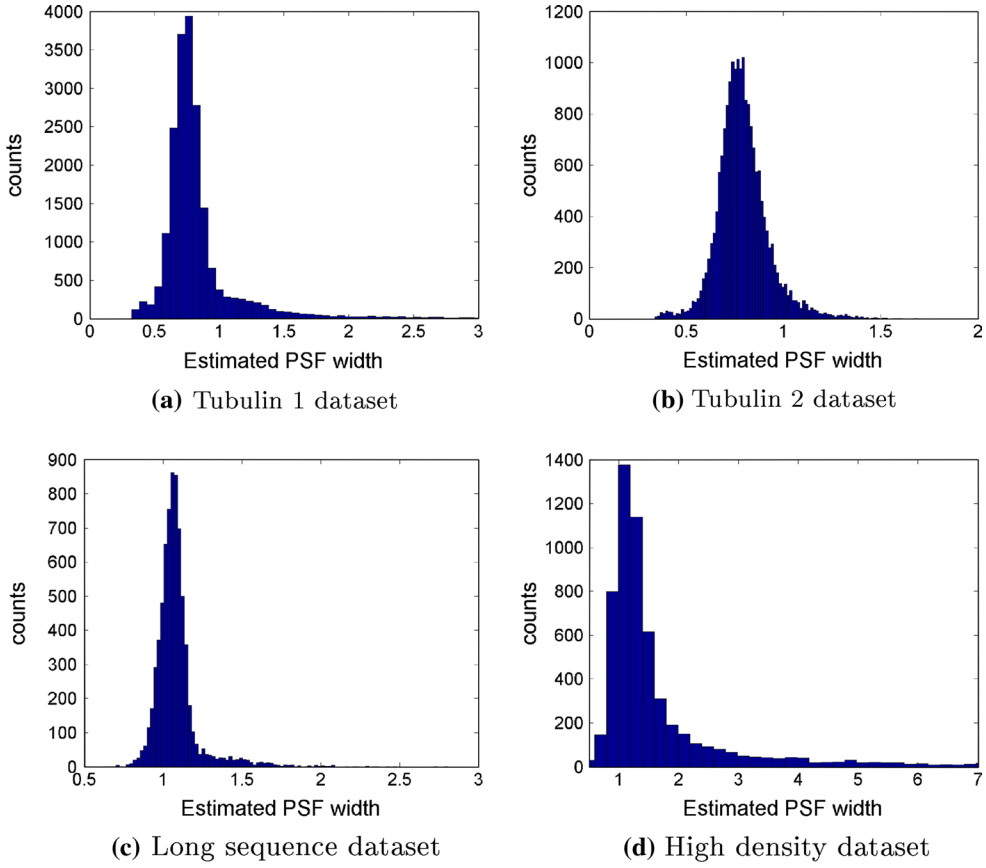

Fig. 5 Histograms of estimated σ_{PSF} for the training datasets from the ISBI localization challenge

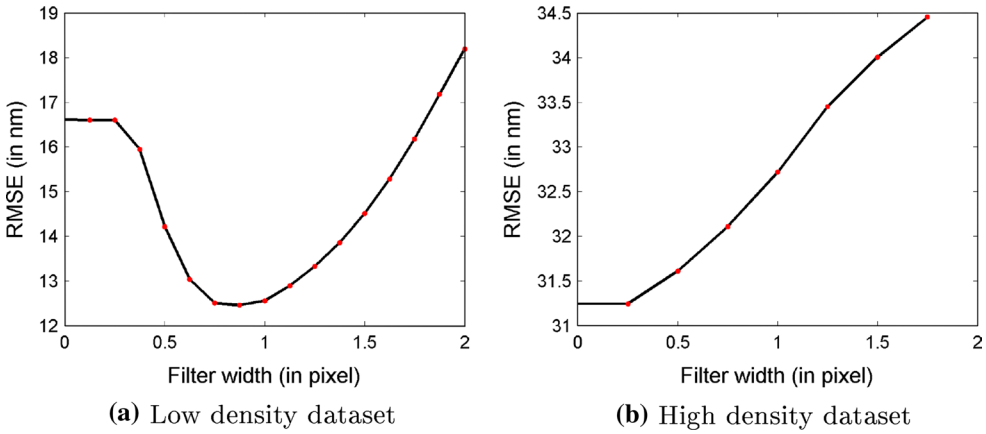

Fig. 6 RMSE for different filter widths

datasets will be referred to as "long sequence" (LS) and "high density" (HD), respectively. Both datasets consist of images with a resolution of 64 px × 64 px at 100 nm per pixel. Full details are available on the challenge web site.

