Results in Optics 1 (2020) 100019

Contents lists available at ScienceDirect

Results in Optics

journal homepage: www.elsevier.com/locate/rio

Single-molecule localization by voxel-wise regression using convolutional neural network

Toshimitsu Aritake [a,*], Hideitsu Hino [b], Shigeyuki Namiki [c], Daisuke Asanuma [c], Kenzo Hirose [c], Noboru Murata [a]

[a] School of Advanced Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan
[b] The Institute of Statistical Mathematics, 10-3 Midori-cho, Tachikawa, Tokyo 190-8565, Japan
[c] Graduate School of Medicine, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8643, Japan

ARTICLE INFO

Keywords:
3D single-molecule localization microscopy
Multi-focal plane microscopy
Convolutional neural network
Regression-based method

ABSTRACT

Single-molecule localization microscopy is widely used in biological research for measuring the nanostructures of samples smaller than the diffraction limit. In this paper, a novel method for regression of the coordinates of molecules for multifocal plane microscopy is presented. A regression problem for the target space is decomposed into regression problems for small subsets of the target space. Then, a deep neural network is used to solve these problems. By decomposing the regression problem, a fully convolutional neural network can be used to solve the regression problems. The computation of the network is efficient, and a simple and parameter-free loss function can be used to train the network. The proposed algorithm is validated by both simulated and real data obtained by quad-plane microscopy.

1. Introduction

Fluorescence microscopy is widely used in biological research to analyze the structure of in vivo samples. However, due to the diffraction limit of light, the resolution of conventional fluorescence microscopy is limited to approximately 200 nm laterally and 500 nm axially. To overcome this diffraction limit, a number of super-resolution microscopy methods (see Schermelleh et al., 2019 and references therein), including single-molecule localization microscopy (SMLM) (Betzig et al., 2006; Hess et al., 2006), have been proposed. The fundamental problem in super-resolution microscopy is to estimate the true molecular distribution from observed images. In SMLM, only several molecules are activated at a time using photoactivatable molecules; therefore, the positions of the activated molecules can be accurately estimated by a localization algorithm, such as Gaussian fitting. Then, by integrating the localization results of many frames into one image, a high-resolution image can be obtained.

However, the problem of SMLM is its slow imaging speed, and the reasons for this are twofold. First, because fluorescent molecules are sparsely activated in a frame to accurately localize the molecules, a long acquisition time is required to obtain a large number of frames for the reconstruction of high-resolution images. The second reason for the slow imaging speed is that conventional localization methods, such as Gaussian fitting (Betzig et al., 2006; Hess et al., 2006), compressed sensing (Gu et al., 2014), and spline interpolation (Babcock and Zhuang, 2017; Li et al., 2017), require a long computation time to localize molecules, especially when the size of an image is large or the super-resolution scale factor is large. Therefore, a faster localization algorithm that can accurately estimate densely activated molecule locations from observations is necessary to achieve interactive imaging.

Recently, deep neural networks (DNNs) have received increased attention and have been successfully used in a wide variety of applications, including various imaging problems (Lucas et al., 2018), due to their high predictive performance. In the past several years, DNNs have been also applied to SMLM problems, such as molecule localization (Zelger et al., 2018; Ouyang et al., 2018; Nehme et al., 2018; Boyd et al., 2018; Zhang et al., 2018), colorization (Hershko et al., 2019), and background estimation (Möckl et al., 2020). In particular, DNNs achieved molecule localization with a higher accuracy and speed than conventional methods.

Conventional approaches for molecule localization use handcrafted model and estimate molecules locations by solving an optimization problem. The benefit of this approach is that the training of the model is not required. However, it is difficult to model various factors that may affect to the observation process for an accurate estimation. In addition, the computational cost to solve an optimization problem of molecule localization is usually high and therefore a long

* Corresponding author.
E-mail address: toshimitsu.aritake@ruri.waseda.jp (T. Aritake).

https://doi.org/10.1016/j.rio.2020.100019
Received 31 July 2020; Revised 28 October 2020; Accepted 3 November 2020

computation time is required to process a large number of images. On the other hand, DNN based methods use a flexible model and instead of hand-crafting the model, the model is trained by a large amount of data for an accurate molecule localization. Also, although the training a neural network takes several hours to several days, the trained network can estimate the molecules locations efficiently by a combination of simple calculations.

The training of a neural network relies on the availability of a large variety and amount of labeled data, which are pairs consisting of an observation and a target molecule distribution. However, the true molecule distribution of a real observation is generally unknown; therefore, most SMLM algorithms use a reasonable generative model to simulate observations from artificial molecule distributions. A pair consisting of an artificial molecule distribution and an observation generated from the distribution is used to train a neural network; therefore, infinite training data can be generated to train a network.

There are two main types of methods to localize molecules by DNNs: voxel-based methods and regression-based methods. In voxel-based methods, such as Nehme et al. (2018, 2019) and Aritake et al. (2020), a convolutional neural network (CNN) is used to localize the molecules. The target three-dimensional (3D) space is split into voxels at a predefined target resolution. Then, molecules are localized by classifying the voxels into voxels containing a molecule and voxels not containing a molecule. These methods treat molecule localization as a classification problem, and the output of the network is the probability of containing a molecule for each voxel. Because the fluorescence of a molecule spreads around the molecule, a CNN is used as a classifier to localize the molecules for each voxel. Because the target resolution is higher than the resolution of the input images, the network contains upsampling layers to enhance the resolution of the latter. To train the network, the true molecule distribution is also voxelized and compared with the estimated molecule distribution. To compare distributions, the sum of the binary cross-entropy (BCE) for each voxel is used as the objective value to train the network.

The drawback of voxel-based methods is their slow computational speed. When the size of the input image or the target resolution increases, the number of intermediate representations becomes extremely large. Because the convolutional layers used in CNNs involve linear operations, the computation time largely depends on the number of intermediate representations. Another shortcoming of voxel-based methods is that the architecture of the network must be modified and trained each time the target resolution is changed.

Unlike voxel-based methods, DeepLoco (Boyd et al., 2018) is a regression-based method that directly estimates the coordinates of molecules in the target 3D space. DeepLoco consists of feature extraction layers and regression layers. An important difference from voxel-based methods is that the network does not use upsampling layers for feature extraction. Therefore, the number of intermediate representations does not significantly increase due to the input image, and the computational cost is much lower than that of voxel-based methods. In addition, because the locations of molecules are directly estimated as continuous values, the trained network can be used for any target resolution.

