Supplemental Materials for DLBI: Deep learning guided Bayesian inference for structure reconstruction of super-resolution fluorescence microscopy

Yu Li[1,†], Fan Xu[2,†], Fa Zhang[2], Pingyong Xu[3], Mingshu Zhang[3], Ming Fan[4], Lihua Li[4], Xin Gao[1,*], Renmin Han[1,*]

[1]King Abdullah University of Science and Technology (KAUST), Computational Bioscience Research Center (CBRC), Computer, Electrical and Mathematical Sciences and Engineering (CEMSE) Division, Thuwal, 23955-6900, Saudi Arabia.
[2]High Performance Computer Research Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China.
[3]Key Laboratory of RNA Biology, Institute of Biophysics, Chinese Academy of Sciences, Beijing 100101, China.
[4]Institute of Biomedical Engineering and Instrumentation, Hangzhou Dianzi University, Hangzhou, 310018, China.

[†]These authors contributed equally to this work.
[*]All correspondence should be addressed to Xin Gao (xin.gao@kaust.edu.sa) and Renmin Han (renmin.han@kaust.edu.sa).

S1 Terminologies in Fluorescence Microscopy

Fluorescent protein: A fluorescent protein refers to a protein that exhibits bright fluorescence when exposed to light under a particular ultraviolet range.

Fluorescence microscope: A fluorescence microscope is an optical microscope that uses fluorescence and phosphorescence instead of, or in addition to, reflection and absorption to study properties of organic or inorganic substances.

Fluorophore: A fluorophore (or fluorescent probe) is a fluorescent protein that can re-emit light upon light excitation. Therefore, the terminology of "fluorophore" is equivalent to "fluorescent protein".

PSF: The point spread function (PSF) describes the response of an imaging system to a point source or point object.

FWHM: Full width at half maximum (FWHM) is an expression of the extent of a function given by the difference between the two extreme values of the independent variable at which the dependent variable is equal to half of its maximum value. Fig. S1 shows the relationship between a Gaussian presented PSF and FWHM.

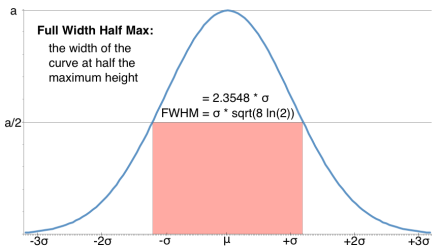


Figure S1: The relationship between a Gaussian presented PSF and FWHM.

HMM: A hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (i.e. hidden) states.

S2 Implementation Details of Deep Neural Networks

S2.1 Detailed generator network

Fig. S2 shows the detailed generator network. As discussed in the main text, the generator network is composed of two major components, the residual network and the multiscale upsampling component. The core of the residual network, the residual network building block, is shown in the snapshot of Fig. S2. Instead of using the convolutional layer to directly fit the transformation between the input feature map and the output feature map, the residual block tries to fit the residue of the output deduced by the input. This architecture is proved to be more effective than the traditional convolutional layer, eliminating the model degradation problem and gradient explosion or vanish problem (He *et al.*, 2016; Lim *et al.*, 2017). The batch normalization layer (Ioffe and Szegedy, 2015), which is proved to be able to deal with the "internal covariate shift" problem and has become a standard configuration of a deep learning model, is also adopted in our model. The multiscale upsampling component, which is the core idea of model to eliminate the fake details, is composed of several pixel shuffle layers and convolutional layers. Using those layers, our model is able to output 2X, 4X, and 8X super-resolution images, which means we have multiple interfaces for calculating the training error and performing error backprobagation. Tuning the model carefully using the above techniques, we can obtain a well-trained model, which can capture the hidden structures while not introducing too much fake detail.