In addition, a free program[5] to compute performance metrics such as Jaccard index, *F* score (both measuring detection reliability) and root mean square localization error (RMSE) accompanies the data.

PSF estimation

We compared a nonparametric (Wiener filter) and two parametric (Gaussian fitting in the Fourier and spatial domains) algorithms for PSF estimation. The three estimation methods performed similarly on the test datasets with non-overlapping spots. This confirms that the Gaussian

[5] http://bigwww.epfl.ch/smlm/evaluation/index.html.

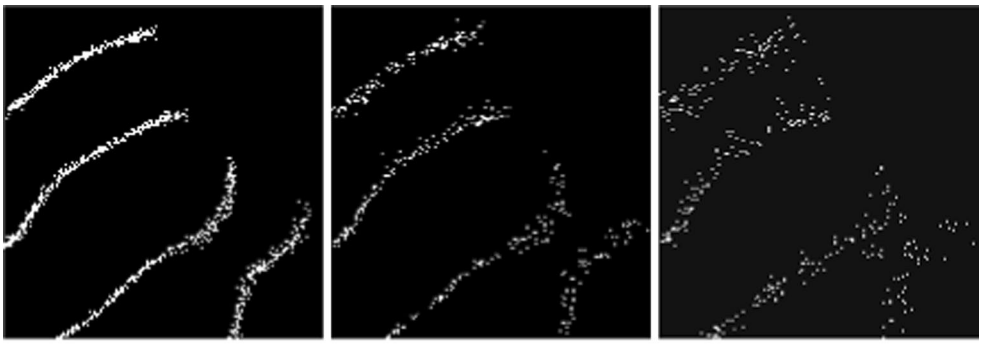

Fig. 7 The effect of filter width on high-density data

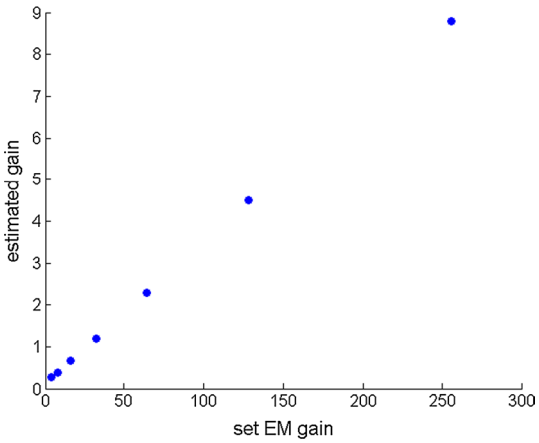

Fig. 8 Scatter plot of estimated gain vs. gain setting in the camera

model is indeed applicable, because otherwise the Wiener filter would have been superior. Spatial domain fitting was slightly more robust than Fourier domain fitting, which tended to overestimate σ_{PSF} and occasionally failed to converge to the correct solution. Since a Gaussian matched filter is faster than a Wiener filter, we prefer the former. Figure 5a–d show the distribution of sigmas estimated by applying the Levenberg-Marquardt algorithm to all spots in the first 2000 frames (respectively all frames in the high-density dataset). The FWHM of the true PSF was 258 nm in all cases, which translates into $\sigma_{\mathrm{true}} = 0.73$ pixels in Tubulin 1 and 2 and into $\sigma_{\mathrm{true}} = 1.09$ pixels in the others. Our estimate of σ_{PSF} is the median of these histograms and within 10 % of the true values, except in the high-density dataset where our estimate is slightly too high because the fitting algorithm falsely merged some overlapping spots.

