Fast, single-molecule localization that achieves theoretically minimum uncertainty

Carlas S Smith[1], Nikolai Joseph[2], Bernd Rieger[1] & Keith A Lidke[2]

We describe an iterative algorithm that converges to the maximum likelihood estimate of the position and intensity of a single fluorophore. Our technique efficiently computes and achieves the Cramér-Rao lower bound, an essential tool for parameter estimation. An implementation of the algorithm on graphics processing unit hardware achieved more than 10^5 combined fits and Cramér-Rao lower bound calculations per second, enabling real-time data analysis for super-resolution imaging and other applications.

In many single-molecule fluorescence applications, researchers often want to find the position and intensity of a single fluorophore as well as to estimate the accuracy and precision of these parameters. Accuracy is a measure of the systematic error or bias, and precision is a measure of the statistical error of an estimator[1]. In recent work using single-molecule localization to generate super-resolution images[2–6], single emitters have been located, and on the mosaic of their found positions a two-dimensional (2D) Gaussian profile has been placed to generate the final super-resolution images. The width of the placed Gaussian blob, σ, is given by the precision of the fluorophore position localization $\sigma = (\sigma^2_x + \sigma^2_y)^{1/2}$, and in these super-resolution techniques it is therefore necessary to both find the parameters and estimate their precision. Reported values are in the range of 20–70 nm. In the application of super-resolution imaging, it may be necessary to find the positions of more than 10^6 fluorophores to generate one final image of a typical field of view of 50×50 μm. In many cases, background rates of light detection may vary across the field of view, and the fluorophore emission rate of chemically identical fluorophores can vary owing to effects such as uneven illumination profile, dipole orientation or different optical path lengths.

Here we describe an iterative routine, implemented on a graphics processing unit (GPU) that calculates the maximum likelihood estimate (MLE) for the *x*-*y*-(*z*) position, the photon count of the fluorophore and the background fluorescence rate (**Supplementary Note 1**). Our approach achieved the Cramér-Rao lower bound (CRLB) over a wide range of parameters. The uncertainties of the fitted parameters are found by calculating their CRLBs[1], and in this sense the estimated σ for building up the super-resolution image is optimal. We provide a software tool (**Supplementary Software 1**) that only requires an inexpensive graphic card for single-molecule fitting speed to be sufficient for real-time data analysis.

As the speed and precision of single-particle localization is important in single-particle tracking as well as in other single-molecule biophysical techniques that rely on fluorescent reporters, others have also considered these issues. Several algorithms from the literature for finding particle positions are compared in reference 7 but without the context of a statistical framework. The theoretically best-possible estimation precision of a fluorophore position has been derived[8] by using the well-established statistical method of finding the CRLB in an unbiased parameter estimation problem. This group considered many of the effects in a real system including background fluorescence, finite camera pixel size and camera readout noise, and they recently made a software tool available for estimation (www4.utsouthwestern.edu/wardlab/estimationtool/). A comparison of nonlinear least-squares (NLLS) and MLE position estimates to the CRLB[9] revealed that, in general, MLE is more precise than NLLS. The better precision of MLE is in concurrence with our results, but this had been investigated[9] with assumed known emission and background rates. Here we developed a robust, iterative routine that finds the particle position, the intensity and the background count rate. We did not consider camera readout noise because for electron-multiplying charge-coupled device (EMCCD) cameras, which are generally used for the fast frame rates desired in super-resolution imaging, the readout noise is much less than 1 root mean square electrons (r.m.s. e^-) when using large electron-multiplying gain. Our method is not restricted to 2D imaging with a symmetric point spread function (PSF) but can be extended to handle super-resolution techniques that encompass astigmatic imaging for *z* resolution as in reference 10 (**Supplementary Note 1**, **Supplementary Data** and **Supplementary Figs. 1–4**). In the astigmatic imaging case, our iterative algorithm calculates the *z* position directly (not from intermediates σx, σy fits) and also returns the CRLB-base uncertainties. For the 2D imaging case, we compared the results of our iterative algorithm to CRLB-based theoretical values for various background rates and total collected photon counts of the PSF (**Fig. 1**). We compared the results of the iterative algorithm compared to CRLB-based theoretical values for various background rates and total collected photon counts of the PSF (**Fig. 1**). We show results for $\sigma_{PSF} = 1$ with the size defined in unitless back-projected pixels. The diffraction limit for high numeric aperture