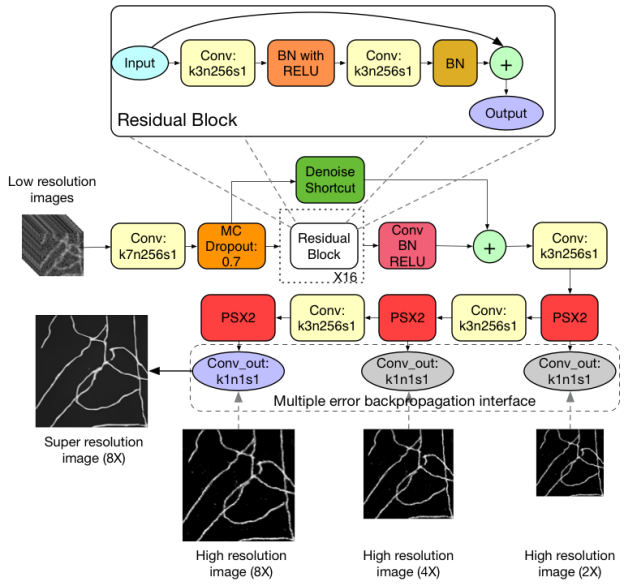


Figure S2: The generator network. According to the inputs' physical meaning, the filter size of the first convolutional layer is set to be 7 by 7, which is larger than the common filters and whose output is fed to the Monte Carlo dropout layer. The output of the Monte Carlo dropout layer would go two directions. The first one is the residual network. The second one is the denoise shortcut. The outputs of these two components are added together elementwisely. (a) The residual network consists of 16 residual blocks in total. The architecture of the residual block is shown in the snapshot. Here, k3n256s1 means that the kernel size is 3 by 3 and the output channel is 256, with the stride step as 1; BN with RELU means that the input would go through a batch normalization layer, followed by the ReLU activation. (b) PSX2 in the above figure means the pixel shuffle layer whose scaling factor is 2, which is used to perform the upscaling of the figure dimensionality. We used a convolutional layer, whose kernel size is 1 by 1 and the output channel number is 1 with the stride step as 1, to convert the feature maps into the final output images. Those output layers provide the training interface for doing error back propagation. So during training, we can tune the model gradually and prevent the 8X image from incorporating too much fake detailed information, which does not exist in the original image.

S2.2 Detailed discriminator network

The detailed discriminator network is shown in Fig. S3. We adopted the commonly used convolution neural network architecture. However, since our goal is to train a generator which can produce super-resolution images that are very similar to the ground-truth high-resolution images, if the discriminator does not have enough detection ability or, in the other extreme case, has too strong detection ability, the generator would ether become too lazy or never converge. To empower the discriminator a reasonable level of detection ability, we added one residual block after those convolutional layers.

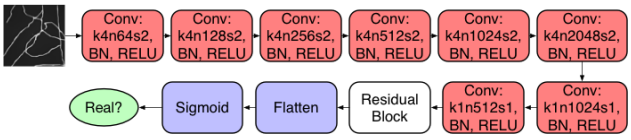


Figure S3: The discriminator network. Each convolutional block contains a convolutional layer, whose hyperparameters are defined by k, n and s, a batch normalization layer and a ReLU activation layer. The meanings of those annotations are the same as those in Fig. S2. The residual block can also be referred to Fig. S2. This network is trained to distinguish the super-resolution images from the real high-resolution images, which can force the generator network, whose task is to perplex the discriminator network, to learn to reconstruct super-resolution images with details.

S3 Implementation Details of Bayesian Inference

The Bayesian inference module takes both the time-series low-resolution images and the primitive super-resolution image produced by the deep learning module as inputs, and generates a set of optimized fluorophore

locations, which are further interpreted as a high-confident super-resolution image. Here, the original time-series low-resolution images are considered as the observations and the primitive super-resolution image produced by the deep learning is given as an initial estimation of the fluorophore positions.

S3.1 Basic concepts

In our Bayesian inference, the emitting fluorophores are modeled as spots with a Gaussian shape. For each activated fluorophore, its property is defined by the following three variables: the positions (x, y), the Gaussian radius and the brightness (radius r and intensity I). As explained in Section 2.1, a log-normal distribution (Cox *et al.*, 2012; Zhu *et al.*, 2012) is used to approximate the experimentally measured single fluorophore photon number distribution.

The switching procedure of fluorophores is modeled by Bayesian inference, i.e., given an observed region R, deciding whether there is a fluorophore (F) or not (N) by

$$\frac{P(F|R)}{P(N|R)} = \frac{P(R|F)P(F)}{P(R|N)P(N)}. \tag{1}$$

Here, $P(F)$ and $P(N)$ are constants which are based on experimental prior, $P(R|F)$ is the probability of the observed data region R given the certain position of the fluorophore, $P(R|N)$ is the probability of the observed data region R if there is no fluorophore.

