

# SCIENTIFIC REPORTS

OPEN

## Analyzing Single Molecule Localization Microscopy Data Using Cubic Splines

Hazen P. Babcock<sup>1</sup> & Xiaowei Zhuang<sup>1,2</sup>

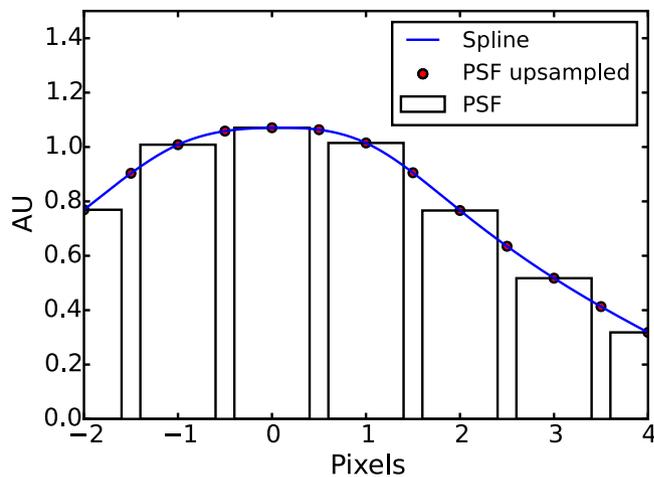
The resolution of super-resolution microscopy based on single molecule localization is in part determined by the accuracy of the localization algorithm. In most published approaches to date this localization is done by fitting an analytical function that approximates the point spread function (PSF) of the microscope. However, particularly for localization in 3D, analytical functions such as a Gaussian, which are computationally inexpensive, may not accurately capture the PSF shape leading to reduced fitting accuracy. On the other hand, analytical functions that can accurately capture the PSF shape, such as those based on pupil functions, can be computationally expensive. Here we investigate the use of cubic splines as an alternative fitting approach. We demonstrate that cubic splines can capture the shape of any PSF with high accuracy and that they can be used for fitting the PSF with only a 2–3x increase in computation time as compared to Gaussian fitting. We provide an open-source software package that measures the PSF of any microscope and uses the measured PSF to perform 3D single molecule localization microscopy analysis with reasonable accuracy and speed.

The accuracy of the fitting algorithm is an important factor in the final resolution of single molecule localization based super-resolution microscopy. This accuracy is in turn determined in part by how well the fitting algorithm models the PSF of the microscope. This is particularly important for 3D super-resolution imaging, where the PSF shapes are often not a simple Gaussian function<sup>1–4</sup>. Approaches to accurately modeling the PSF for the purpose of localizing single molecules include PSF correlation<sup>5–7</sup> and pupil function fitting<sup>8</sup>. In the PSF correlation approach, the PSF of the microscope is measured by scanning the objective (or the sample) in Z to capture the shape of the PSF at multiple focal planes. The resulting z stack of PSFs can then be cross-correlated with the image of a single molecule to determine its position. This is computationally expensive as it requires multiple 2D Fourier transforms, followed by further fitting if sub-pixel position information is desired. Alternatively, the molecule's position can be determined directly by providing an initial estimate for the molecule's position and computing the fitting error between the molecule's image and the PSF, then appropriately updating the position estimate. This can be done using a simplex optimization algorithm such as Nelder-Mead<sup>9</sup>. However this class of algorithms, which are used when the derivatives of the error function are not known or are hard to calculate, converge slowly on the optimal solution. An additional complication of this latter approach is the need to interpolate between the measured PSF values in x, y and z.

These problems are largely avoided with a pupil function fitting approach<sup>8, 10, 11</sup>, as interpolation is not necessary and the derivatives of the error function can be calculated, enabling the use of more rapidly converging algorithms such as those based on the Newton-Raphson algorithm. In the pupil function approach, the PSF is again measured by scanning the sample containing point emitters (such as beads) in Z, passing through the focal plane of the objective. The measured PSF is then fit to determine the pupil function of the microscope. As the pupil function captures the PSF of the microscope with high accuracy, fitting the images of single emitters to the pupil function allows the positions of these emitters to be determined with high accuracy. However calculating the PSF and its derivatives with respect to the fitting parameters at a particular x, y and z location from the pupil function is computationally expensive, and could be slow unless GPU acceleration is used<sup>8</sup>.

Another approach to accurately model the PSF for the purpose of localizing single molecules is to use splines. The most commonly used splines approximate the function of interest using one or more polynomials. This class

<sup>1</sup>Center for Advanced Imaging, Harvard University, Cambridge, MA, 02138, USA. <sup>2</sup>Howard Hughes Medical Institute, Department of Chemistry and Chemical Biology and Department of Physics, Harvard University Cambridge, Cambridge, MA, 02138, USA. Correspondence and requests for materials should be addressed to H.P.B. (email: [hbabcock@fas.harvard.edu](mailto:hbabcock@fas.harvard.edu))



**Figure 1.** 1D Spline example. Each cubic spline interval (the blue line) is determined such that itself and its first derivative are continuous across the red points which are the interval boundaries. The red points are calculated by up-sampling the measured PSF (grey bars) by a factor of two using 3<sup>rd</sup> order spline interpolation.

of splines can be calculated rapidly, and it is also easy to compute their derivatives. 2D cubic splines were used in the DAOPHOT astronomy package to provide higher order corrections to a 2D Gaussian for the purpose of fitting the locations and magnitudes of stars<sup>12</sup>, or single molecules<sup>13</sup>, the latter using the DAOSTORM algorithm, the adaptation of DAOPHOT for single molecule localization microscopy. B-splines, of which cubic splines are a sub-set, were used previously in localization microscopy to estimate the maximal localization accuracy in x, y and z of an arbitrarily shaped 3D point spread function (PSF)<sup>14</sup>. B-splines were also used in a general method for 3D localization fitting similar to what is presented here<sup>15</sup>.