However, training a regression-based neural network is not as simple as training a voxel-based neural network. Because the network directly estimates the set of coordinates of molecules, a function is required that quantifies the difference between the set of true coordinates and the set of estimated coordinates. DeepLoco regards molecule distributions as weighted empirical distributions and uses the weighted sample maximum mean discrepancy (MMD) (Gretton et al., 2012) as the difference between two sets of coordinates. However, the difference calculated by the weighted sample MMD depends on the kernel function and its variance parameter. It is difficult to select an optimal kernel and its parameter, however, as the optimal choice varies depending on the density of the molecules and the target resolution. This makes training the network more difficult than for voxel-based methods. Another problem of DeepLoco is that it only accepts fixed-size input. DeepLoco uses convolutional layers such as VGG (Simonyan and Zisserman, 2015) to extract the features of a given image and then uses the extracted features to estimate the coordinates of molecules and the confidence by stacked residual blocks used in a residual neural network (ResNet) (He et al., 2016). A residual block is fully connected; therefore, the size of a feature must be fixed for the regression.

In this work, we propose a novel method to regress molecule locations by a DNN. The basic principle of our method is to decompose a regression of multiple molecules into voxel-wise regressions of a single molecule. The benefit of the decomposition is that training the network becomes much simpler by considering regression problems for only one molecule. The architecture of the proposed network is simple, and commonly used parameter-free loss functions can be used to train the network. An outline of our proposed method is presented in Fig. 1.

In our proposed method, a regression problem is decomposed into two subproblems: the classification of voxels and the regression of molecular coordinates of the voxels. We assume that at most one molecule exists in a voxel in the observed image. Then, we solve two subproblems under this assumption. The first subproblem is the classification of voxels to distinguish voxels in an input image that contain a molecule from voxels that do not contain a molecule. This problem can be solved by the same method used in voxel-based methods. Here, the resolution of a voxel does not need to be enhanced by upsampling; therefore, the computational cost to solve this classification problem is much smaller than that of molecule localization by voxel-based methods. The second subproblem is the regression of the molecule coordinate in a voxel. Because we assume that only one molecule exists in a voxel, we only need to estimate the coordinate of one molecule in the voxel by a regression method.

By decomposing the regression problem into two subproblems, the probability of containing a molecule and the coordinate of the molecule in the voxel are estimated for each voxel. Because fluorescence is spread around the molecule, and it is only necessary to localize at most one molecule in a voxel, we use a CNN to estimate the probability

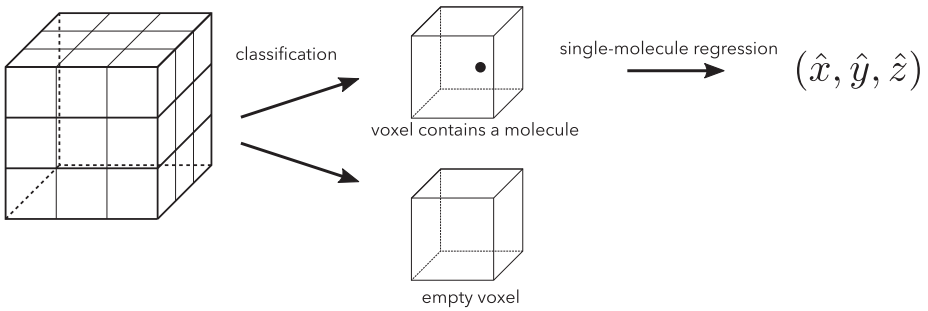


Fig. 1. Outline of proposed method for regressing molecule coordinates.

and coordinate for each voxel. In practice, we can use a single CNN to solve both the classification and regression subproblems simultaneously. In addition, we can use a simple loss function to calculate the objective value for training the neural network. In this work, we use the sum of the BCE and ℓ_1 distance as the loss function for each voxel. These functions are parameter-free and as simple as the objective function in voxel-based methods; Therefore, training the network becomes as simple as in voxel-based methods. However, our method does not require the target resolution parameter and maintains the computational efficiency of regression-based methods.

In this work, we apply our proposed method to an image obtained by multifocal plane microscopy (MUM). Like other localization methods, we use a point spread function (PSF) that approximates the imaging process of MUM to generate artificial data, and we use the artificial data to train our model. Whereas the lateral drift of MUM reduces the accuracy of localization algorithms, our algorithm is robust to lateral drift by being trained on artificial data containing lateral drift.

The contribution of this study are summarized as follows. We propose a novel method to regress the coordinates of molecules in the target space using a CNN. The key principle of our method is to decompose a regression problem into classification and regression problems for each voxel in an observed image. The proposed network is also fully convolutional, and images of any size can be processed by the same network. The comparison with other types of DNN based method is summarize in the Table 1. The source code to train and test our proposed model is available at https://github.com/t-aritake/voxel-wise-regression-SMLM.

The remainder of this paper is organized as follows. In Section 2, we describe the generative model and PSF used in our work. In Section 3, our proposed method, which consists of classification and regression problems, is formulated, and the architecture of the neural network to solve these problems is presented. Then, in Section 4, we validate our method by applying it to both artificial and real datasets. In Section 5, we conclude the paper.

2. Formulation

In this section, we formulate the observation model and PSF used in this work. In this work, we use MUM to observe the fluorescence of molecules, and the fluorescence intensities are observed at several depths in the target space.

2.1. Observation model

A set of images obtained by MUM represent the fluorescence intensities in the target space. Let $\Phi = [0, S_1] \times [0, S_2] \times [0, S_3] \subseteq \mathbb{R}^3$ be a target 3D space that is observed by a microscope. Practically, the observed image is discretized and finite-dimensional; thus, we assume that the target space Φ is discretized into $n_1 \times n_2 \times n_3$ disjoint voxels, where n_1, n_2, and n_3 are the number of bins along the x-, y-, and z-axis, respectively. We also assume that the elements of the observed image are fluorescence intensities observed at the center of the voxels. The voxels are represented as Φ_{ijk} $((i,j,k) \in \Omega)$, where $\Omega = \{(i,j,k) | i \in \{1,2,\dots n_1\} \wedge j \in \{1,2,\dots n_2\} \wedge k \in \{1,2,\dots n_3\}\}$. We assume that all voxels have the same size and that the size of the voxels is $s_1 \times s_2 \times s_3$, where $s_d = S_d / n_d$ $(d = 1,2,3)$. In addition, the coordinate of the center of the voxel Φ_{ijk} is written as $\boldsymbol{x}_{ijk} = (x_{ijk,1}, x_{ijk,2}, x_{ijk,3}) \in \mathbb{R}^3$; therefore, $\Phi_{ijk} = [x_{ijk,1} - s_1/2, x_{ijk,1} + s_1/2] \times [x_{ijk,2} - s_2/2, x_{ijk,2} + s_2/2] \times [x_{ijk,3} - s_3/2, x_{ijk,3} + s_3/2]$.