If there is one fluorophore, the positive probability can be calculated employing the integration of all the probability of observing pixels:

$$P(R|F) = \int_{a\in\mathbb{R}^4} \int_{b\in\mathbb{Z}_3^N} P(R, a, b|F)\,\mathrm{d}b\,\mathrm{d}a, \tag{2}$$

where a represents the variable containing the positions (x, y), the radius r and the intensity I, b represents the hidden Markov state, and N is the number of image frames. As the switching procedure is modeled by a hidden Markov model (HMM), the integration over b can be solved by the forward algorithm (Rabiner, 1989), while the optimization of a can be performed by MAP (maximum a posteriori) estimation.

If there are multiple fluorophores, for example, M fluorophores, the procedure will be evolved to factorial hidden Markov model, and the calculation of positive probability is extended as:

$$P(R|F_M) = \int_{a\in\mathbb{R}^{4M}} \int_{b\in\mathbb{Z}_3^{MN}} P(R, a, b|F_M)\,\mathrm{d}b\,\mathrm{d}a, \tag{3}$$

where $a = \{a_1, a_2, ..., a_M\}$ and $b = \{b_1, b_2, ..., b_M\}$, representing the assemble of each ith fluorophore's variables. Exact integration over state sequences rapidly becomes intractable with increasing M so we perform integration using MCMC (Markov Chain Monte Carlo) sampling. The discrete states are sampled using Gibbs sampling (Geman and Geman, 1984) since we can sample from Markov chains efficiently using forward filtering backwards sampling (Godsill *et al.*, 2004).

Compared with the 3B analysis, we proposed a limited forward algorithm instead of the classic forward algorithm to accelerate the computation without loss of accuracy, and used an integration expansion sampling strategy, which is based on already known localization spots to get neighboring spots to assist the solving of a.

S3.2 Detailed modeling

With a large number of fluorophores transferring among different states, the independent behavior of each fluorophore will generate the high-density fluorescent image sequences, which can be modeled as FHMM. The parameters of FHMM are estimated by the expectation-maximization (EM) algorithm:

$$Q\left(\phi^{new}|\phi\right) = E\left\{\log P\left(\{F_t, D_t\}\,|\phi^{new}\right)|\phi, \{D_t\}\right\}, \tag{4}$$

where the observation is a T-frame data $\{D_t\}$, with $t = 1, ..., T$. The hidden data is $\{F_t\}$, where each fluorophore has three possible states in the model. Q is a function of the fluorophore parameters ϕ^{new}, given

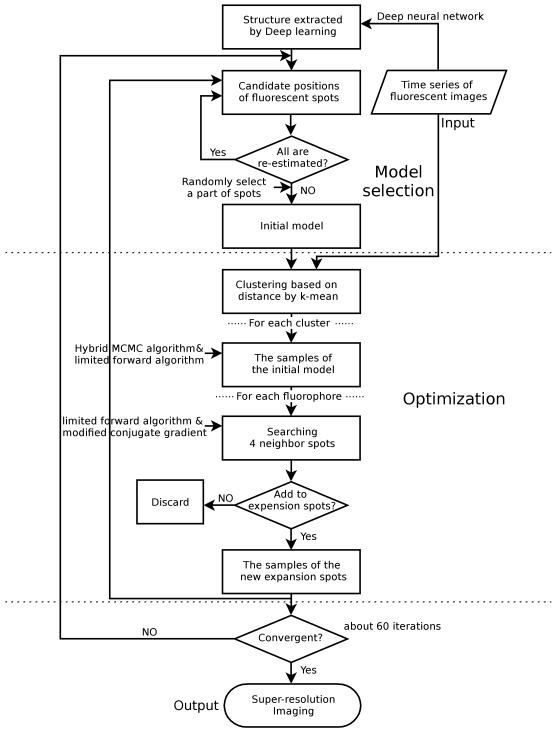


Figure S4: Schematic diagram of the Bayesian inference in DLBI. The Bayesian inference in DLBI mainly includes three steps: 1) structure extraction by deep learning, 2) model selection, and 3) model optimization. In step 1, structure information is extracted by the deep learning module described in Section 2.2, which composes a good initial estimation of the fluorophore positions. Then, by re-sampling the fluorophore distribution from the deep learning output with a random punishment, a large number of candidate fluorophore positions are generated, which will be used as the initial model in the next step. In step 2, the initial model is determined by a set of randomly selected candidate fluorescent spots. When candidate spots are selected, they will be removed afterwards to guarantee no duplicate counting. In step 3, we optimize the initial model using hybrid MCMC, limited forward algorithm and modified conjugate gradient method. Based on the initial model, it will search 4 neighboring spots and further extend the search-space around the structure. Repeating this process, a super-resolution image will be generated.

the current parameter estimation and the observation sequence $\{D_t\}$. The procedure iterates between a step that fixes the current parameters and computes posterior probabilities over the hidden states (the E-step) and a step that uses these probabilities to maximize the expected log likelihood of the observations as a function of the parameters (the M-step).