To improve results in high-density datasets, another advantage of the parametric PSF model comes into effect: It is easy to change the filter strength by choosing $\sigma_{\mathrm{filter}} < \sigma_{\mathrm{PSF}}$. This reduces the tendency of overlapping spots to be erroneously fused. Figure 6 shows the RMSE as a function of σ_{filter} for low- and high-density data. It can be seen that the matched filter indeed minimizes the RMSE when spots do not overlap, whereas the optimal filter size is much smaller when spots overlap. The same effect is illustrated qualitatively in Fig. 7: the detection quality clearly decreases with increasing σ_{filter}. Therefore, we choose a small value for σ_{filter} for high-density data.

Gain estimation

A key feature of SimpleSTORM is the estimation of the camera gain directly from the data. This is important since the actual gain factor may differ from the value indicated by the camera's control settings. For validation purposes, we captured a series of 500 frames of a standard cell culture with phalloidin-labeled actin filaments using different gain settings in the camera. Figure 8 shows that the true and indicated gain values differ, but are proportional to each other. This justifies our linear model (3) and the gain estimation method of section "Noise normalization."

Image standardization

To check whether the image standardization works as desired, we measure the false positive rate of the statistical test according to section "Spot detection" on the image background. Since the ground truth is known, we can define the background precisely as the set of pixels which are at least four pixels away from any true spot position. For the plain statistical test (without connectivity constraint), the false positive rate should equal the p value. To verify this, we counted the fraction of background pixels whose intensity exceeds the threshold according to (7) after image standardization. Figure 9a shows results for Tubulin 1 where the actual error rate is even a bit less than expected. As can be seen in Fig. 9b, the additional connectivity test reduces the error rate further: Out of 46 million background pixels, we found 228 false positives for $p = 1$ % and only 16 for $p = 0.1$ %.

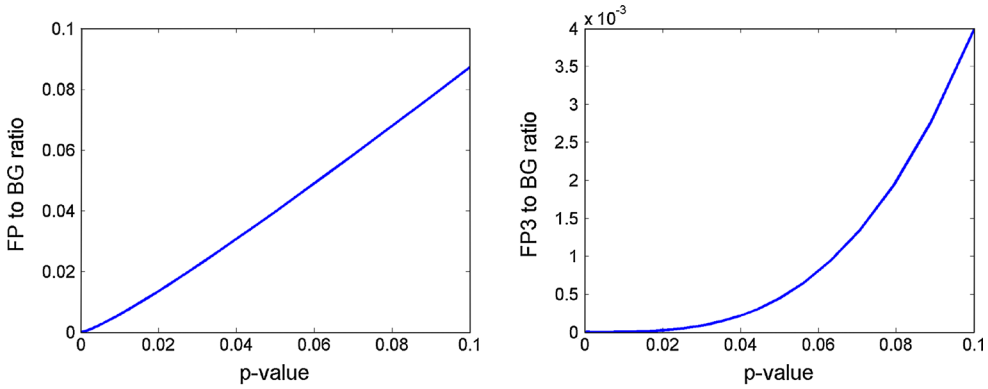

Fig. 9 False positives rates for Tubulin 1 as a function of p

(a) without connectivity test **(b)** with connectivity test

Reconstruction results

We finally report results for the entire SimpleSTORM algorithm on artificial and real data. For the artificial data, we adopt the performance metrics suggested by the ISBI challenge, namely the *Jaccard index*

$$\text{Jaccard} = \frac{\text{TP}}{\text{FN} + \text{TP} + \text{FP}},$$

the F score which is defined via precision and recall as

$$\text{Recall:} \quad \text{R} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Precision:} \quad \text{P} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$F \text{ score:} \quad \text{F} = \frac{2\,\text{R}\,\text{P}}{\text{R} + \text{P}}$$

and the RMSE

$$\text{RMSE} = \left(\frac{1}{\text{TP}} \sum_{i=1}^{\text{TP}} d(\boldsymbol{x}_i^*, \boldsymbol{x}_i)^2 \right)^{1/2}$$