The fluorescence intensity at each voxel is determined by the coordinate of observation $\boldsymbol{x}_{ijk}$, the coordinates of molecules in the target space, the weights of the molecules, and the PSF, which depends on the optical layout of the microscope. Let $\boldsymbol{y} \in \mathbb{R}_+^{n_1 \times n_2 \times n_3}$ be an observed image and $\boldsymbol{z}_m \in \Phi$ $(m = 1,2,\dots,M)$ be the coordinate of a molecule, where M is the number of molecules in the target space Φ. The weights that control the peak fluorescence intensity of a molecule are represented as $w_m \in [0,1] \subset \mathbb{R}$ $(m = 1,2,\dots,M)$. Then, the fluorescence intensity is determined by these values and the PSF. The PSF is represented by f, and the fluorescence intensity at $\boldsymbol{x}_{ijk}$ is formulated as

$$y_{ijk} = f(\boldsymbol{x}_{ijk}, \{(\boldsymbol{z}_m, w_m)\}_{m=1}^M). \tag{1}$$

The fluorescence intensities of the molecules are disjoint; therefore, y_{ijk} can be written as the linear combination of the fluorescence of each molecule as

$$y_{ijk} \approx \sum_{m=1}^{M} w_m f(\boldsymbol{x}_{ijk}, \boldsymbol{z}_m). \tag{2}$$

In addition, the observation contains shot noise and Gaussian noise; therefore, the observation can be modeled as

$$y_{ijk} = \sum_{m=1}^{M} w_m f(\boldsymbol{x}_{ijk}, \boldsymbol{z}_m) + \varepsilon_{ijk} \quad (i,j,k) \in \Omega, \tag{3}$$

where ε_{ijk} consists of Poisson noise, which represents the shot noise, and Gaussian noise, which represents the observation noise.

2.2. Point spread function (PSF)

In this study, although an observation model is not explicitly used for molecule localization, it is used to generate artificial data to train a model to localize molecules. However, true PSF is generally unknown; therefore, we use an approximate PSF that models the imaging process of the microscope that is used to obtain the observations.

In this paper, we use quad-plane microscopy as MUM, and four cameras are used to observe fluorescence at different depths of the target space Φ. Therefore, hereinafter, $n_3 = 4$ is assumed. The PSF of quad-plane microscopy is modeled by the following function:

$$f(\boldsymbol{x}_{ijk}, \boldsymbol{z}_m) = a(x_{ijk,3} - z_{m,3}^h) \exp\left(-\frac{(x_{ijk,1} - z_{m,1})^2 + (x_{ijk,2} - z_{m,2})^2}{2(r(x_{ijk,3} - z_{m,3}^h))^2} \right) + b, \tag{4}$$

where $a(x)$ is the peak fluorescence intensity, $r(x)$ is the width of fluorescence, and b is the background fluorescence intensity. This PSF is similar to the PSF used in Gu et al. (2014) for biplane microscopy and models the distribution of fluorescence as a Gaussian function. The width of the PSF $r(x_3)$ varies depending on the distance between the molecule and the focal planes, and is described by the following defocus curve:

$$r(x) = r_0 \sqrt{1 + \left(\frac{x}{D}\right)^2 + A\left(\frac{x}{D}\right)^3 + B\left(\frac{x}{D}\right)^4}, \tag{5}$$

where r_0 is the width of the PSF when a molecule is on the focal plane, and D is the focus depth of the microscope. The peak $a(x)$ of the PSF depends on the width $r(x)$, and is modeled as

$$a(x) = \frac{a'}{2\pi r(x)^2}. \tag{6}$$

Table 1
The comparison with other types of neural network methods.

	Voxel-based	DeepLoco	Proposed
Type of the output	discrete	**continuous**	**continuous**
Parameter of the loss function	**not required**	required	**not required**
Computational Speed	slow	**fast**	**fast**
Input size	**any**	fixed	**any**

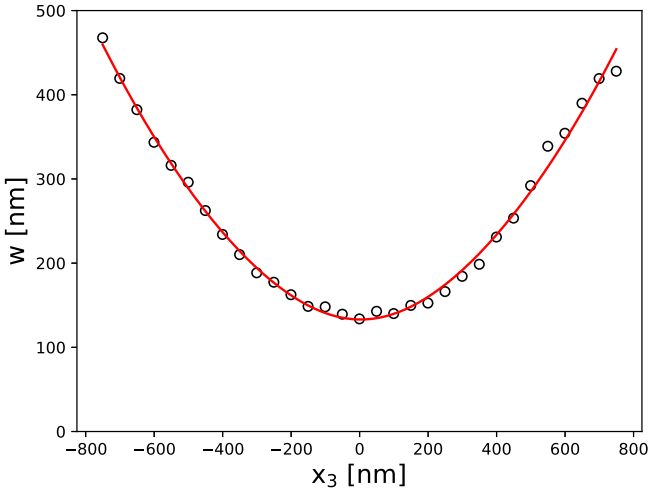


Fig. 2. Width of observed fluorescent beads and values of the defocus curve. The observed width of the fluorescent beads w at each depth x_3 are represented by a circle. The red line represents the value of the defocus curve to approximate the width with the parameters in Table 2. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Parameters of the point spread function.

Parameters	Value
a'	5.00×10^7
b	0
r_0	1.33×10^2
D	3.02×10^2
A	7.37×10^{-4}
B	6.27×10^{-3}

Fig. 2 presents the width of the observed fluorescent beads and the value of the defocus curve (5). The parameters of the PSF is decided as Table 2 so that the PSF well approximates the width and peak intensity of the observed fluorescence of the fluorescent beads.

In addition, observation by MUM is affected by the lateral drift of the camera positions. The amount of lateral drift is identical for each focal plane, and we denote the amount of lateral drift of the kth focal plane along the horizontal and vertical axis by $\Delta_k = (\Delta_{k1}, \Delta_{k2}, 0) \in \mathbb{R}^3$. When lateral drift is considered, the observed fluorescence at the coordinate $\boldsymbol{x}_{ijk}$ from a molecule located at z_m is modeled as $f(\boldsymbol{x}_{ijk} - \boldsymbol{\Delta}_k, z_m)$ using PSF. As the PSF is defined in Eq. (4), the following equation holds:

$$f(\boldsymbol{x}_{ijk} - \boldsymbol{\Delta}_k, z_m) = f(\boldsymbol{x}_{ijk}, z_m + \boldsymbol{\Delta}_k). \tag{7}$$

This equation implies that when the camera positions have lateral drift $\boldsymbol{\Delta}_k$, the fluorescence observed at $\boldsymbol{x}_{ijk}$ from the molecule at z_m is identical to the fluorescence observed at $\boldsymbol{x}_{ijk}$ from the molecule at $z_m + \boldsymbol{\Delta}_k$. Therefore, the lateral drift of cameras must be considered for accurate molecule localization. In practice, because the absolute amount of drift cannot be estimated from observation, we consider the relative amount of drift $\boldsymbol{\Delta}_{k'}$ $(k = 2, 3, 4)$ from the reference focal plane $k = 1$.