The structure extracted from the deep learning module contains abundant information, which composes a good initial estimation of the fluorophore positions. However, our Bayesian inference will not instantaneously consider all positional data. Instead, candidate positions are randomly selected to build the initial model. In the subsequent cycle, new candidate spots are selected from the remaining fluorophores. This method is equivalent to a "model pool" wherein newly obtained fluorophores are added to the pool to expand the size of the pool, further extending the neighbor search-space around the structure. By this way, the abundant detailed structure from deep learning is recalibrated and the effect of artifacts is suppressed.

Several key techniques are adapted to accelerate the speed and improve the optimization:

Limited forward algorithm: In the 3B analysis, the fluorophores are assumed to be occurred anywhere in the region with equal probability. A hybrid Markov chain Monte Carlo (MCMC) algorithm and the forward algorithm are used to optimize the parameters of a candidate fluorophore (Cox *et al.*, 2012). For each fluorophore with considerable observing probability, the entire image region will be calculated in the 3B analysis, and the workload of the 3B analysis scales with the product of the number of fluorophores and the number of pixels, which results in a high computational-cost. In our implementation, instead of uniform sampling, the structure extracted by deep learning is served as prior knowledge. For a fluorophore candidate in our workflow, only the information of its nearby area is considered, which is defined by several factors of the radius of PSF. Considering the fast decay of PSF (3σ rule of Gaussian shape) and the principle of

linear optical system, our forward algorithm is a close approximation to the original one with dramatically decrease of computational workload.

Expansion sampling: In the E-step (sampling), the hybrid MCMC and the limited forward algorithm are used to sample the initial model from the deep learning output. When a new fluorophore is determined, the forward filtering and backward sampling algorithm (Godsill *et al.*, 2004) are used to take samples of this fluorophore. These new samples will be added to the sampling of the initial model. By this way, the initial model for each turn's optimization is expanded and the chance to catch "missed" and "real" structures is increased.

Modified conjugate gradient: In the M-step (maximizing the expected log likelihood of the system), it is very possible that two very close initial fluorophores jump into the same local optimal positions, which may result in artificially "bright" areas. A further effect is the enforcement of the gradient near the local optimal minimum which leads to a local optimal trap. Here, similar to the stochastic gradient descent, we introduce a random re-weighting of conjugate gradient's direction to overcome local optimal without loss of efficiency.

Fig. S4 shows the overall workflow of our Bayesian inference.

S3.3 Parameter initialization

Each fluorophore is modeled as a Markov model, which transfers among three states, i.e., emitting, not emitting and bleached (Fig. 3 in the main text). To solve the inverse HMM problem, two probabilities are needed: the initial probability of each fluorophore's state and the transition probability between different states. Here, we set the initial probability of the three states for each single fluorophore b_i as $p_i = [0.5, 0.5, 0]$. The transition probability is given by the photo-convertible fluorescent protein (PCFP) mEos3.2 (Zhang *et al.*, 2012) in real experimental conditions, whose transition matrix is defined as

$$A = \begin{pmatrix} 0.21 & 0.79 & 0 \\ 0.305 & 0.685 & 0.01 \\ 0 & 0 & 1 \end{pmatrix} \tag{5}$$

where the labeling of each row and column is {1 = emitting, 2 = not emitting, 3 = bleached}. Because the imaging environment and fluorescent protein characteristics are quite stable, the transition matrix A is considered as constant during optimization.

For a variable vector a_i, the statistical intensity I and the shape of a fluorophore are considered following log-normal distribution. For intensity I, where $\ln(I) \sim \mathcal{N}(\mu, \sigma^2)$, we generally initialize $I.\mu$ to 2 and $I.\sigma$ to 1, similar to the 3B analysis.