This current work builds on previous work by further exploring the use of splines for rapid and accurate analysis of 3D super-resolution microscopy data based on single molecule localization. In particular, we provide an open-source maximum likelihood estimation (MLE) based cubic spline analysis package, side-by-side comparisons of its performance with elliptical Gaussian fitting, and a demonstration of its broad application by fitting simulated 3D super-resolution microscopy data derived from several different PSFs.

## Results

A cubic spline approximates a one dimensional function using piecewise 3<sup>rd</sup> order polynomials of the form:

$$f_i(x) = a_i \left( \frac{x - t_i}{\Delta t} \right)^3 + b_i \left( \frac{x - t_i}{\Delta t} \right)^2 + c_i \left( \frac{x - t_i}{\Delta t} \right) + d_i (t_i \leq x \leq t_{i+1}, \Delta t = t_{i+1} - t_i) \quad (1)$$

where each  $f_i$  is valid on small equal length intervals  $[t_i, t_{i+1}]$ . Any function with a continuous first derivative can be approximated with arbitrary accuracy by choosing an appropriate interval size. Here we follow DAOPHOT/DAOSTORM and choose an interval size that is  $\frac{1}{2}$  of the image pixel size as a good compromise between accurately modeling the PSF and minimizing the total number of intervals and coefficients.

In Fig. 1 we provide a simple 1D example of the cubic spline construction process that we use in this work. We start with a hypothetical measurement of the PSF (gray bars in Fig. 1). Next the measured PSF is up-sampled by 2x using 3<sup>rd</sup> order spline interpolation (red points in Fig. 1). Finally the coefficients  $a_i, b_i, c_i, d_i$  of the spline (blue line in Fig. 1) for each interval are calculated with the constraint that the polynomials  $f_i(x)$  and their first derivatives  $f_i'(x)/dx$  are continuous at the red points, the interval boundaries. We chose to use the additional constraint that the second derivative of the spline is zero at the start of the first interval and at the end of the last interval. We use the approach described in ref. 16 to solve for the coefficients of the (natural) cubic spline.

We used the following procedure to measure the PSF in three dimensions. We placed 4x up-sampled PSFs with high photon counts on a xy grid. Then we scanned the z values of the PSFs to create z scan image stacks. Finally, we downsampled the image stacks by 4x in x and y to create the final PSF measurement image stack. Then, using the known emitter locations, images of isolated emitters were cropped out of these image stacks and up-sampled by 2x in xy using the `ndimage.interpolation.zoom()` function from the Scipy Python package<sup>17</sup> with 3<sup>rd</sup> order spline interpolation. Next, the up-sampled images of these emitters were aligned to their centroid positions using the `ndimage.interpolation.shift()` function and averaged to obtain a single image for each z position. Finally, images covering 50 nm z ranges were further averaged together to generate the PSF for the z position that is at the center of the 50 nm range and the average PSFs determined from a series 50 nm intervals were used to create a 3D array that represents the 3D PSF. To measure the 3D PSF in real experiments, small fluorescent beads can be attached to a coverslip. Then z scan movies can be taken with a piezo positioner. In this situation the true emitter locations are not known, but they can be estimated with sufficient precision for this purpose by using another fitting algorithm such as 3D-DAOSTORM<sup>18</sup> because the fluorescent beads typical emit a very large number of photons.

The extension of the cubic spline construction to 3 dimensions is described below. In three dimensions, the cubic spline is:

$$f_{i,j,k}(x, y, z) = \sum_{m=0}^3 \sum_{n=0}^3 \sum_{o=0}^3 a_{i,j,k,m,n,o} \left( \frac{x-t_i}{\Delta t} \right)^m \left( \frac{y-t_j}{\Delta t} \right)^n \left( \frac{z-u_k}{\Delta u} \right)^o \quad (2)$$

$$(t_i \leq x \leq t_{i+1}, t_j \leq y \leq t_{j+1}, u_k \leq z \leq u_{k+1}, \Delta t = t_{i+1} - t_i, \Delta u = u_{k+1} - u_k)$$

where each  $f_{i,j,k}$  is valid on  $\frac{1}{2}$  pixel length intervals  $[t_i, t_{i+1}]$  in  $x$ ,  $[t_j, t_{j+1}]$  in  $y$  and  $[u_k, u_{k+1}]$  intervals in  $z$ . We start with a 3D PSF that is determined by either simulation or experiments, and then up-sampled such that it has an equal number of elements in  $x$ ,  $y$  and  $z$ . This is not strictly necessary but having an equal number of elements for each axis simplifies later computations. We then use a series of 1D cubic spline interpolations in  $x$ ,  $y$  and  $z$  to further upsample the PSF, by 4 fold in each dimension, to obtain 64 values of the PSF within each element. These 64 values then allow us to uniquely specify the 3D cubic spline  $f_{i,j,k}(x, y, z)$  and determine its 64 coefficients  $a_{i,j,k,m,n,o}$  in the intervals that correspond to each element of the PSF.