3. Method

3.1. Molecule localization by regression-based method

The problem of molecule localization is to estimate the coordinates of molecules in the target space. In this work, we solve the molecule localization problem as a regression problem. Namely, the coordinates of molecules z_m $(m = 1, 2, \ldots, M)$ are directly estimated by regression, and $\hat{z_m}$ $(m = 1, 2, \ldots, \hat{M})$ denote the estimated coordinates of molecules, where $\hat{M}$ is the estimated number of molecules. To estimate the coordinates of molecules from an observed image $\boldsymbol{y}$, we use a regression model g parameterized by $\boldsymbol{\theta}$; therefore, $\{\hat{z_m}\}_{m=1}^{\hat{M}} = g(\boldsymbol{y}; \boldsymbol{\theta})$.

The accuracy of the regression model g depends on the parameter $\boldsymbol{\theta}$. To train the model g for accurate regression, we solve the following minimization problem:

$$\text{minimize}_\theta L(\{\mathbf{z}_m\}_{\mathrm{m}=\mathbb{1}}^{\mathrm{M}}, \{\hat{\mathbf{z}}_m\}_{\mathrm{m}=\mathbb{1}}^{\hat{\mathrm{M}}}), \tag{8}$$

where L is the loss function to quantify the difference between the set of true coordinates of molecules $\{\mathbf{z}_m\}_{\mathrm{m}=1}^{\mathrm{M}}$ and the estimated coordinates of molecules $\{\hat{\mathbf{z}}_{\mathrm{m}}\}_{\mathrm{m}=1}^{\mathrm{M}}$. If $M = \hat{M} = 1$, simple loss functions such as the ℓ_1 distance or ℓ_2 distance can be used as L. However, in general, $M, \hat{M} > 1$ and $M \neq \hat{M}$, and the sets $\{\mathbf{z}_m\}_{\mathrm{m}=\mathbb{1}}^{\mathrm{M}}, \{\hat{\mathbf{z}_{\mathrm{m}}}\}_{\mathrm{m}=\mathbb{1}}^{\hat{\mathrm{M}}}$ represent the distribution of molecules in the target space. Therefore, the loss function L must satisfy the following conditions to quantify the difference between the two sets. First, L must be defined between two sets. Because different sets have a different number of elements, and the elements of a set are generally not ordered, the loss function L must directly quantify the difference between the two distributions of molecules in the target space. Second, the loss function must be differentiable to train the model g by gradient-based methods, which can be easily applied for training.

By estimating the confidence of regression with the coordinates, $\hat{M}$ can be set to a sufficiently larger number than the true number of molecules M. However, the choice of the loss function is limited due to the conditions mentioned above. For example, DeepLoco (Boyd et al., 2018) uses the weighted sample MMD (Gretton et al., 2012) as the loss function. However, it is difficult to identify a loss function that is parameter-free, easy to calculate, and easy to minimize for training a model. To overcome this difficulty, we avoid the direct estimation of the molecule coordinates $\{\hat{\mathbf{z}}_{\mathrm{m}}\}_{\mathrm{m}=1}^{\hat{\mathrm{M}}}$ in the target 3D space, and allow a simple loss function to be used to train the model.

3.2. Proposed method

In this work, instead of solving a regression problem to estimate the coordinates of multiple molecules in the entire target space, we decompose the regression problem into a regression problem for each voxel Φ_{ijk} $((i, j, k) \in \Omega)$. Namely, a single regression problem of an entire image is decomposed into multiple regression problems of small regions of the target space. Then, we assume that at most one molecule is contained in a single voxel. This assumption is reasonable because fluorescent molecules are sparsely activated; therefore, multiple molecules are rarely contained in a single voxel. Under this assumption, a regression problem for a single molecule is solved for each voxel. Therefore, a simple loss function can be used as the loss function to train the regression models for the voxels.

In practice, if a molecule is not contained in a voxel, the regression problem for the voxel cannot be solved, as the target variable for the regression problem does not exist. Therefore, we solve the regression problem for a voxel as the composition of two subproblems. The first subproblem is a binary classification problem to distinguish whether a voxel contains a molecule, while the second subproblem is a regression problem for the voxel. The second problem must be solved for voxels that contain a molecule.

Let c_{ijk} be a binary value that is a label of the binary classification subproblem, where

$$c_{ijk} = \begin{cases} 1 & \exists m \in \{1, 2, \ldots M\}, z_m \in \Phi_{ijk}, \\ 0 & \text{otherwise.} \end{cases}$$

We consider label c_{ijk} to be positive if $c_{ijk} = 1$ and negative otherwise, and voxel Φ_{ijk} is considered a positive or negative voxel according to its label. We also define a set of indices of positive voxels as $\Omega_+ = \{(i,j,k)|c_{ijk} = 1\} \subseteq \Omega$ and a set of indices of negative voxels as $\Omega_- = \Omega \setminus \Omega_+$. In addition, for a regression for a voxel, the coordinate of a molecule z_m must be converted to the coordinate of a voxel. Assuming $z_m \in \Phi_{ijk}$, the coordinate of the molecule in voxel Φ_{ijk} is the coordinate relative to the size of the voxel, and is represented as $\boldsymbol{\zeta}_{ijk} = (\zeta_{ijk,1}, \zeta_{ijk,2}, \zeta_{ijk,3})$, where

$$\zeta_{ijk,d} = \frac{z_{ijk,d} - x_{ijk,d}}{s_d} + \frac{1}{2} \in [0,1] \quad (d = 1,2,3).$$

As evident in this equation, the coordinate z_{ijk} can be restored from $\boldsymbol{\zeta}_{ijk}$ and $\boldsymbol{x}_{ijk}$. Note that the true coordinates of molecules $\boldsymbol{\zeta}_{ijk}$ are defined only for voxels that contain a molecule, namely, for all voxels Φ_{ijk} $((i,j,k) \in \Omega_+)$.

In summary, it is necessary to estimate the label of voxels in an observed image $\boldsymbol{y}$ and the relative coordinates of the voxels containing a molecule. We represent the estimated label and coordinate for voxel Φ_{ijk} by $\hat{c}_{ijk}$ and $\hat{\boldsymbol{\zeta}}_{ijk} = (\hat{\zeta}_{ijk,1}, \hat{\zeta}_{ijk,2}, \hat{\zeta}_{ijk,3})$, respectively. Then, a set of four values $\hat{\boldsymbol{\eta}}_{ijk} = (\hat{c}_{ijk}, \hat{\zeta}_{ijk,1}, \hat{\zeta}_{ijk,2}, \hat{\zeta}_{ijk,3})$ is estimated for each voxel Φ_{ijk} $((i,j,k) \in \Omega)$ by the model g as follows:

$$\hat{\boldsymbol{\eta}} = g(\boldsymbol{y}; \theta), \tag{9}$$

where $\hat{\boldsymbol{\eta}} = (\boldsymbol{\eta}_{ijk})_{(i,j,k)\in\Omega} \in \mathbb{R}^{n_1 \times n_2 \times 4 \times 4}$ is an array of estimated values. Here, instead of estimating the binary label, our model estimates the positive label probability as $\hat{c}_{ijk} \in [0,1]$ for each voxel. It should be noted that although the coordinates of molecules are estimated for all voxels, the estimation is reliable only when the voxel contains a molecule. Therefore, the probability $\hat{c}_{ijk}$ can also be seen as the confidence of the estimated coordinate $\hat{\boldsymbol{\zeta}}_{ijk}$.