[1]Department of Imaging Science and Technology, Delft University of Technology, Delft, The Netherlands. [2]Department of Physics and Astronomy, University of New Mexico, Albuquerque, New Mexico, USA. Correspondence should be addressed to K.A.L. (klidke@unm.edu).

RECEIVED 3 SEPTEMBER 2009; ACCEPTED 22 FEBRUARY 2010; PUBLISHED ONLINE 4 APRIL 2010; DOI:10.1038/NMETH.1449

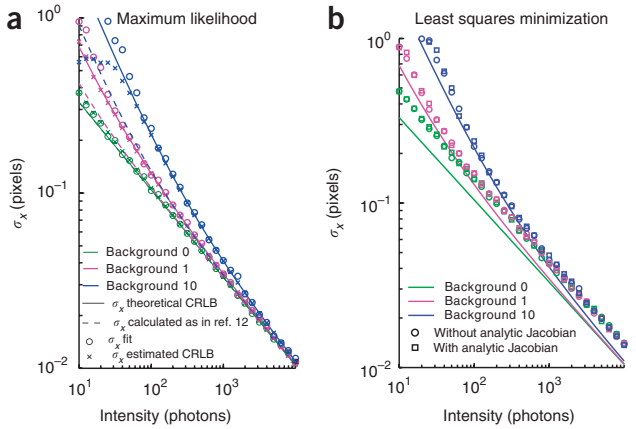

Figure 1 | Performance comparison on simulated data. Uncertainty is specified in one direction (x) only. (**a**) Localization precision of our iterative method and that given by the CRLB. Also shown are the mean uncertainties reported from CRLB calculations for every image using the found intensity and background rates (constant offset). Calculations were made using a square fitting region of size 2 × 3 σ_{PSF} + 1 and ten iterations. The theoretical uncertainty calculated from the four-parameter fit CRLB is compared to the commonly used formula of reference 12, equation 14 for estimating localization precision. (**b**) Fits were performed using nonlinear least-squares Levenberg-Marquardt minimization with and without an analytical Jacobian.

(NA) visible light imaging is >200 nm and has a σ_{PSF} >90 nm[11]. The algorithm both achieved and correctly reported the CRLB uncertainties over a wide range of background and fluorophore intensities. Calculated precision remained within a few percent of the theoretically achievable value even for less than 100 collected photons. We found that in all conditions, when the reported CRLB was less than $\sigma_{PSF}/2$ (here 0.5), the reported CRLB matched the theoretical position and the routine achieved the CRLB. In typical super-resolution applications this corresponds to <50 nm. Addition of camera readout noise has effectively the same negative influence on the parameter estimation as high background. Fortunately, this can be excluded for an EMCCD for the reasons mentioned above. We collected example images of single fluorophores with intensities and background rates near the $\sigma_{PSF}/2$ value (**Supplementary Fig. 5**). The classical approach to position fitting is via NNLS optimization (**Fig. 1b**). Here we chose a Levenberg-Marquardt (LM) optimization scheme with an analytic and computed first derivatives with respect to the optimization parameters. Note that it is common practice to use computed derivatives only. The NNLS optimization performed worse in terms of precision than our iterative MLE approach mainly because of the incorrect Gaussian noise model implicitly present in any least squares–based optimization scheme. We compared the predicted uncertainty of the fit as in reference 12 equation 14 with the theoretical CRLB (**Fig. 1a**). The fits were identical for no background fluorescence, but for any nonzero background and low light conditions the deviation was almost a factor of two. This means that in these cases the suggested uncertainty σ used to constitute the super-resolution image is estimated at nearly half the CRLB-based value (overly optimistic). The formula presented in reference 12 has the advantage that it can be readily calculated by hand from measurable quantities. However, our algorithm yields a precise estimate.