The shape parameter is depended on FWHM and the pixel size in imaging. Assuming we have the FWHM in nm and the pixel size in nm, the radius r can be calculated by the following equation:

$$r.\mu = \ln\left(\frac{FWHM/(pixel\ per\ nm)}{2\sqrt{2 * \ln 2}}\right) + (r.\sigma)^2, \tag{6}$$

where we always use $r.\sigma = 0.1$ as initialization.

S4 Real-world Data Preparation

S4.1 Cell culture, transfection and fixation

The two actin datasets are from two different U2OS cells in McCoy's 5A Medium Modified (MCMM) (Life Technologies) and the Endoplasmic reticulum dataset is from the COS7 cells in Dulbecco's Modified Eagle Medium (DMEM) supplemented with glucose (Life Technologies) and 10% fetal bovine serum (Life Technologies) and penicillin/streptomycin (Hyclone) were grown at 37 ℃with 5% CO2. For transient expression, cells cultured in 12-well plates (Nunc) at 80% confluence were transfected with 1 μg lifeact-mEos3.2 (Zhang *et al.*, 2012) plasmid using Lipofectamine 2000 (Life Technologies) following the manufacturer's protocol. Five hours post transfection, cells were trypsinized and plated on clean coverslips (Fisher Scientific) coated with 10 mg/ml fibronection (Millipore, FC010) to induce spreading for an additional 24 h. Fixations were

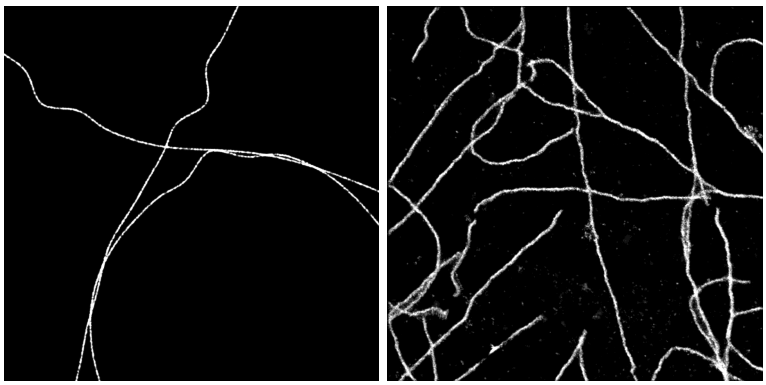
performed with PBS buffer (pH 7.4) containing 4% paraformaldehyde and 0.2% glutaraldehyde for 15 min at 37℃just before imaging. In sample preparation of ER structure, approximately 48 h after plating, COS7 cells were rinsed with DMEM with glucose and no phenol red (Life Technologies). Then, the cells were incubated in staining solution containing 10 μM ER-Tracker Red (Invitrogen) for 4 min. Imaging buffer was prepared with DMEM supplemented with 2% glucose, 6.7% of 1 M HEPES (pH 7.4), and an oxygen scavenging system (0.5 mgMmL glucose oxidase and 40 μg/mL catalase).

S4.2 Imaging system and acquisition

A homemade TIRF microscopy system with an Olympus IX71 body (Olympus) and high-NA oil objectives was used for imaging acquisition. For actin imaging, a 100×, 1.49 NA oil objective (Olympus PLAN APO) was used with image pixel size of 160nm. For ER structures, a 100×, 1.7 NA objective and 1.6X intermediate magnifications were used with image pixel size of 100nm. The fluorescence signals were acquired using an electron-multiplying charge coupled device (EMCCD) camera (Andor iXon DV-897 BV).

S5 Experimental Results

S5.1 Simulated datasets



(A) MT0.N1.HD (B) Tubulin ConjAL647

Figure S5: The whole-scale images of the simulated datasets.

Limited by space, we did not show the entire simulated datasets in our manuscript. Fig. S5 shows the ground-truth high-resolution images of MT0.N1.HD and Tubulin ConjAL647[1]. For the convenience of computation, both MT0.N1.HD and Tubulin ConjAL647 were cropped into four subareas and each subarea generated a time-series of simulated low-resolution fluorescent images. For MT0.N1.HD, because the right bottom area is almost blank, the subareas used in the simulation were cropped from ROI: (40,0,480,480), (480,0,480,480), (0,480,480,480), and (260,120,480,480), respectively. For Tubulin ConjAL647, the subareas used in the simulation were cropped from ROI: (40,0,480,480), (480,0,480,480), (0,480,480,480), and (480,480,480,480), respectively.