The model g is trained by minimizing the following loss function:

$$L(\eta, \hat{\eta}) = \sum_{(i,j,k)\in\Omega} L^{\mathrm{BCE}}(c_{ijk}, \hat{c}_{ijk}) + \sum_{(i,j,k)\in\Omega_+} L^{\mathrm{pos}}(\boldsymbol{\zeta}_{ijk}, \hat{\boldsymbol{\zeta}}_{ijk}), \tag{10}$$

where $\hat{c}_{ijk}$ and $\hat{\zeta}_{ijk}$ are estimated by g as (9), and the parameters $\boldsymbol{\theta}$ of g are optimized to minimize the loss function. Here, L_i^{BCE} is the BCE for the classification subproblem:

$$L^{\mathrm{BCE}}(c_{ijk}, \hat{c}_{ijk}) = c_{ijk} \log \hat{c}_{ijk} + (1 - c_{ijk}) \log(1 - \hat{c}_{ijk}),$$

and L^{pos} is the loss function for the regression subproblem in each voxel. Because the true molecule coordinate $\boldsymbol{\zeta}_{ijk}$ is defined only for voxels that contain a molecule, this loss is calculated only for voxel Φ_{ijk} $((i,j,k) \in \Omega_+)$. Because we assume that only one molecule exists in a voxel Φ_{ijk} $((i,j,k) \in \Omega_+)$, we use the ℓ_1 loss function as follows:

$$L^{\mathrm{pos}}(\boldsymbol{\zeta}_{ijk}, \hat{\boldsymbol{\zeta}}_{ijk}) = \|\boldsymbol{\zeta}_{ijk} - \hat{\boldsymbol{\zeta}}_{ijk}\|_1.$$

Training the model involves the class imbalance problem. Because fluorescent molecules are sparsely activated in the target space, the number of positive voxels is much smaller than the number of negative voxels. Therefore, the sum of the loss of negative voxels,

$$\sum_{(i,j,k)\in\Omega_-} L^{\mathrm{BCE}}(c_{ijk}, \hat{c}_{ijk}),$$

is much larger than the sum of the loss of positive voxels,

$$\sum_{(i,j,k)\in\Omega_+} (L^{\mathrm{BCE}}(c_{ijk}, \hat{c}_{ijk}) + L^{\mathrm{pos}}(\boldsymbol{\zeta}_{ijk}, \hat{\boldsymbol{\zeta}}_{ijk})).$$

Therefore, a model trained by minimizing this loss function tends to estimate the label probabilities as close to 0 even for positive voxels. To address this problem, we use hard negative mining, which uses only a part of the loss of negative voxels. This technique has been recently used in learning a neural network for object detection (Liu et al., 2016), and allows the trained model to perform more accurate estimation for positive voxels. Let π be the pth largest loss of negative voxels $L^{\mathrm{BCE}}(c_{ijk}, \hat{c}_{ijk})$ $((i,j,k) \in \Omega_-)$, where the confidence $\hat{c}_{ijk}$ is estimated by the model g with the current parameters. Then, we solve the following minimization problem to train g:

$$\mathrm{minimize}_\theta \sum_{(i,j,k)\in\Omega\prime} L^{\mathrm{BCE}}(c_{ijk}, \hat{c}_{ijk}) + \sum_{(i,j,k)\in\Omega_+} L^{\mathrm{pos}}(\boldsymbol{\zeta}_{ijk}, \hat{\boldsymbol{\zeta}}_{ijk}), \tag{11}$$

where $\Omega\prime = \Omega_+ \cup \{(i,j,k) \in \Omega_- | L^{\mathrm{BCE}}(c_{ijk}, \hat{c}_{ijk}) \geqslant \pi\}$.

In this work, we use a CNN as the regression model $g(\boldsymbol{y}; \boldsymbol{\theta})$, and the architecture of the model g is presented in Fig. 3. As the figure indicates, our model is a simple fully convolutional neural network, and the label probability and coordinate are estimated for each voxel. Because the fluorescence of a molecule spreads around the molecule, the values of a voxel can be estimated from fluorescence around the voxel using convolution operations. In the first two layers, 5×5 convolutions are used to extract features, while 3×3 convolutions are used in the remaining layers. Here, the width and height of intermediate representations are identical to those of the input image, and only the number of channels is changed in each layer. Then, in each layer of the network except for the last layer, the rectified linear unit (ReLU) is used as the activation function, followed by batch normalization (Ioffe and Szegedy, 2015). In the last layer, a sigmoid function is used as the activation function, as the class probability and the values of the relative coordinate are elements in $[0,1]$. Finally, the array of size $n_1 \times n_2 \times 16$ is resized to $n_1 \times n_2 \times 4 \times 4$ so that the network outputs four values for each voxel of the input image.

Although, we use the 8 layered simple CNN, shallower or deeper networks can also be considered. If we use shallower network, computation become slightly faster and the training of the network becomes easier. However, it is difficult to extract appropriate features for the classification and the regression from the input image by shallow networks. On the other hand, deeper networks are able to extract more complex features than shallow networks. Instead, the computation become slightly slower, and the training of the network becomes difficult and requires more training data. Also, more complex networks such as U-net used in Nehme et al. (2018) can also be used.

The benefit of this model is that an input image of any width and height can be processed because the model g is fully convolutional. In addition, the number of elements of intermediate representations is increased only by changing the number of channels; therefore, estimation by the network is computationally and memory efficient.

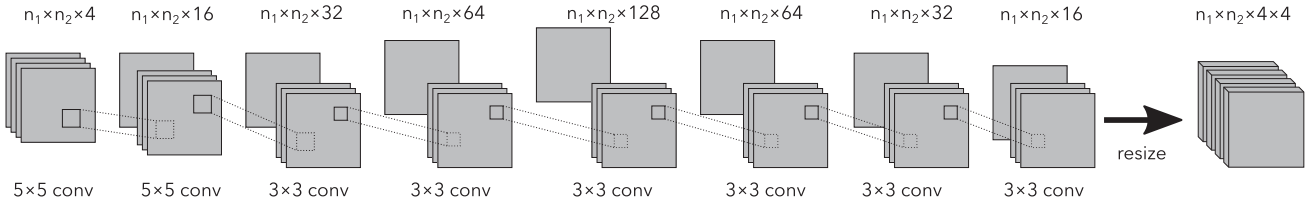


Fig. 3. Architecture of a convolutional neural network. The activation function used in this model is the rectified linear unit (ReLU) and batch normalization except for the last layer. In the last layer, sigmoid activation is used to predict the four values $(\hat{c}_{ijk}, \hat{\zeta}_{ijk,1}, \hat{\zeta}_{ijk,2}, \hat{\zeta}_{ijk,3}) \in [0,1]^4$.