Our iterative update scheme is similar to that previously described[13]. We show, however, that a Gaussian approximation for the 2D fluorophore PSF[11] and subsequent localization leads to a compact analytical expression that allows for computationally fast localization without compromising the localization precision. Our approach achieved the CRLB after a few (~10) iterations (**Supplementary Fig. 6**). Notably, the CRLB predicts the correct precision only when the model function is correct, and the isotropic Gaussian model may not be appropriate when imaging fluorophores with a fixed dipole orientation[14], leading to anisotropic emission. A 'rule of thumb' fitting region size of $2 \times 3\sigma_{PSF} + 1$ is used (**Supplementary Data** and **Supplementary Fig. 7**). For z resolution imaging, the Gaussian PSF model is less reliable because of optical aberrations and the simplified model itself[11].

GPU-based computation has the potential to increase floating point calculation speed by a factor of 10–100 as compared to a modern central processing unit (CPU) if the problem is amenable to a parallel processing approach (http://www.nvidia.com/). A generic C-like language interface is available for simplified GPU programming (Nvidia compute unified device architecture (CUDA)), and a Matlab (Mathworks) interface has been developed. Operating with a fixed number of iterations complements the GPU's single instruction multiple data (SIMD) strategy. A GPU implementation of our iterative method can perform 10^5 combined MLE and CRLB calculations per second of the four-parameter model needed to describe the emission of a fluorophore (**Fig. 2**).

We compared the speed of CPU and GPU implementations, characterized by the number of combined position fits and CRLB calculations performed per second (**Table 1**). The slowest GPU

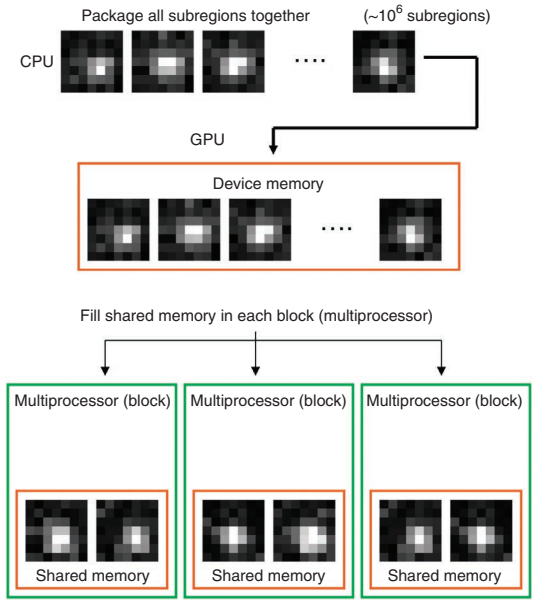

Figure 2 | Basic concept of single-molecule localization via GPU implementation. Multiple (up to millions) preselected image regions of interest are arranged into a three-dimensional dataset on the CPU and passed into GPU memory. Smaller datasets are arranged and processed in identically sized chunks that fill the multiprocessor shared memory (here shown as just two images, but this can be up to 64 images for 7 × 7 fit regions). Each image is analyzed with the same iterative algorithm. One thread (subprocessor) processes one image. The hundreds of subprocessors available on a GPU result in a speed increase because of massive parallel processing.

Table 1 | Computational performance

	Iterative MLE method				LM (numeric Jacobian)
Box size (pixels)	AMD phenomII (10^2 fits s^{-1})	Nvidia 8600GTS (10^2 fits s^{-1})	Nvidia 8800GTS (10^2 fits s^{-1})	Nvidia GTX285 (10^2 fits s^{-1})	AMD phenomII (10^2 fits s^{-1})
7 × 7	31	430	880	2,600	15
13 × 13	9.4	45	100	950	5.2
15 × 15	4.3	10	22	330	2.3

The speed of CPU and GPU implementations of the iterative MLE and a Levenberg-Marquardt nonlinear least-squares fit.