Table 1 SimpleSTORM performance

	Precision	Recall	RMSE (nm)	Frames per sec
Tubulin1	0.938	0.542	16.9	51
Tubulin2	0.912	0.368	20.4	72
LS	0.946	0.802	10.2	563
HD	0.437	0.033	20.5	257

where TP, FN, and FP are the number of true positives, false negatives and false positives respectively, and $d(\boldsymbol{x}_i^*, \boldsymbol{x}_i)$ is the Euclidean distance of the ith detection from the corresponding ground truth location. A true match is recorded if the ground truth position is within radius r of the detection, and each ground truth point can be assigned at most once. We use the evaluation tool from the ISBI challenge website to compute these metrics with $r = 0.3$ px (which corresponds to 30 nm for the high-density and long-sequence dataset and 45 nm for the tubulin1 and tubulin2 datasets).

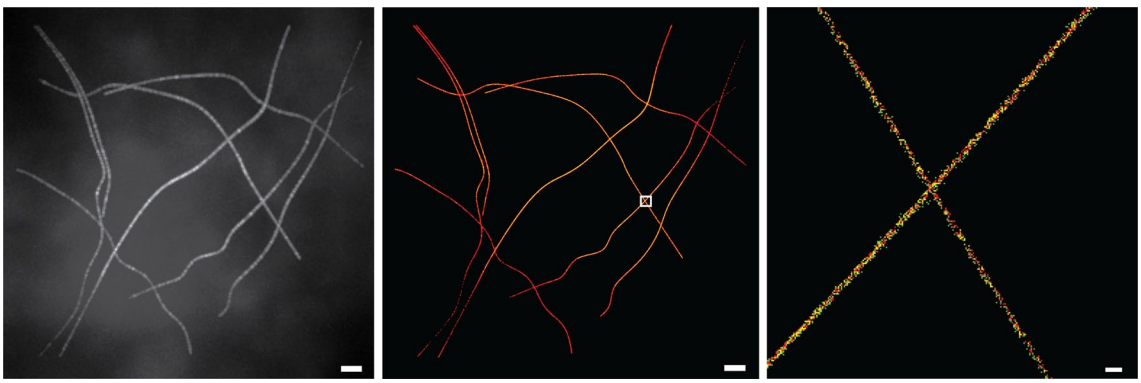

Fig. 10 Tubulin 1. *Left* mean projection of the raw data. *Center* reconstructed image. *Right* localizations (*green*) and ground truth (*red*), *yellow* indicates perfect alignment. *Scalebar* represents 2 μm for the first two images and 100 nm for the third (best viewed in color)

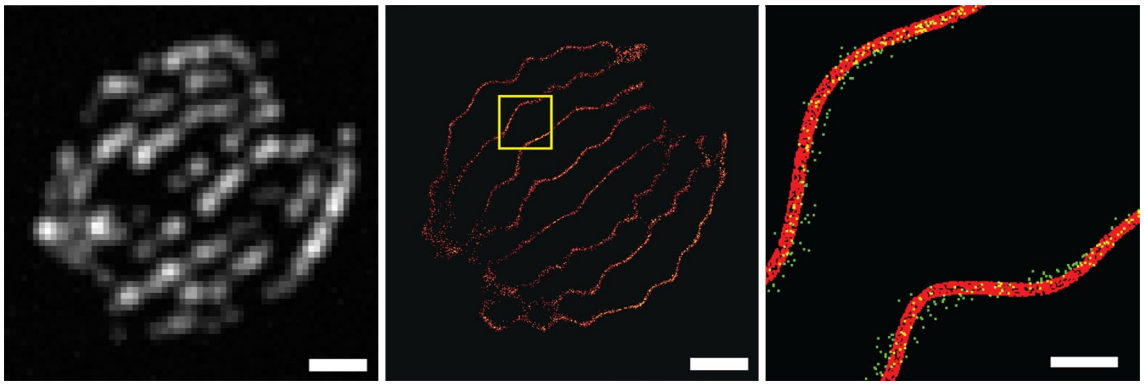

Fig. 11 High-density dataset. *Left* frame 3 of the raw data. *Middle* reconstructed image. *Right* localizations (*green*) and ground truth (*red*). *Scalebars* 1 μm (first two images), 200 nm (third image, best viewed on color)