We then use this cubic spline approximation of the 3D PSF to fit the image of individual emitters and determine their 3D coordinates. Based on the 3D cubic spline formula it can be seen that the evaluation of  $f_{i,j,k}(x, y, z)$  at a single point will require 128 operations, as compared to 10 operations to evaluate the elliptical Gaussian that is often used for 3D PSF fitting:

$$f(x, y, z) = a \exp \left( -\frac{(x-x_0)^2}{w_x(z)} - \frac{(y-y_0)^2}{w_y(z)} \right) + b \quad (3)$$

Here we've ignored the calculation of the  $x^m y^n z^o$  terms as for fitting the evaluation is always done at the same point in each of the spline intervals. Similarly, we have ignored the calculation of the  $w_x(z)$  and the  $w_y(z)$  terms in the Gaussian as these are also the same for each point in the PSF. We wrote a simple C program to time how long it took to perform these calculations, and found that the cubic spline calculation is approximately 2.3x slower. However, we note that this is a best case estimate as the cubic spline calculation is more memory intensive and thus is likely to be somewhat slower in general. However, this rough estimate suggests that the computational expense of the cubic spline fitting will be similar to that of Gaussian fitting.

The highest accuracy estimation of the parameters of the localization of an emitter is typically achieved using non-linear least squares or maximum likelihood estimation (MLE) based fitting. This can be done efficiently if both the error function and its derivatives are easily calculated for any point in the PSF. As described in ref. 19, the error function for MLE fitting is:

$$\chi_{mle}^2 = 2 \sum_{i=1}^n f_i - x_i - 2 \sum_{i=1, x_i \neq 0}^n x_i \ln(f_i/x_i) \quad (4)$$

And its derivatives with respect to the fitting parameters ( $p$ ) are:

$$\frac{\partial \chi_{mle}^2}{\partial p_k} = 2 \sum_{i=1}^n \left( 1 - \frac{x_i}{f_i} \right) \frac{\partial f_i}{\partial p_k} \quad (5)$$

$$\frac{\partial^2 \chi_{mle}^2}{\partial p_k \partial p_l} = 2 \sum_{i=1}^n \frac{\partial f_i}{\partial p_k} \frac{\partial f_i}{\partial p_l} \frac{x_i}{f_i^2} \quad (6)$$

where we have followed ref. 19 and ignored the second derivative terms in the calculation of equation (6).

The cubic spline PSF and its derivatives with respect to the fitting parameters  $x_c, y_c, z_c$  (the 3D coordinates of the emitter),  $h$  (the height of the PSF), and  $b$  (the baseline of the PSF) are easy to calculate, and they are:

$$f_{i,j,k}(x, y) = h \sum_{m=0}^3 \sum_{n=0}^3 \sum_{o=0}^3 a_{i,j,k,m,n,o} \left( \frac{(x-x_c)-t_i}{\Delta t} \right)^m \left( \frac{(y-y_c)-t_j}{\Delta t} \right)^n \left( \frac{z_c-u_k}{\Delta u} \right)^o + b \quad (7)$$

$$\frac{\partial f_{i,j,k}(x, y)}{\partial h} = \sum_{m=0}^3 \sum_{n=0}^3 \sum_{o=0}^3 a_{i,j,k,m,n,o} \left( \frac{(x-x_c)-t_i}{\Delta t} \right)^m \left( \frac{(y-y_c)-t_j}{\Delta t} \right)^n \left( \frac{z_c-u_k}{\Delta u} \right)^o \quad (8)$$

$$\frac{\partial f_{i,j,k}(x, y)}{\partial x_c} = -h \sum_{m=0}^2 \sum_{n=0}^3 \sum_{o=0}^3 \frac{(m+1)}{\Delta t} a_{i,j,k,m,n,o} \left( \frac{(x-x_c)-t_i}{\Delta t} \right)^m \left( \frac{(y-y_c)-t_j}{\Delta t} \right)^n \left( \frac{z_c-u_k}{\Delta u} \right)^o \quad (9)$$

$$\frac{\partial f_{i,j,k}(x, y)}{\partial y_c} = -h \sum_{m=0}^3 \sum_{n=0}^2 \sum_{o=0}^3 \frac{(n+1)}{\Delta t} a_{i,j,k,m,n,o} \left( \frac{(x-x_c)-t_i}{\Delta t} \right)^m \left( \frac{(y-y_c)-t_j}{\Delta t} \right)^n \left( \frac{z_c-u_k}{\Delta u} \right)^o \quad (10)$$

$$\frac{\partial f_{i,j,k}(x, y)}{\partial z_c} = h \sum_{m=0}^3 \sum_{n=0}^3 \sum_{o=0}^2 \frac{(o+1)}{\Delta u} a_{i,j,k,m,n,o} \left( \frac{(x-x_c)-t_i}{\Delta t} \right)^m \left( \frac{(y-y_c)-t_j}{\Delta t} \right)^n \left( \frac{z_c-u_k}{\Delta u} \right)^o \quad (11)$$

$$\frac{\partial f_{i,j,k}(x, y)}{\partial b} = 1 \quad (12)$$

We minimize  $\chi_{mle}^2$  using the above equations and the approach described in ref. 19, which is a modified version of the Levenberg-Marquadt algorithm<sup>20</sup>. Given a reasonable starting point the optimal values for  $x_c, y_c, z_c, h$  and  $b$  can be found with high accuracy in just a few (~10) iterative updates for isolated emitters. The algorithm executes the following pseudo-code on each frame of a single-molecule imaging movie until either no new peaks are found, or the maximum number of iterations is reached.