tested outperformed the CPU by more than one order of magnitude, with the fastest achieving 2.6×10^5 fits per second on a 7×7 fitting box size. We attribute this performance gain over the CPU to the fact that this estimation problem is almost ideal for the GPU SIMD architecture. Many iterations are performed on the same data, which are stored in local shared memory, and each thread is independent, eliminating synchronization delays. All CPU computation only ran on a single thread. We determined the performance of a nonlinear, least-squares fit (in C code) on a CPU for reference (**Table 1**), and it was twice as slow as the performance of our iterative algorithm on a CPU. Commonly used Matlab LM optimization only computed about five fits per second. Given the readout rate of current high-end (EM)CCD cameras (~10 MHz), our GPU implementation performed combined fits and uncertainty estimates at a speed sufficient for real-time analysis (Online Methods).

Here we derived an iterative approach for making an MLE of the position and intensity of a single fluorophore as well as the background count rate using a two-dimensional Gaussian PSF model and a Poisson noise model. The iterative method achieves the minimal possible estimation uncertainty, as given by the CRLB, over a wide range of emission and background rates that could be found in single-molecule experiments. A demonstration of the fitting results on immobilized single molecules is shown in **Supplementary Figure 8**. Implementation of the iterative method on a modern GPU yields more than 10^5 combined fits and CRLB calculations per second, greatly facilitating the analysis of large datasets found in single molecule–based super-resolution techniques and is suitable for real-time data analysis.

METHODS

Methods and any associated references are available in the online version of the paper at http://www.nature.com/naturemethods/.

Note: Supplementary information is available on the Nature Methods website.

ACKNOWLEDGMENTS

This work was supported by the American Cancer Society (IRG-92-024) and the US National Institutes of Health (1P50GM085273-01A1). We thank M. Malik and I.T. Young for reading the manuscript.

AUTHOR CONTRIBUTIONS

C.S.S. and K.A.L. worked out the algorithm and implementation; N.J. and K.A.L. performed the experiments; and B.R. and K.A.L. designed the research and wrote the paper.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

Published online at http://www.nature.com/naturemethods/.
Reprints and permissions information is available online at http://npg.nature.com/reprintsandpermissions/.

1. van den Bos, A. *Parameter Estimation for Scientists and Engineers* (Wiley & Sons, 2007).
2. Betzig, E. *et al. Science* **313**, 1642–1645 (2006).
3. Bates, M., Huang, B., Dempsey, G.T. & Zhuang, X. *Science* **317**, 1749–1753 (2007).
4. Egner, A. *et al. Biophys. J.* **93**, 3285–3290 (2007).
5. Hess, S.T., Girirajan, T.P.K. & Mason, M.D. *Biophys. J.* **91**, 4258–4272 (2006).
6. Lidke, K.A., Rieger, B., Jovin, T.M. & Heintzmann, R. *Opt. Express* **13**, 7052–7062 (2005).
7. Cheezum, M.K., Walker, W.F. & Guilford, W.H. *Biophys. J.* **81**, 2378–2388 (2001).
8. Ober, R.J., Ram, S. & Ward, E.S. *Biophys. J.* **86**, 1185–1200 (2004).
9. Abraham, A.V. *et al. Opt. Express* **17**, 23352–23373 (2009).
10. Huang, B., Wang, W., Bates, M. & Zhuang, X. *Science* **319**, 810–813 (2008).
11. Zhang, B., Zerubia, J. & Olivo-Marin, J.C. *Appl. Opt.* **46**, 1819–1829 (2007).
12. Thompson, R.E., Larson, D.R. & Webb, W.W. *Biophys. J.* **82**, 2775–2783 (2002).
13. Aguet, F., Van De Ville, D. & Unser, M. *Opt. Express* **13**, 10503–10522 (2005).
14. Enderlein, J., Toprak, E. & Selvin, P. *Opt. Express* **14**, 8112–8120 (2006).