Table 1 shows the metrics and the processing speed for the four training datasets, where SimpleSTORM was run with standard settings on the first three datasets. That is, gain, offset, and PSF size were determined by self-calibration, the p value was set to 0.3 %, and the reconstructed images were upsampled 16-fold relative to the originals. In the high-density dataset, matched filtering was skipped, and an asymmetry threshold of two was introduced. Corresponding reconstructions are shown in Figs. 10 and 11. The reconstructed images contain both our detections and ground truth. It can be seen in the low-density dataset (Fig. 10 right) that the detections are typically within one pixel of the ground truth position in the reconstructed image. On the difficult high-density dataset, SimpleSTORM still performs reasonably after the indicated manual parameter adjustments. While these adjustments result in a rather low recall, the localization error of the surviving spots is acceptably low. For comparison, Table 2 contains corresponding results for rapidSTORM (version 3.2) whose parameters were manually adjusted for good performance (this takes some time and experience, especially for datasets with variable background). The performance in terms of precision and localization error is comparable, but SimpleSTORM achieved this without parameter tuning. On average, rapidSTORM processes the data 5–6 times faster than SimpleSTORM, but on very small images with low-spot density, the speed difference can be up to a factor of 11. In summary, SimpleSTORM is easier to use, rapidSTORM is faster, and both produce images of comparable quality.

While SimpleSTORM produces good results on "normal" data without parameter tuning, advanced users can still adjust the settings to tailor the output to the needs of a particular experiment. Specifically, it is possible to strengthen the criteria for true positives, so that only points with high certainty and correspondingly better localization accuracy are recorded. This trade-off reduces the detection performance (Jaccard and F-score), but improves the RMSE (it is not possible to improve both at the same time). To this end, SimpleSTORM offers the opportunity to set the p value, an asymmetry threshold and the upscaling factor for the reconstructed image. Table 3 shows the changes in Jaccard index, F score, and RMSE for the 'long-sequence' dataset after these changes. Baseline settings are: p value = 0.1 %, upscaling factor = 10 and asymmetry threshold off.

Lowering the p value does not alter the results significantly, because the false positive rate is already very low. Applying an asymmetry threshold improves the RMSE, but discards a number of true positives along with the undesirable distorted spots. A higher upsampling factor increases the localization accuracy without removing any points, but increases the reconstruction time.

Table 2 rapidSTORM performance

	Precision	Recall	RMSE (nm)	Frames per sec
Tubulin1	0.890	0.537	16.9	333
Tubulin2	0.901	0.378	20.2	387
LS	0.935	0.797	9.63	6000
HD	0.467	0.041	20.3	256

Table 3 Effect of advanced parameter settings

	Jaccard	F score	RMSE (nm)
Baseline	0.824	0.903	12.88
p value = 0.001 %	0.815	0.898	12.76
Asymmetry (1.5)	0.620	0.765	12.13
Upscale factor = 100	0.619	0.764	11.45