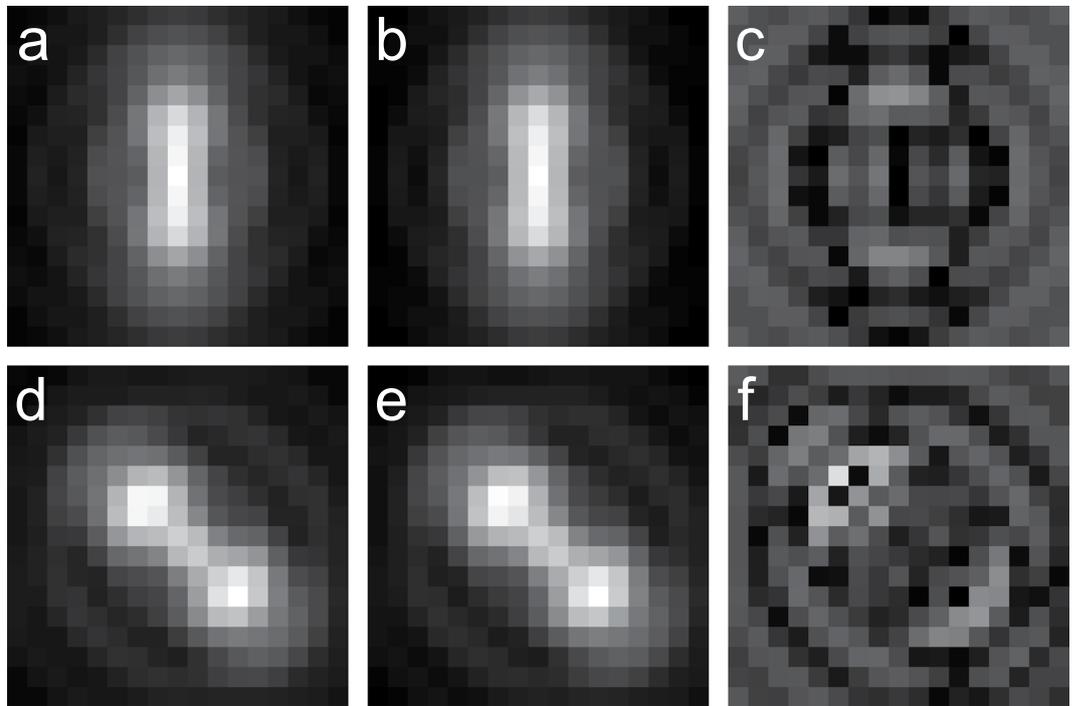
Algorithm:

- 1: Load the current image.
- 2: Perform an initial estimate of the image background.
- 3: **while not** all converged **and** repeats < max repeats **do**
- 4:     Subtract current background estimate from the image.
- 5:     Find localizations in the background subtracted image and add these to the list of all localizations.
- 6:     Subtract localizations from the image.
- 7:     **while** iterations < max iterations **and not** all converged **do**
- 8:         **for each** localization **do**
- 9:             **if not** converged **then**
- 10:                 **Add localization to image**
- 11:                 Levenberg-Marquadt update of localization fitting parameters
- 12:                 Subtract updated localization from image
- 13:                 **if** converged **then**
- 14:                     mark localization as converged
- 15:             **for each** localization **do**
- 16:                 **if** distance to a brighter neighbor < minimum distance **then**
- 17:                     discard localization
- 18:                     mark all neighbors as unconverged
- 19:             Subtract localizations from the image.
- 20:     Estimate background of the localization subtracted image.
- 21: Save fit parameters of all the localizations.

Note that this algorithm ignores cross-terms for overlapping localizations. Each localization's fitting parameters are updated in a context where all the other localizations are fixed. This increases the simplicity of the algorithm implementation at the price of a reduced convergence speed for overlapping localizations.

This approach is very similar to that used by 3D-DAOSTORM<sup>18</sup>, differing only at the peak finding and peak fitting steps. In order to do peak finding the image is convolved with the spline representation of the PSF at one or more  $z$  values, then local maxima in the convolved images are used as the initial locations for the subsequent peak fitting step. We found this approach to be more robust, particularly with PSFs that do not have a single local maxima, such as the double helix PSF<sup>2</sup>, than using local maxima in the original image. When we performed convolutions at multiple  $z$  values each of the convolved images was treated as a single plane in a  $z$  stack. Then local maxima were found in the  $z$  stack and peak fitting was started at the  $x, y$  and  $z$  values of the local maxima. In our experience the use of multiple  $z$  values was only helpful for analyzing data taken with the double helix PSF<sup>2</sup>, probably because among the PSFs that we tested, the double helix PSF is the only one that is not degenerate in  $z$ . By this we mean the PSF shape at one  $z$  value cannot be approximately recreated by a linear sum of PSF shapes at other  $z$  values. This is in contrast with other PSFs such as the astigmatic PSF whose shape at large  $z$  offset values can be approximated quite well by a sum of multiple  $z=0$  PSFs with differing  $xy$  displacements.