ONLINE METHODS

GPU implementation. The iterative method to solve the MLE problem described in the main text was implemented on a GPU using Nvidia CUDA, a C-based programming language that makes it possible to readily program parallelized algorithms that are executed on a GPU. High-end gaming and computing GPUs have a (card-dependent) large region of global device memory, usually several hundred megabytes. Execution is performed on many (card-dependent) multiprocessors. Multiprocessors contain 8 subprocessors and have 16 kilobytes local memory that is shared between subprocessors. Access to local shared memory is fast, whereas access to the device global memory incurs a large performance penalty and should be avoided when possible. The programming model follows the GPU architecture in that parallelized execution is performed by breaking down computations into 'blocks' and 'threads'. Each block executes a set of threads using one multiprocessor. Typically, performance is optimal when multiples of 32 threads (called a 'warp') are scheduled and executed using the 8 subprocessors. However, we found for all fit box sizes, the maximum fits per second occurred when the maximum fits per block were used, which is limited by the available 16 kilobytes of shared memory per block. This is likely due to compiler optimization and the fact that each fit is independent of all others (no thread synchronization is required).

We mapped our iterative algorithm on this programming model in the following way. A data stack consisting of identically sized subregions, which are centered single-emitter images, are input to the function. These data are copied from host (CPU) memory to device (GPU) main memory. This dataset is divided into blocks, which consist of the largest number of images that can fit into shared memory. The execution of a block begins by copying the data subset into local shared memory. Each thread then calculates a complete fit and CRLB calculation for one image. The Fisher information matrix is calculated using equations 9 and 11 (**Supplementary Note 1**), and the CRLB is calculated using the analytical expression for the inverse of a 4×4 matrix. The CPU performance was measured by replacing the block/thread architecture with nested loops, which call the same subfunction, and was compiled using Microsoft Visual Studio Express 2008. Images were loaded in Matlab (Mathworks) as arrays and the C code and the CUDA GPU code were called via mex files. The CPU was an AMD Phenom II X3 720 at 2.8 GHz and only one core was used for the C code. We provide a software tool and an example that implements and demonstrates the GPU analysis (**Supplementary Software 1**, http://www.diplib.org/add-ons/).

Synthetic data generation and analysis. The CUDA routine described above operates on a dataset consisting of identically sized images that contain images of (potential) single molecules. For the analysis of synthetic data, a stack is generated using the same finite pixel approximation, including background as described in the **Supplementary Note 1.** The center coordinate of each simulated emitter is randomly shifted within the central pixel to prevent a biased result. After the generation of the data stack, the images are corrupted with Poisson noise. This data stack is analyzed by the routine, which returns the estimated position, intensity, and background and the CRLB calculation for each 2D image in the stack. No camera read noise was added. Camera read noise for EMCCD cameras, which are generally used for the fast frame rates desired in super-resolution imaging, is much less than 1 r.m.s. e^- with large electron multiplying gain.

Levenberg-Marquardt nonlinear least-squares fitting. The standard way of fitting a Gaussian to data is via least-squares fitting. We used (i) an existing implementation of Matlab via the optimization toolbox (lsqcurefit), in which we enabled the Levenberg-Marquardt option as minimization scheme and (ii) an implementation from Numerical Recipes in C[15]. We ran the test on fits with computed Jacobian only and on fits for which we supplied analytic first derivatives. We used the following limits in the stopping criterion: tolerance on the parameters 10^{-4}, tolerance on the function 10^{-15} and maximal 10^5 function evaluations. The stopping criterion was in all cases determined by the accuracy put on the parameters. The Matlab routine was a lot slower (about two orders of magnitude) than the C implementation, although it makes automatic use of all available cores on the CPU if multithreading is enabled.