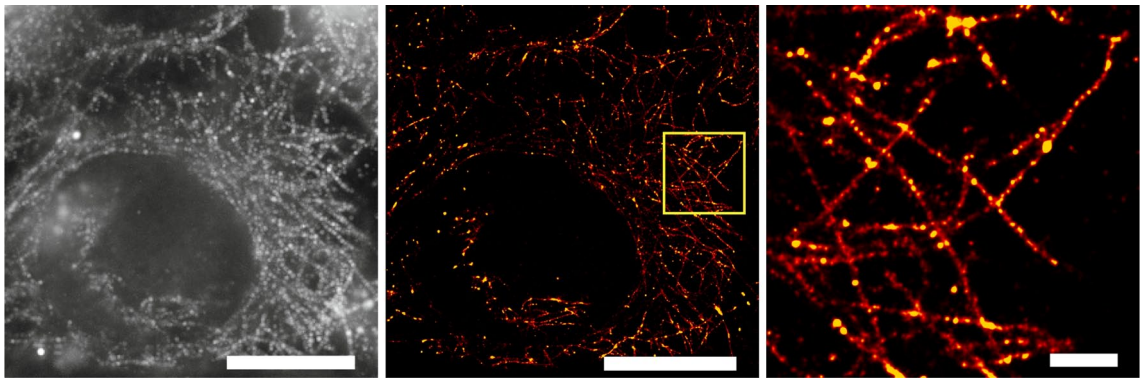

Fig. 12 Micro-tubulin labeled with AlexaFluor647 in HEK cells. *Left* mean image of the raw data. *Middle* reconstruction by SimpleSTORM. *Right* magnification of the indicated subregion. *Scalebars* 10 μm (first two images), 1.5 μm

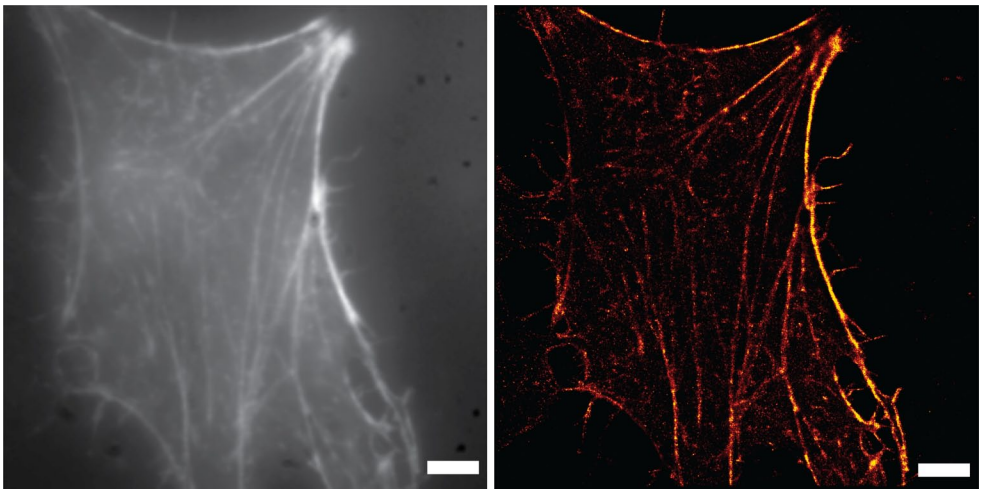

Fig. 13 Actin labeled with mEos2 in HeLa cells. *Left* mean image of the raw data (data courtesy of M. Heilemann). *Right* reconstruction by SimpleSTORM. *Scalebars* represent 3 μm

Experimental data

Finally, we applied SimpleSTORM to experimental single-molecule localization data recorded from cellular structures. The images we processed were recorded either using the principle of direct stochastic optical reconstruction microscopy [dSTORM, Heilemann et al. (2008), Figs. 1 and 12], or photoactivated-localization microscopy [PALM, Betzig et al. (2006), Fig. 13]. All data were recorded with an exposure time of 100 ms and irradiation intensities between 1 and 4 kW/cm^2 (for experimental protocols, the reader is referred to Heilemann et al. (2009) (dSTORM, ATTO520), Heilemann et al. (2008) (dSTORM, Alexa Fluor 647) and Muranyi et al. (2013) (PALM, mEos2). HEK cells for Fig. 12 were fixed with methanol at −20 °C for 5 min and left over night in PBS. Fixed cells were blocked with 5 % horse serum in PBS and incubated with the primary antibody against alpha-tubulin (Sigma, T6199) for 1h. Cells were washed three times for 5 min with blocking buffer (5 % horse serum in PBS). Afterward, cells were incubated with the secondary antibody (Invitrogen catalog no. A-21236) against mouse coupled to the fluorophore Alexa 647 for 1 h. Again, cells were washed three times 5 min with blocking buffer. With 4 % PFA, cells were post-fixed for 10 min before putting the cells in a 0.1 % poly-L-lysine solution for 4 h. 100 mM MEA solution and Tetraspeck (LifeTechnologies) beads with a concentration of 1:200 were used as imaging buffer.