The optimal parameter $\boldsymbol{\theta}$ of the network is different according to the optical layout of the microscope. Then, the user needs to train the network so that the output of the network is close to the actual class labels and the coordinates before applying to the real samples. We use artificial data generated from the observation model described in Section 2 to train the network g. We determine the number of molecules M and sample the coordinates of the molecules from a uniform distribution of the target space Φ. These coordinates are converted to the label and coordinates of the voxel; then, these values are used as target values for the regressions. A set of observed images at each focal plane is calculated using the observation model. When calculating the sets of observed images, the relative lateral drift $\boldsymbol{\Delta}_k\prime$ $(k = 2, 3, 4)$ is also randomly determined for each sample. By training a neural network using these data, the network becomes robust to the lateral drift of camera positions.

We note that when the optical layout of the microscope is changed, the network needs to be retrained. For example, when the type of optical lenses are changed, the PSF of the optical system is also changed. Then the network needs to be retrained by the new training data generated from new observation model. However, when the optical system is only slightly changed, only small number of training data is required for fine-tuning the network.

4. Experiments

In this section, we present the experimental results of localization by the proposed method using both artificial images and real microscopy images. In the following experiments, the images observed by the microscope described below were processed on an NVidia Tesla V100 32 GB graphics processing unit (GPU) to localize molecules in the images.

4.1. Optical layout of microscope

In this subsection, we present the experimental settings of the microscope used in the experiments. The microscope settings were the same as those used in prior work (Aritake et al., 2020). A multi-focus microscope equipped with four EM-CCD cameras was constructed based on a commercial inverted microscope, as illustrated in Fig. 4. A 640-nm laser beam was focused on the back focal plane of a 100× oil immersion objective lens to illuminate an Alexa Fluor 647-stained specimen at an excitation intensity of approximately 5 kW/cm^2. The fluorescence emitted from the specimen was collected by the same objective lens. The fluorescence image formed by the internal tube lens of the inverted microscope was relayed by an achromatic lens (f = 125.0 mm), split twice by 1:1 beam-splitter mirrors, and refocused onto the four cameras via achromatic lenses (f = 100.0 mm, Thorlabs). The axial positions of the achromatic lenses in front of the cameras were adjusted so that the four planes were at 400-nm intervals in the Z-axis direction of the specimen corresponding to the conjugate planes of the sensor surface of the respective cameras. The difference in the field of view of the cameras was corrected by coordinate registration using affine transformation, whose parameters were determined by images of multiple fluorescent beads captured on different cameras.

4.2. Experiments with artificial data

We used 100,000 artificial frames generated by the observation model to train the network. Each frame consisted of four images that approximated quad-plane microscopy observations. The parameters of the experiments are presented in Table 3. The coordinates of molecules z_m $(m = 1, 2, \ldots M)$ were uniformly randomly sampled from the target space Φ, and the weights of the molecules $\mathbf{w}_m$ $(m = 1, 2, \ldots M)$ were also randomly sampled from a uniform distribution in [0.3, 1.0]. In addition, we used random lateral drift $\boldsymbol{\Delta}_k\prime = (\Delta_{k1}\prime, \Delta_{k2}\prime, 0)$ $(k = 2, 3, 4)$ for each observation. The observed value was then calculated by (3). An example of the training data is presented in Fig. 5. We used Adam (Kingma and Ba, 2015) as the optimizer, where the parameters were $\beta_1 = 0.9$ and $\beta_2 = 0.99$, the initial learning rate was set to 1.0×10^{-3}, and the batch size was 50. The epoch number of the optimization was 20, and the dataset was randomly shuffled at the end of each epoch. Our method does not require adjustment of the learning rate based on the number of epochs.

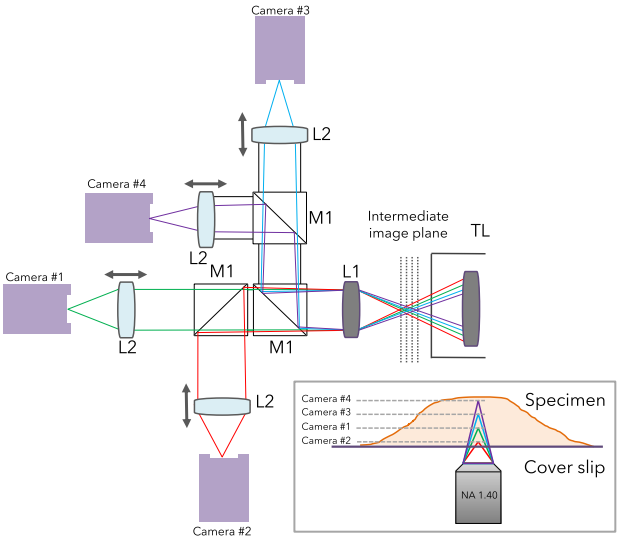


Fig. 4. Optical layout of quad-plane microscope. The intermediate image is relayed onto each camera via a pair of lenses (L1, f = 125.0 mm; L2, f = 100.0 mm). TL, tube lens; M1, 1:1 beam-splitter mirror. The inset displays the focusing planes of four cameras.

Table 3
Parameters for experiments with artificial data.

Parameter	Value
Target space size	$S_1 = S_2 = 12,288$ [nm], $S_3 = 1,600$ [nm]
Resolution of a voxel	$s_1 = s_2 = 192$ [nm], $s_3 = 400$ [nm]
Size of images	$n_1 = n_2 = 64, n_3 = 4$
Number of molecules	$M = 50$
Variance of Gaussian noise	$\sigma^2 = 9$
Rate parameter of Poisson noise	$\lambda = 10$

To validate the accuracy of the trained network, we used artificial test images that were generated in the same way as the training data. In this experiment, we estimated that a molecule existed in a voxel Φ_{ijk} if the label probability $\hat{c}_{ijk}$ exceeded a threshold value τ, and we used $\tau = 0.5$ or $\tau = 0.9$. Fig. 6 provides the localization results that were estimated by the trained network. As the figure indicates, most of the molecules in the target space were localized close to the true coordinates for both threshold values. Unlike the voxel-based method which localize the molecule at predefined target resolution, the coordinates of molecules are localized as continuous values. Although some molecules were not localized for $\tau = 0.9$, the localization results were almost identical for both threshold values; thus, most molecules were localized accurately.