Single-molecule imaging. Single-molecule imaging experiments were performed in an epifluorescence microscope setup consisting of an inverted microscope (IX71; Olympus), 1.45 NA objective (U-APO 150×, NA 1.45; Olympus), 635 nm diode laser (Radius 635; Coherent Inc.) and an EMCCD camera (Luca DL6581-TIL; Andor Technologies PLC.). The pixel size was 10 μm. The epifluorescence filter setup consisted of a dichroic mirror (650 nm; Semrock) and an emission filter (692/40; Semrock). Individual Cy5 molecules were immobilized on an amino-silane ((3-amino propyl)triethoxysilane; Sigma-Aldrich) treated 8-well chambered cover slips (Lab-Tek II; Nunc) via an NHS-ester linkage attached to the Cy5 (Cy5 monoreactive dye pack; GE Healthcare). An oxygen scavenging system[16] was used to extend fluorophore lifetimes and quench fluorophore triplet states. This was necessary to perform repeated measurements of the same single emitter for several frames while acquiring sufficient photons to address localization accuracy. This is not necessary in a dedicated experiment.

Data were recorded by a CCD camera at either 10 or 20 frames per second. All data were postprocessed by (i) subtracting a pixel-dependent camera offset, which was created by averaging 300 dark frames, and (ii) multiplying the resulting image by a gain factor to restore correct Poisson statistics, as done in previous work[17]. Single-molecule candidates were identified in each time frame as regions in which the 2D Gaussian filtered image ($\sigma = \sigma_{PSF}$) was greater than 1 s.d. of this image. Note that owing to the speed of the GPU implementation, a simple but fast method for identifying candidates is preferred as well as one that errs on the side of including regions that do not contain single molecules. Square regions of a specified number of pixels that included all identified regions in the time series were collected into one stack and input to the GPU routine. The resulting found coordinates were used to build trajectories only if they passed the following criteria: (i) reported localization accuracy was less than one-fifth σ_{PSF} in each dimension, and (ii) $1/N \, \Sigma_k \ln(L(x_k|\theta)) > -1$, in which N is the number of pixels and $L(x_k|\theta)$ is the likelihood of the data x_k, given the model parameterized by θ (**Supplementary Note**, equation 7). Criterion (ii) essentially performs a shape test and can rule out obvious cases of two proximate emitters. The remaining coordinates were connected into 'trajectories' using an existing single-particle tracking routine[18]. Only 'trajectories' that showed little

doi:10.1038/nmeth.1449

triplet state blinking were used in the final analysis, with a cut-off criterion that $\mathrm{var}(I(t)) < 2 \times \mathrm{mean}(I(t))$, in which $I(t)$ is the sum over all pixels in the analyzed region in frame t. 'Trajectories' were adjusted to compensate for microscope stage drift by subtracting a linear regression line from each single-particle trajectory.

The width of the PSF used in the fitting routine was found by minimizing the mean square error between the finite pixel model and the summed projection over a 100-frame time series. The σ_{PSF} used in subsequent analysis was the average found from analyzing the summed projection of 5 different single emitters.

Astigmatic imaging. The three-dimensional astigmatic imaging was calibrated by imaging 100 nm red (690 nm emission) beads (FluoSphere; Invitrogen) bound to the bottom of an 8-well cover slip chamber (Lab-Tek II; Nunc). The filter setup used was the same as that used for single-molecule Cy5 imaging. We imaged using a 60× 1.2 NA water objective. A 500 mm focal length cylindrical lens was inserted in the emission beam path just after the first lens of a two-color beam splitter (OptoSplit II; Cairn Research). A piezoelectric z-stage (Nano-LPS; Mad City Labs) translated the focal plane in steps of 50 nm from −0.5 µm to 0.5 µm. At each focal plane, 20 images of a bead were captured. The fit box size used was again calculated by $2 \times 3 \times \sigma_{PSF} + 1$, but here σ_{PSF} is taken as the maximum value of either σ_{PSFx} or σ_{PSFy}, in this case giving a fit box size of 13×13 pixels.

After gain and background correction, the sum of all images from each focal plane were used to find $\sigma_x(z)$ and $\sigma_y(z)$, which were then fit to the model of equation 15 (**Supplementary Note 1**). The fit is shown in **Supplementary Figure 1**. From the calibration, the following values for the parameters of equation 15 (**Supplementary Note 1**) were found: $\sigma_{0x} = 1.08$, $\sigma_{0y} = 1.01$, $Ax = -0.0708$, $Ay = 0.164$, $Bx = -0.073$, $By = 0.0417$, $d = 0.531$ and $\gamma = 0.389$. The depth of field for a high-NA imaging system is given by $\mathrm{DOF} = \lambda/(4n(1 - (1 - \mathrm{NA}^2/n^2)^{1/2})) \approx 230$ nm[19] but here was included as a fit parameter.