Reconstructed super-resolution images are shown for tubulin labeled with ATTO520 (Fig. 1) or Alexa Fluor 647 (Fig. 12), as well as for actin labeled with mEos2 (Fig. 13). We further estimate the spatial resolution by measuring the width of tubulin filaments, which were found to be about 60 nm (Fig. 1) and comparable to previously published

values [Heilemann et al. (2008)]. Considering the expected RMSE and the average distance of the fluorescent markers from the actual molecule, this is in the expected range.

Acknowledgments This research was supported by contract research "Methoden für die Lebenswissenschaften" of the Baden-Württemberg Stiftung. We are grateful to Mike Heilemann, Varun Venkataramani and Benjamin Flottmann for providing the raw data of the images presented in this paper, as well as for giving many helpful comments on the algorithm. We also thank the organizers of the ISBI Localization Microscopy Challenge for the permission to use their artificial data.

References

Abraham AV, Ram S, Chao J, Ward ES, Ober RJ (2009) Quantitative study of single molecule location estimation techniques. Opt Express 17(26):23,352

Andersson SB (2008) Localization of a fluorescent source without numerical fitting. Opt Express 16(23):18,714–18,724

Anscombe FJ (1948) The transformation of Poisson, binomial and negative-binomial data. Biometrika 35(3/4):246–254

Baddeley D (2012) PYME—the python localization microscopy environment. http://code.google.com/p/python-microscopy/. Accessed: 12 Feb. 2014

Bates M, Huang B, Zhuang X (2008) Super-resolution microscopy by nanoscale localization of photo-switchable fluorescent probes. Curr Opin Chem Biol 12(5):505–514

Betzig E, Patterson GH, Sougrat R, Lindwasser OW, Olenych S, Bonifacino JS, Davidson MW, Lippincott-Schwartz J, Hess HF (2006) Imaging intracellular fluorescent proteins at nanometer resolution. Science 313(5793):1642–1645

Boulanger J, Kervrann C, Bouthemy P, Elbau P, Sibarita JB, Salamero J (2010) Patch-based nonlocal functional for denoising fluorescence microscopy image sequences. IEEE Trans Med Imaging 29(2):442–454

Brede N, Lakadamyali M (2012) GraspJ: an open source, real-time analysis package for super-resolution imaging. Optl Nanoscopy 1(1):1–7

Egner A, Geisler C, von Middendorff C, Bock H, Wenzel D, Medda R, Andresen M, Stiel AC, Jakobs S, Eggeling C et al (2007) Fluorescence nanoscopy in whole cells by asynchronous localization of photoswitching emitters. Biophys J 93(9):3285–3290

Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395

Grüll F, Kirchgessner M, Kaufmann R, Hausmann M, Kebschull U (2011) Accelerating image analysis for localization microscopy with FPGAs. In: Field Programmable Logic and Applications (FPL), 2011 International Conference, pp 1–5

Hedde PN, Fuchs J, Oswald F, Wiedenmann J, Nienhaus GU (2009) Online image analysis software for photoactivation localization microscopy. Nat Methods 6:689–690

Heilemann M (2010) Fluorescence microscopy beyond the diffraction limit. J Biotechnol 149(4):243–251

Heilemann M, van de Linde S, Schüttpelz M, Kasper R, Seefeldt B, Mukherjee A, Tinnefeld P, Sauer M (2008) Subdiffraction-resolution fluorescence imaging with conventional fluorescent probes. Angewandte Chemie Int Ed 47(33):6172–6176

Heilemann M, van de Linde S, Mukherjee A, Sauer M (2009) Super-resolution imaging with small organic fluorophores. Angewandte Chemie Int Ed 48(37):6903–6908