To test how well a cubic spline with  $\frac{1}{2}$  pixel interval size in  $x$  and  $y$  can capture the shape of non-trivial PSFs, we used the pupil function approach<sup>10,11</sup> to generate 3D PSFs for a purely astigmatic PSF (Zernike mode  $z_{22} = 1.3$ ) and for the saddle-point PSF described in ref. 3, both of which have been previously used for 3D super-resolution imaging<sup>1,3</sup>. For each condition we generated noise free images of the PSF at  $z$  positions ranging from  $-1 \mu\text{m}$  to  $1 \mu\text{m}$  in  $10 \text{ nm}$  steps. Then we determined the coefficients of the cubic spline for this PSF as described above. We found that the maximum pixel wise amplitude difference between the astigmatic PSF and the cubic spline was 3.3% (at  $z=0 \text{ nm}$ ) over the entire  $2 \mu\text{m}$   $z$  range (Fig. 2). Similarly, we found for the saddle-point PSF that the maximum pixel wise difference was 3.2% (at  $z=40 \text{ nm}$ ). We also tested how well an elliptical Gaussian was able to capture the shape of a purely astigmatic PSF, as an elliptical Gaussian is the analytical function typically used



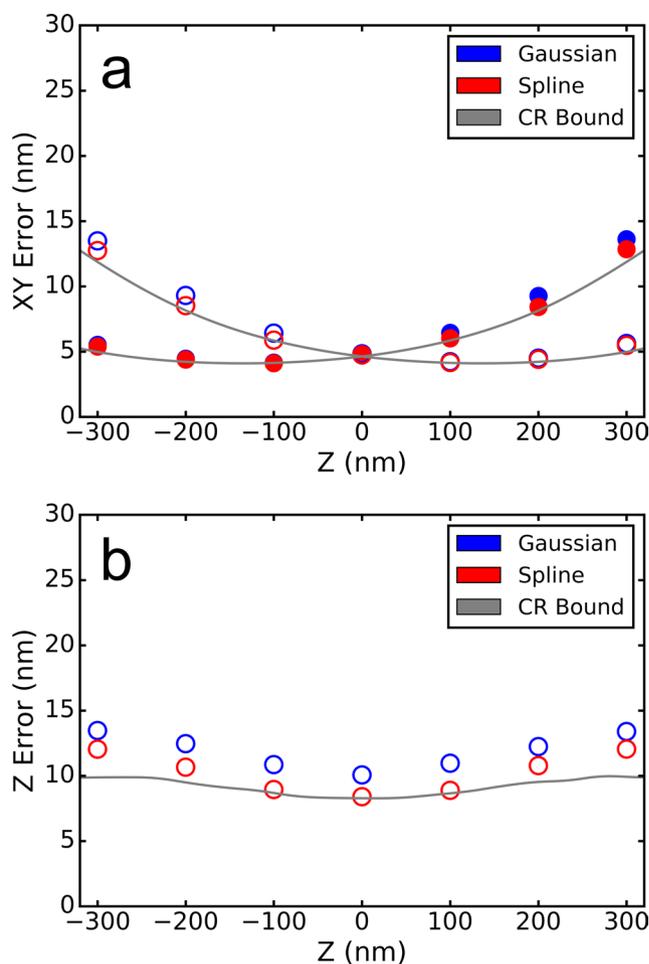
**Figure 2.** Analytical PSFs and their spline representations for the astigmatic PSF and saddle-point PSF. (a) Astigmatic PSF at  $z = -420$  nm. (b) Cubic spline representation of (a). (c) Difference between (a) and (b) multiplied by 100. (d) Saddle-point PSF at  $z = -420$  nm. (e) Cubic spline representation of (d). (f) Difference between (d) and (e) multiplied by 100.

in astigmatism-based 3D super-resolution imaging. We calculated the best fit elliptical Gaussian to the astigmatic PSF for each  $z$  position in the same 2  $\mu\text{m}$   $z$  range and found that the error was somewhat larger than for cubic spline fitting, with a maximum pixel wise difference of 7.7% (at  $z = 60$  nm).