Real-time data analysis. Initially, the bottleneck for the build-up of a super-resolution image was the switching cycle speed for activating only a very small number of particles per image, which resulted in imaging times of many hours[20]. Subsequent acceleration was achieved by optimizing the fluorophores for activation-based super-resolution, protocol improvements or reducing the number of time frames[21–23].

The fundamental relationships between error and acquisition rate (number of activation cycles) have been investigated in a theoretical study[24]. The findings are relevant to specific chosen or given activation probability. However, to assess the required speed for real-time data analysis, we addressed the problem somewhat differently. We used the field of view V, the size of the footprint of the PSF P, the frame rate F and the fill factor f of the single-emitter distribution on the field of view. The required fits per second for real-time data analysis are then fits per second = VFf / P.

We considered two common cases of EMCCD cameras for the maximal fill factor of $f = 1$ and a PSF of $P = 7 \times 7$ pixels: (i) $V = 128 \times 128$ pixels, $F = 500$ frames s^{-1} and (ii) $V = 512 \times 512$ pixels, $F = 30$ frames s^{-1}. For the first case $\sim 1.7 \times 10^5$ fits s^{-1} are required, and for the second $\sim 1.6 \times 10^5$ fits s^{-1}; these values are about equal as the total readout rate (~10 Mhz) is the limiting factor and is about equal. The PSF footprint can vary for different physical CCD camera pixel sizes and magnifications. In any case, the fastest GPU (2.6×10^5 fits s^{-1}) tested already fulfills this requirement for current fluorophores and CCD cameras. Of course, a fill factor of $f = 1$ is optimistic; more realistic values are 0.1–0.5 but are dependent on the experimental conditions and can be chosen as previously described[24]. This means that also the slower (and cheaper) cards already are sufficient in current practice for real-time fitting of positions. The importance of the GPU fitting in the context of the entire process of segmentation (identifying regions of interest for single-molecule fits), organizing regions of interest, single-molecule fitting and reconstruction, is shown in **Supplementary Table 1**. Segmentation and reconstruction were performed as described previously[25] with the segmentation performed on the GPU. The results show that with even with 10^6 total fits, corresponding to 100 fits per frame, the overall processing could exceed the maximum possible frame rate of 500 Hz of available EMCCDs.

15. Press, W., Teukolsky, S., Vetterling, W. & Flannery, B. *Numerical Recipes in C: The Art of Scientific Computing* 2nd edn. (Cambridge University Press, Cambridge, UK, 1992).
16. Rasnik, I., McKinney, S.A. & Ha, T. *Nat. Methods* **3**, 891–893 (2006).
17. Lidke, K.A., Rieger, B., Lidke, D.S. & Jovin, T.M. *IEEE Trans. Image Process.* **14**, 1237–1245 (2005).
18. Andrews, N.L. *et al. Nat. Cell Biol.* **10**, 955–963 (2008).
19. Young, I. *et al.* in *Proceedings of the 8th Scandinavian Conference on Image Analysis* (eds., Høgda, K.A., Braathen, B. & Heia, K.) 493–498 (Norwegian Society for Image Processing and Pattern Recognition, Tromsø, Norway, 1993).
20. Betzig, E. *et al. Science* **313**, 1642–1645 (2006).
21. Egner, A. *et al. Biophys. J.* **93**, 3285–3290 (2007).
22. Rust, M.J., Bates, M. & Zhuang, X.W. *Nat. Methods* **3**, 793–795 (2006).
23. Bates, M., Huang, B., Dempsey, G.T. & Zhuang, X. *Science* **317**, 1749–1753 (2007).
24. Small, A. *Biophys. J.* **96**, L16–L18 (2009).
25. Wolter, S. *et al. J. Microsc.* **237**, 12–22 (2010).