Henriques R, Mhlanga MM (2009) PALM and STORM: what hides beyond the Rayleigh limit? Biotechnol J 4(6):846–857

Henriques R, Lelek M, Fornasiero EF, Valtorta F, Zimmer C, Mhlanga MM (2010) QuickPALM: 3D real-time photoactivation nanoscopy image processing in ImageJ. Nat Methods 7(5):339–340

Herbert A (2012) PeakFit ImageJ plugins for single-molecule light microscopy. http://www.sussex.ac.uk/gdsc/intranet/microscopy/imagej/smlm_plugins. Accessed: 12 Feb. 2014

Högbom JA (1974) Aperture synthesis with a non-regular distribution of interferometer baselines. Astron Astrophys Suppl 15:417

Holden S, Uphoff S, Kapanidis A (2011) DAOSTORM: an algorithm for high-density super-resolution microscopy. Nat Methods 8(4):279

Huang B (2010) Super-resolution optical microscopy: multiple choices. Curr Opin Chem Biol 14(1):10–14

Huang F, Schwartz SL, Byars JM, Lidke KA (2011) Simultaneous multiple-emitter fitting for single molecule super-resolution imaging. Biomed Opt Express 2(5):1377–1393

Izeddin I, Boulanger J, Racine V, Specht C, Kechkar A, Nair D, Triller A, Choquet D, Dahan M, Sibarita J (2012) Wavelet analysis for single molecule localization microscopy. Opt Express 20(3):2081–2095

Kim K, Min J, Carlini L, Unser M, Manley S, Jeon D, Ye J (2013) Fast maximum likelihood high-density low-SNR super-resolution localization microscopy. In: Proceedings 10th International Workshop Sampling Theory and Applications (SampTA'13), pp 285–288

Křížek P, Raška I, Hagen GM (2011) Minimizing detection errors in single molecule localization microscopy. Opt Express 19(4):3226–3235

Lidke K, Rieger B, Lidke D, Jovin T (2005) The role of photon statistics in fluorescence anisotropy imaging. IEEE Trans Image Process 14(9):1237–1245

Min J, Vonesch C, Olivier N, Kirshner H, Manley S, Ye J, Unser M (2013) Continuous localization using sparsity constraints for high-density super-resolution microscopy. In: Proceedings ISBI'13, pp 181–184

Muranyi W, Malkusch S, Müller B, Heilemann M, Kräusslich HG (2013) Super-resolution microscopy reveals specific recruitment of hiv-1 envelope proteins to viral assembly sites dependent on the envelope c-terminal tail. PLoS Pathog 9(2):9(2):e198–e1003

Smith CS, Joseph N, Rieger B, Lidke KA (2010) Fast, single-molecule localization that achieves theoretically minimum uncertainty. Nat Methods 7(5):373–375

Stallinga S, Rieger B (2012) Position and orientation estimation of fixed dipole emitters using an effective Hermite point spread function model. Opt Express 20(6):5896–5921

Stetson PB (1987), DAOPHOT: a computer program for crowded-field stellar photometry. Publications of the Astronomical Society of the Pacific, pp 191–222

Stuurman N (2012) Localization Microscopy MicroManager Plugin. http://micro-manager.org/wiki/Localization_Microscopy. Accessed: 12 Feb. 2014

Thompson RE, Larson DR, Webb WW (2002) Precise nanometer localization analysis for individual fluorescent probes. Biophys J 82(5):2775–2783

Turin G (1960) An introduction to matched filters. IRE Trans Inf Theory 6(3):311–329

Wolter S, Schüttpelz M, Tscherepanow M, van de Linde S, Heilemann M, Sauer M (2010) Real-time computation of subdiffraction-resolution fluorescence images. J Microsc 237(1):12–22

Yu J (2011) Octane ImageJ plugin for super-resolution imaging and single molecule tracking. https://github.com/jiyuuchc/Octane. Accessed: 12 Feb. 2014

Zhu L, Zhang W, Elnatan D, Huang B (2012) Faster STORM using compressed sensing. Nat Methods 9(7):721–723