Fig. S6 shows the results on the fourth subarea that are not shown in the main text.

S5.2 Real-world datasets

The three raw real-world datasets were included as movies and available at `https://drive.google.com/file/d/1DOMv2Fo9WbIyz7-biWEyVybqknJ65lid/view?usp=sharing`. Due to the large size of reconstructed super-resolution images, we only showed the compressed whole-scale super-resolution images in the manuscript, and included the original versions in the package. Here is the list of main data files included in the package:

[1]Downloaded from http://bigwww.epfl.ch/smlm/datasets/; and the fluorophore positions are binned into 960 × 960 images to serve as the ground-truth.

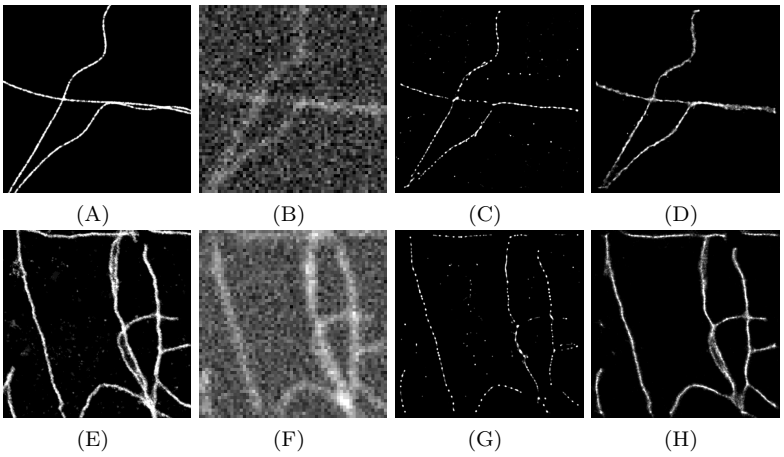


Figure S6: Reconstruction performance on the fourth subarea of the two simulated datasets. The images in the first row are the ground-truth, the first frame of simulated low-resolution fluorescence images, the 3B analysis reconstruction, and the DLBI reconstruction on the fourth subarea of the MT0.N1.HD dataset. The images in the second row are the ground-truth, the first frame of simulated fluorescence images, the 3B analysis reconstruction, and the DLBI reconstruction on the fourth subarea of the Tubulin ConjAL647 dataset.

- real1.avi: Movie of the first real-world dataset (Actin1).
- real2.avi: Movie of the second real-world dataset (Actin2).
- real3.avi: Movie of the third real-world dataset (ER).
- real1.png: Large-field reconstruction of the yellow area from the first real-world dataset (Actin1).
- real2.png: Large-field reconstruction of the yellow area from the second real-world dataset (Actin2).
- real3.png: Large-field reconstruction of the yellow area from the third real-world dataset (ER).
- CC.png: Overlap of the reconstruction images by DLBI (in red) and PALM (in green) (Actin1).

S5.3 Reconstruction quality assessment by SQUIRREL

We further assessed the reconstruction quality of the 3B analysis and DLBI by SQUIRREL (super-resolution quantitative image rating and reporting of error locations), the most recent super-resolution (SR) quality assessment method (Culley *et al.*, 2018). SQUIRREL compares the diffraction-limited image (the reference image) and the corresponding SR equivalents to generate a quantitative map, in which two scores are generated: the resolution-scaled Pearson coefficient (RSP) and the resolution-scaled error (RSE). The higher RSP and lower RSE value outputs, the higher the image quality is. By using the ImageJ plugin of SQUIRREL and operating according to the workflow introduced in Culley *et al.* (2018), we obtained the following quantitative mapping results for the simulated datasets MT0.N1.HD and Tubulin ConjAL647, and the local patches of the real-world datasets Actin1, Actin2 and ER. The corresponding quantitative mapping results are demonstrated in Fig. S7 to Fig. S9.