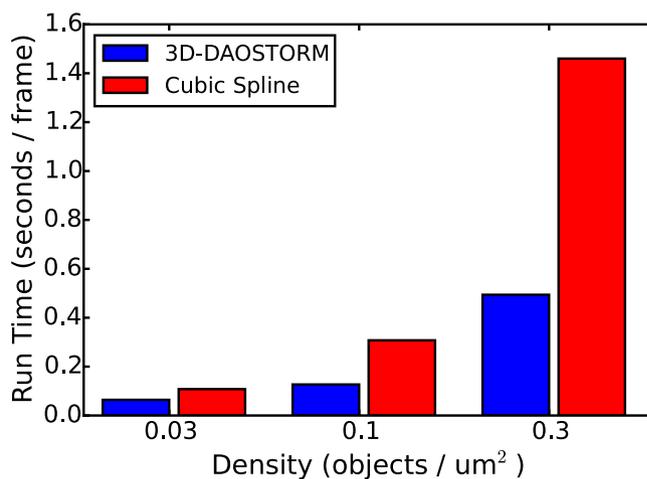
Next we investigated whether the fact that the cubic spline more accurately captured the shape of a purely astigmatic PSF resulted in improved fitting performance. As described above we first create a high photon count  $z$  image stack for measuring the PSF, then we create a lower photon count  $z$  image stack for measuring the fitting performance. In addition we note that, in this work, we only test the performance of this cubic spline approach on the localization precision of the emitters, but do not test the performance of this approach on the peak finding step here; in fact, the initial values for peak fitting are chosen as the known locations of the peaks used in the simulation. To measure the PSF, we created a 4x up-sampled representation of an astigmatic PSF using the pupil function approach<sup>10,11</sup>. We used this up-sampled PSF to create high photon count 2D images by placing multiple well separated copies of it on a  $xy$  grid. To make PSF measurement movies we scanned the  $z$  position of the PSF with a 10 nm interval size and down-sampled the resulting 2D images by 4x to generate the final images. We then used these calibration movies to determine the coefficients of the cubic spline representation of the PSF and also to determine the  $w_x(z)$  and  $w_y(z)$  calibration curve for elliptical Gaussian fitting. Then, to measure the fitting performance, a test movie was created using the same approach but with 4000 photons per localization and a constant background of 100 photons. In the test movie, we chose the  $z$  position of the emitters to range from  $-300$  nm to 300 nm, as it has been noted previously that the localization precision of the astigmatism approach depends on the  $z$  position of the emitter and that relatively high localization precisions have been achieved experimentally in the  $z$  range of  $-300$  to 300 nm<sup>1</sup>. These movies were created using a Poisson noise model, with no camera read noise, 600 nm emission wavelength and a 160 nm pixel size. The test movie was analyzed by fitting the single-emitter images to the cubic spline representation and elliptical Gaussian representation of the astigmatic PSF (Fig. 3). The fitting error in  $x$ ,  $y$  and  $z$  for each  $z$  value was determined as the average distance in  $x$ ,  $y$  and  $z$  between the localizations determined by the fitting and their nearest ground-truth emitter positions. We also include the Cramer-Rao lower bound (CRLB) calculation for an ideal estimator. The CRLB was calculated using the spline representation of the PSF and the approach in ref. 8 equation 18.

The  $x$  and  $y$  localization error was pretty similar for both approaches, with cubic spline fitting giving  $\sim 5\%$  better accuracy at the largest  $z$  offsets from the focal plane ( $\pm 300$  nm). The  $z$  localization accuracy of cubic spline fitting was noticeable better, with an average improvement of  $\sim 15\%$  within  $\pm 300$  nm of the focal plane. Spline fitting is able to come very close to the CRLB for the  $x$  and  $y$  locations of an emitter, however at the largest  $z$  values tested it is  $\sim 20\%$  worse than the CRLB.

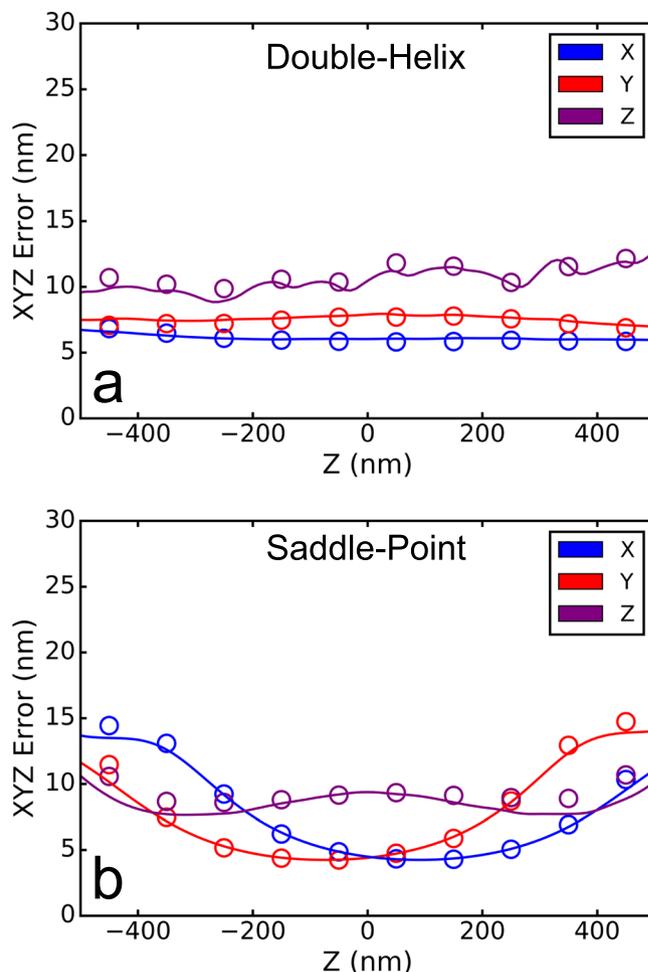
We note that in real experiments, the PSF often deviates from the ideal PSF derived from the pupil function due to non-ideal optics and optical alignment errors, which can lead to systematic errors in the determination of the  $x$ ,  $y$  and  $z$  positions of the emitters when a simple elliptical Gaussian function is used to fit the PSF<sup>21, 22</sup>. Since the cubic spline representation can capture these experimental PSFs with arbitrarily high accuracy given a



**Figure 3.** Astigmatic PSF fitting accuracy. **(a)** Average xy fitting error as a function of  $z$  for cubic spline versus elliptical Gaussian fitting. Elliptical Gaussian fitting was performed with the 3D DAOSTORM algorithm that we reported previously<sup>18</sup>. The filled points are the error in  $x$  and the hollow points are the error in  $y$ . Each data point is the average of multiple independent  $z$  positions in the 100 nm range centered on the data point. The solid gray lines are the Cramer-Rao theoretical bounds calculated following ref. 8. **(b)** Average  $z$  fitting error as a function of  $z$  and the Cramer-Rao theoretical bound. The  $z$  fitting error and the Cramer-Rao bounds are determined as described in **(a)**.