Fig. 7(a) presents the Jaccard index of binary classification for various molecule densities. We used 1,000 artificial images, and the number of activated molecules differed for each density. As seen in Fig. 7 (a), the Jaccard index of binary classification slightly decreased as the density of activated molecules increased. This demonstrates that the

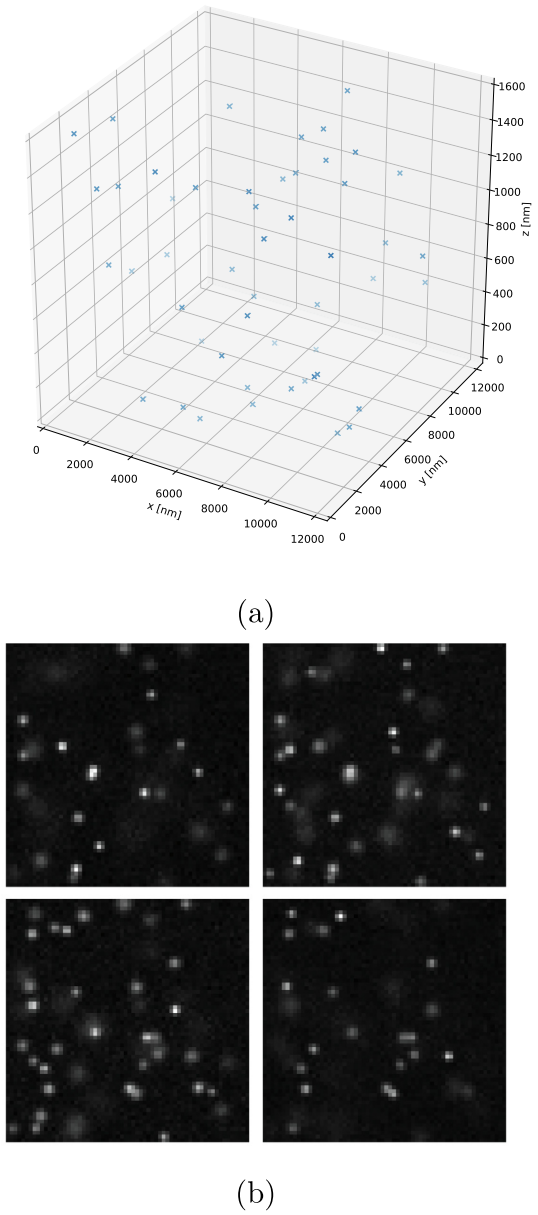


Fig. 5. (a) Example of artificial molecule distribution containing 50 molecules in the target space. The true locations of molecules are represented by blue crosses. A list of coordinates of the molecules was used to train the network. (b) Observed image simulated from the artificial distribution. The size of the image was $64 \times 64 \times 4$.

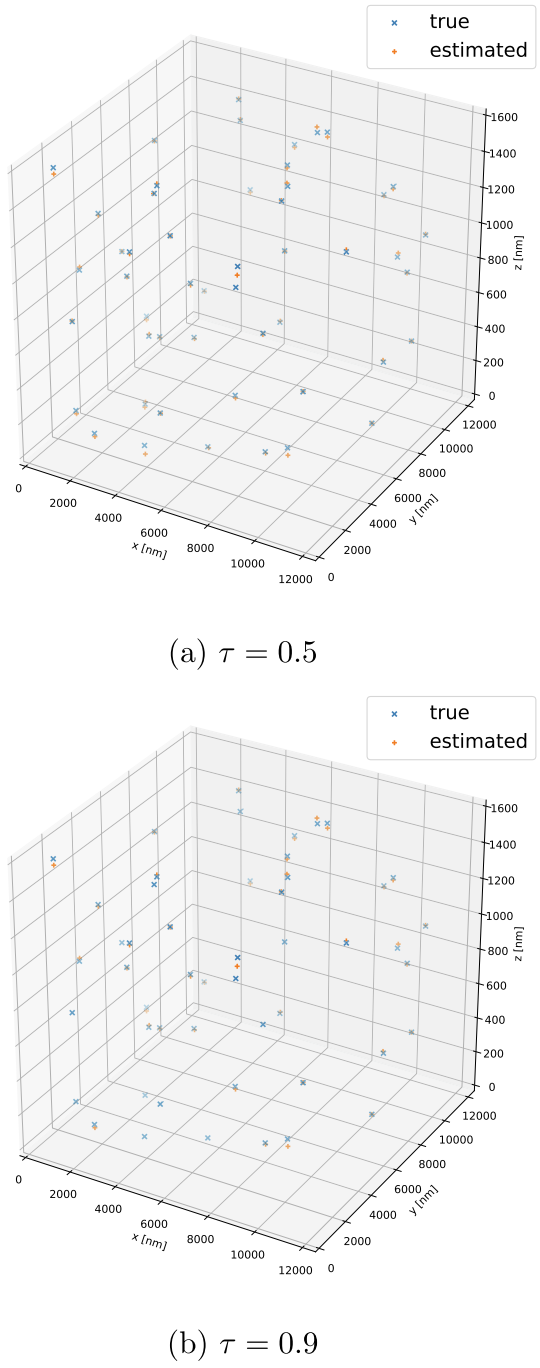


Fig. 6. Examples of molecule distribution of test data and molecule distribution estimated by the trained network with the threshold values (a) $\tau = 0.5$ and (b) $\tau = 0.9$. Blue crosses represent the true molecule coordinates, while orange plus symbols represent the estimated coordinates. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

binary classification became more difficult as the density increased due to the overlap of fluorescence. In practice, as the density of the molecules increased, the number of false-positive (FP) cases increased more than the number of false-negative (FN) cases because close molecules are difficult to localize correctly. However, as Fig. 7(b) indicates, the average distance from a localized molecule to its nearest molecule along each axis was close to zero for all densities. Although the average error along each axis was slightly biased, this is possibly because the artificial training data did not have enough variety to alleviate the effect of lateral drifts in average. In addition, the errorbar of Fig. 7(b) shows the standard deviation of the error and Fig. 8 presents the distribution of the errors along each axis for different thresholding value and molecule density. Here, we selected the distribution of the minimum and maximum density in Fig. 7(b). As seen in the figures, the variance of the distribution increased as the molecule density increased and the errors along each axis followed a Poisson-like distribution. These results suggest that even when voxels were not completely correctly classified, molecules could be localized from adjacent voxels.