To apply SQUIRREL, a reference low-resolution image is needed. Here, we built the reference image by summing all the 200 frames of the input low-resolution images and scaling from 60 pixels to 480 pixels based on bi-cubic interpolation. The reference images are illustrated in the first column in Fig. S7, Fig. S8, and Fig. S9. The second and third columns of the figures are the reconstruction results of the 3B analysis and DLBI, respectively. The fourth and fifth columns are the quantitative maps of errors between the reference and the reconstructed images of the 3B analysis and DLBI, respectively. in which the blue color indicates small errors and the yellow color indicates large errors, as described in Culley *et al.* (2018). It can be seen that on both simulated data and real data, the quantitative maps of DLBI always have higher RSP and lower RSE values, which indicate better performance than the 3B analysis. The details of the quantitative maps also support our conclusion, in which the corresponding maps of DLBI have much more blue areas and much less yellow areas.

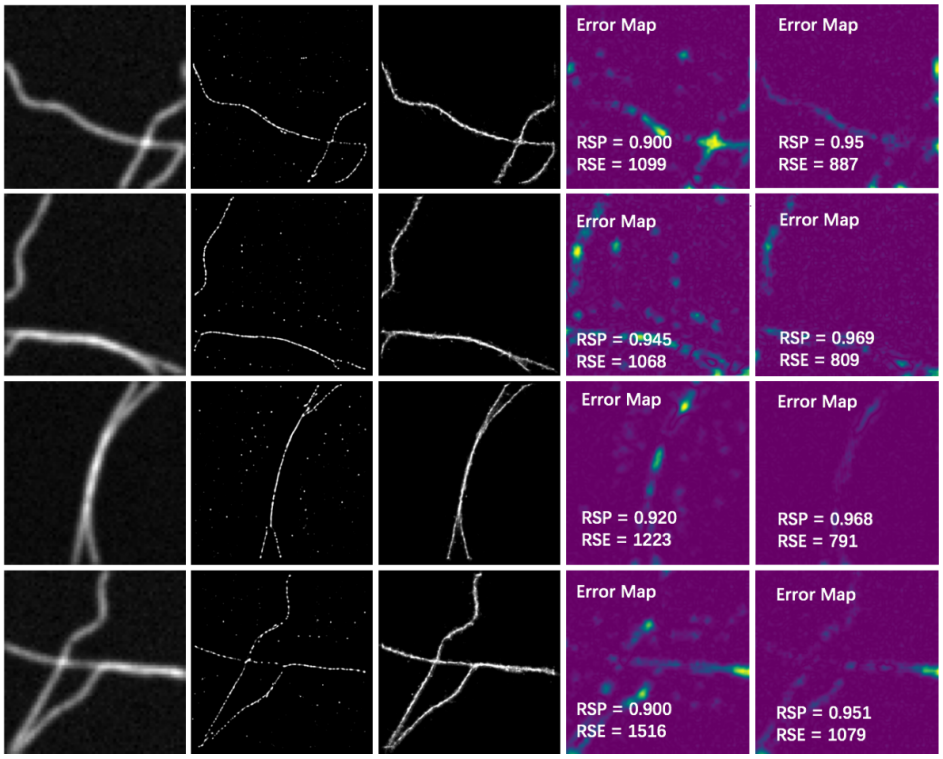


Figure S7: Quantitative mapping results for simulation data MT0.N1.HD.