**Figure 4.** Fitting speed comparison. A comparison of the time it takes to analyze a single frame ( $256 \times 256$  pixels) of a simulated movie with indicated emitter densities using 3D-DAOSTORM or the cubic spline approach.



**Figure 5.** PSF localization accuracy. Localization errors in x, y and z are plotted as a function of z. **(a)** Double helix PSF. **(b)** Saddle-point PSF. The solid lines are the Cramer-Rao theoretical bounds calculated following ref. 8.

sufficiently small interval size, we expect cubic spline fitting to provide a greater improvement in the localization accuracy in experiments.

The average amount of time that it takes to analyze a single image with the cubic spline approach as a function of emitter density is shown in Fig. 4. Timing information for the elliptical Gaussian fitting (using the 3D-DAOSTORM algorithm) is provided for comparison. All timing measurements were performed on a laptop computer with an Intel Core i7 CPU running at 2 GHz. Simulated movies ( $41 \mu\text{m} \times 41 \mu\text{m}$  field of view) were generated with emitters randomly located in x and y in a  $\pm 500 \text{ nm}$  z range at densities of 0.03, 0.1 and 0.3 per  $\mu\text{m}^2$  (41, 136 and 408 emitters respectively). The PSF that we used in these simulations was an elliptical Gaussian whose width in x and y varied as a function of z. To reduce the variance in the timing results, we increased the number of frames of the movies until they took at least 4 seconds to analyze regardless of the emitter density. As 3D-DAOSTORM and the cubic spline algorithm share most of the other components of their analysis pipeline, the timing differences reported are primarily due to differences in how long the localization fitting step takes. At lower emitter densities the overall analysis time is more dependent on how long the various non-fitting steps, such as reading the image from disk and background estimation, take, therefore the computation time of the two algorithms is similar. At higher emitter densities the fitting step takes a larger fraction of the total time, and hence the cubic spline approach takes a proportionally longer time. These results are also consistent with our estimates for the relative computation cost of the two approaches described earlier.

It is worth noting that the fitting time will depend on the number of pixels that the PSF covers. In the above test, we used a spline that covers  $14 \times 14$  pixels in x and y. This is a reasonable size for most PSFs, but a larger size may be necessary for PSFs with a larger spatial extent or for microscope setups with a smaller pixel size. We found a size of between 12 and 16 pixels (in each dimension) was sufficiently large to work well for the PSFs that are typically used in single-molecule imaging and our pixel size of 160 nm.

An important advantage of the cubic spline approach is that it can be used to fit arbitrarily shaped PSFs, which makes it widely applicable to many different 3D localization methods using various engineered PSF shapes. Here, we demonstrate the versatility of the cubic spline approach by using it to analyze simulated data from two additional types of PSFs that have been used for super-resolution imaging, the double helix PSF<sup>2</sup> and the saddle-point PSF<sup>3</sup>. We generated two kinds of simulated movies using these PSFs, a high photon count movie for measuring

the PSF and determining the spline and a low photon count movie similar to typical single-molecule localization conditions. The saddle-point PSFs were generated using the pupil function approach<sup>10,11</sup>, and we rotated the saddle-point PSF by 45 degrees relative to the orientation reported in ref. 3 so that the PSF stretches along the x and y axes. The double helix PSF was obtained from the 2016 SMLM localization challenge web-site<sup>23</sup> and spline interpolation was used to position it at arbitrary x, y and z values. In the movies simulating single molecules, we used 4000 photons per localization, a constant background of 100 photons, and a camera pixel size of 160 nm. The x, y and z errors were again determined as the average distance in x, y and z between the localizations determined by the fitting and their nearest ground-truth emitter positions and shown in Fig. 5. From these error plots, it is evident that the cubic spline approach can fit any of these PSFs with an accuracy that approaches the optimal accuracy as determined using the CRLB.

We note that the localization errors reported in this work are all determined by fitting the relatively ideal, simulated images of single molecules with the same photon numbers for each PSF. In real experiments, the photon loss associated with generating relatively complex PSFs, for example by using spatial light modulators or deformable mirrors, also needs to be taken into account when estimating the final localization error.

## Discussion

In this work, we demonstrate that a cubic spline approach provides a 5% improvement in x and y localization accuracy and up to a 15% improvement in z localization accuracy relative to elliptical Gaussian fitting for a purely astigmatic PSF. Although the improvement in localization accuracy is moderate, the cubic spline approach has the additional advantage of being highly versatile, as it is easily adapted to fit any experimental realizable PSF. It is also a practical choice as it is only ~2–3x slower than a Gaussian fitting approach at typical emitter densities encountered in super-resolution imaging experiments. To make it easier for others to explore this approach to single-molecule localization analysis, our implementation of this algorithm, Docker images, instructions and an example of its use are available on Github<sup>24</sup>.

## References

- Huang, B., Wang, W., Bates, M. & Zhuang, X. Three-Dimensional Super-Resolution Imaging by Stochastic Optical Reconstruction Microscopy. *Science* **319**, 810–813 (2008).
- Pavani, S. R. P. *et al.* Three-dimensional, single-molecule fluorescence imaging beyond the diffraction limit by using a double-helix point spread function. *Proc Natl Acad Sci USA* **106**, 2995–2999 (2009).
- Shechtman, Y., Steffen, J. S., Backer, A. S. & Moerner, W. E. Optimal Point Spread Function Design for 3D Imaging. *Physical Review Letters* **113**, 1–5 (2014).
- Jia, S., Vaughan, J. C. & Zhuang, X. Isotropic three-dimensional super-resolution imaging with a self-bending point spread function. *Nature Photonics* **8**, 302–306 (2014).
- Juette, M. F. *et al.* Three-dimensional sub-100 nm resolution fluorescence microscopy of thick samples. *Nature Methods* **5**, 527–529 (2008).
- York, A. G., Ghitani, A., Vaziri, A., Davidson, M. W. & Shroff, H. Confined activation and subdiffraction localization enables whole-cell PALM with genetically expressed probes. *Nature Methods* **8**, 327–333 (2011).
- Mlodzianoski, M. J., Juette, M. F., Beane, G. L. & Bewersdorf, J. Experimental characterization of 3D localization techniques for particle-tracking and super-resolution microscopy. *Optics Express* **17**, 8264–8277 (2009).
- Liu, S., Kromann, E. B., Krueger, W. D., Bewersdorf, J. & Lidke, K. A. Three dimensional single molecule localization using a phase retrieved pupil function. *Optics Express* **21**, 29462–29487 (2013).
- Nelder, J. A. & Mead, R. A simplex method for function minimization. *The Computer Journal* **7**, 308–313 (1965).
- Hanser, B. M., Gustafsson, M. G. L., Agard, D. A. & Sedat, J. W. Phase retrieval for high-numerical-aperture optical systems. *Optics Letters* **28**, 801–803 (2003).
- Hanser, B. M., Gustafsson, M. G. L., Agard, D. A. & Sedat, J. W. Phase-retrieved pupil functions in wide-field fluorescence microscopy. *Journal of Microscopy* **216**, 32–48 (2004).
- Stetson, P. B. DAOPHOT: A Computer Program for Crowded-Field Stellar Photometry. *Publications of the Astronomical Society of the Pacific* **99**, 191–222 (1987).
- Holden, S. J., Uphoff, S. & Kapanidis, A. N. DAOSTORM: an algorithm for high-density super-resolution microscopy. *Nature Methods* **8**, 279–280 (2011).
- Tahmasbi, A., Ward, E. S. & Ober, R. J. Determination of localization accuracy based on experimentally acquired image sets: applications to single molecule microscopy. *Optics Express* **23**, 7630–7652 (2015).
- Kirshner, H., Vonesch, C. & Unser, M. Can localization microscopy benefit from approximation theory? *2013 IEEE 10th International Symposium on Biomedical Imaging*, 588–591 (2013).
- Anton, H. & Rorres, C. *Elementary linear algebra with applications* (Wiley, 1987).
- Jones, E., Oliphant, T. & Peterson, P. *SciPy: Open source scientific tool for Python*, [www.scipy.org](http://www.scipy.org) (2001).
- Babcock, H. P., Sigal, Y. M. & Zhuang, X. A high-density 3D localization algorithm for stochastic optical reconstruction microscopy. *Optical Nanoscopy* **1** (2012).
- Laurence, T. A. & Chromy, B. A. Efficient maximum likelihood estimator fitting of histograms. *Nature Methods* **7**, 338–339 (2010).
- Marquardt, D. W. An algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics* **11**, 431–441 (1963).
- Carlini, L., Holden, S. J., Douglass, K. M. & Manley, S. Correction of a Depth-Dependent Lateral Distortion in 3D Super-Resolution Imaging. *Plos One* **e0142949**, 1–15 (2015).
- Proppert, S. *et al.* Cubic B-spline calibration for 3D super-resolution measurements using astigmatic imaging. *Optics Express* **22**, 10304–10316 (2014).
- Single-Molecule Localization Microscopy & Software Benchmarking*. <http://bigwww.epfl.ch/smlm/> (2017).
- Storm-Analysis, Zhuang lab STORM movie analysis code*. <https://github.com/ZhuangLab/storm-analysis> (2017).

## Acknowledgements

We thank Yoav Shechtman for providing the pupil function for the saddle-point PSF. This work was supported in part by the National Institute of Mental Health Conte Center Program (P50MH094271). X.Z. is a HHMI investigator.

### Author Contributions

H.B. conceived the project, wrote and tested the spline fitting algorithms and performed the analysis with input from X.Z. H.B. and X.Z. wrote the paper.

### Additional Information

**Supplementary information** accompanies this paper at doi:[10.1038/s41598-017-00622-w](https://doi.org/10.1038/s41598-017-00622-w)

**Competing Interests:** The authors declare that they have no competing interests.

**Publisher's note:** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



This work is licensed under a Creative Commons Attribution 4.0 International License. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in the credit line; if the material is not included under the Creative Commons license, users will need to obtain permission from the license holder to reproduce the material. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>

© The Author(s) 2017