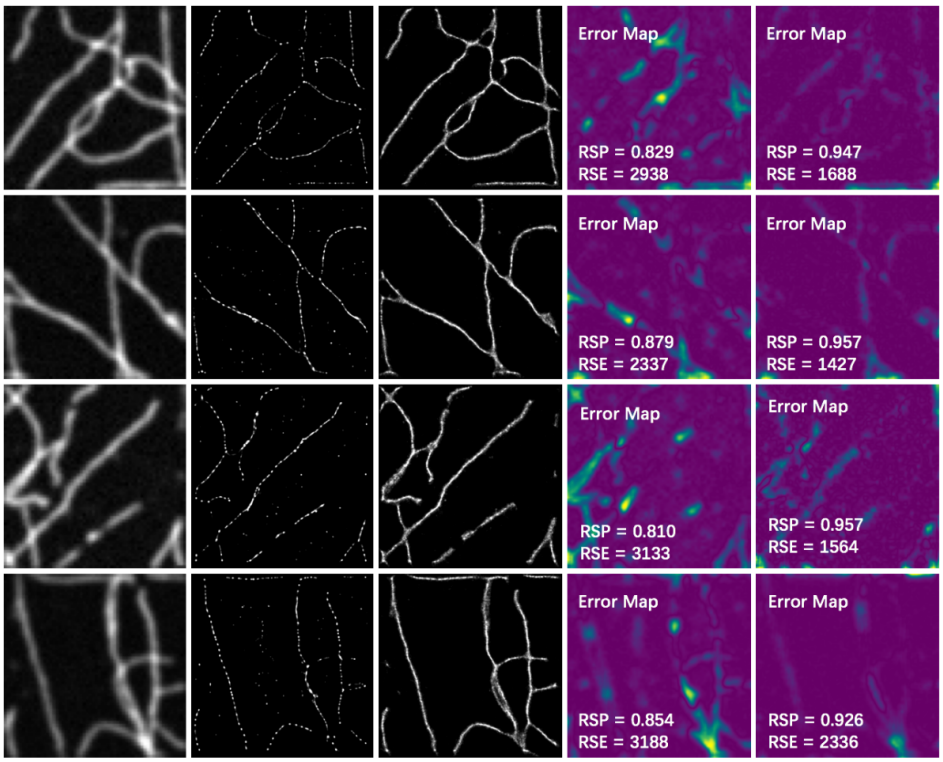


Figure S8: Quantitative mapping results for simulation data Tubulin ConjAL647.

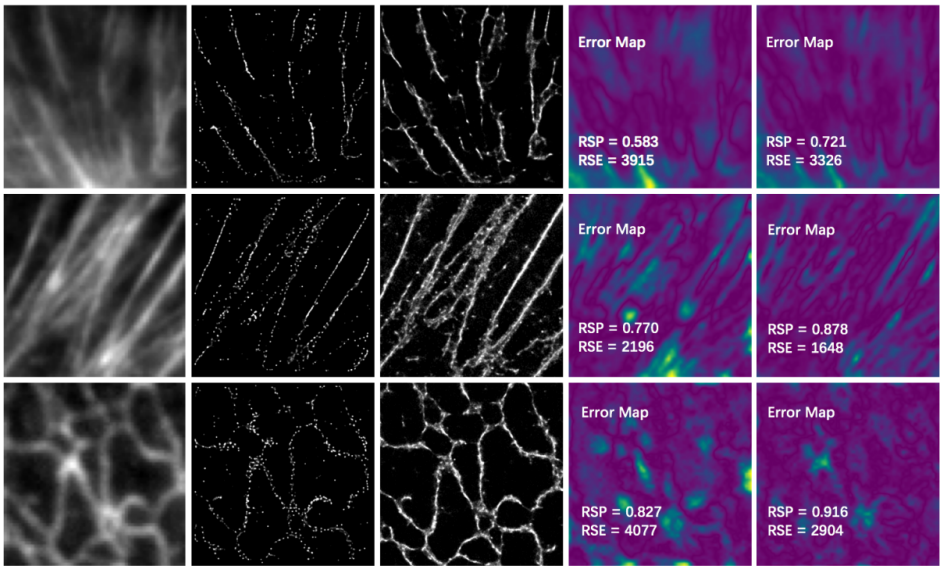


Figure S9: Quantitative mapping results for real data.

References

Cox, S., Rosten, E., Monypenny, J., Jovanovic-Talisman, T., Burnette, D. T., Lippincott-Schwartz, J., Jones, G. E., and Heintzmann, R. (2012). Bayesian localization microscopy reveals nanoscale podosome dynamics. *Nature methods*, **9**(2), 195–200.

Culley, S., Albrecht, D., Jacobs, C., Pereira, P. M., Leterrier, C., Mercer, J., and Henriques, R. (2018). Quantitative mapping and minimization of super-resolution optical imaging artifacts. *Nat. methods*.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 721–741.

Godsill, S. J., Doucet, A., and West, M. (2004). Monte carlo smoothing for nonlinear time series. *Journal of the american statistical association*, **99**(465), 156–168.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456.

Lim, B., Son, S., Kim, H., Nah, S., and Lee, K. M. (2017). Enhanced deep residual networks for single image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, volume 2.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**(2), 257–286.

Zhang, M., Chang, H., Zhang, Y., Yu, J., Wu, L., Ji, W., Chen, J., Liu, B., Lu, J., Liu, Y., *et al.* (2012). Rational design of true monomeric and bright photoactivatable fluorescent proteins. *Nature methods*, **9**(7), 727–729.

Zhu, L., Zhang, W., Elnatan, D., and Huang, B. (2012). Faster storm using compressed sensing. *Nature methods*, **9**(7), 721–723.