Table 4 presents a confusion matrix of the binary classification of voxels for a threshold value of 0.5. The label probability estimated by the trained network was binarized using the threshold value. We used 500 artificial images for this experiment, and 50 molecules were activated for each image. Therefore, a confusion matrix of a total of

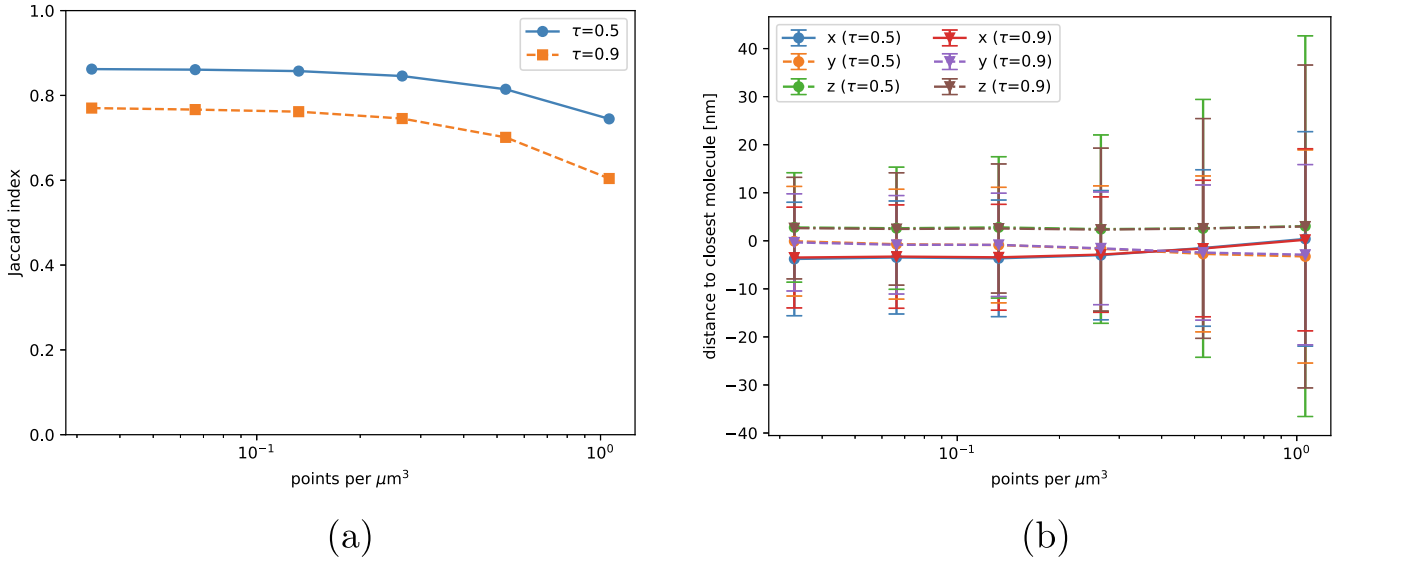


Fig. 7. Error of binary classification and error of regression for various molecule densities. (a) Jaccard index of binary classification; (b) average and standard deviation of the error between the estimated coordinate and the nearest molecule.

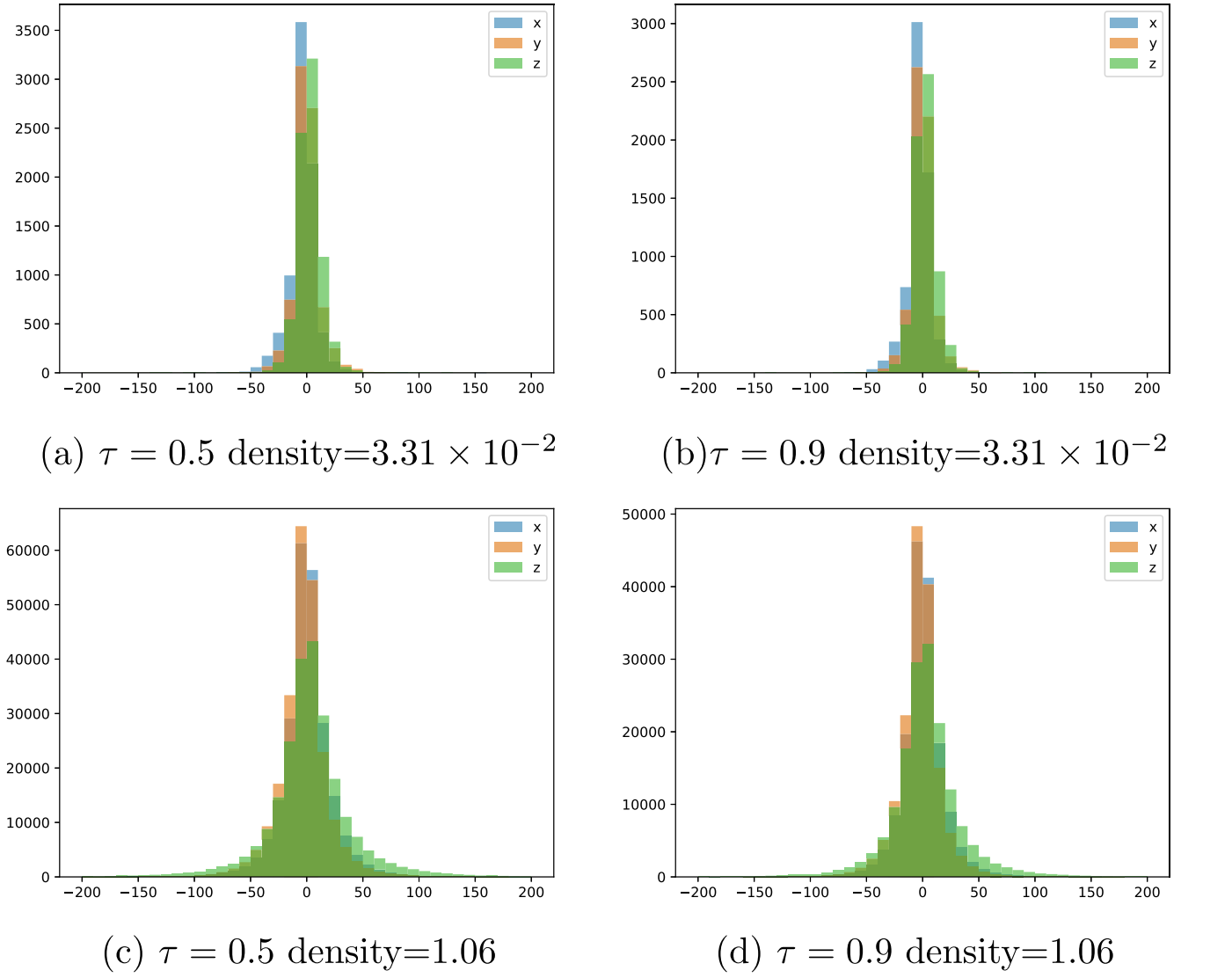


Fig. 8. Histograms of errors of regression for different density. The variance of the distribution mainly depends on the density of the molecules. Especially the errors along z-axis are Poisson distributed and the variance of the distribution becomes large when the molecule density is high.

Table 4
Confusion matrix of binary classification by the trained network. The estimated label was binarized using a threshold value of $\tau = 0.5$.

		True label	
		Positive	Negative
Estimated label	Positive	22,878	1,844
	Negative	2,089	8,165,189

Table 5
False positive and false negative cases considering the adjacent voxels of false positive and false negative voxels.

	Adjacent1	Adjacent2	Non-adjacent
False Positive	1,399	445	0
False Negative		318	372

8,192,000 voxels is presented in Table 4. When the coordinate of a molecule was close to the boundary of a voxel, the network sometimes failed to estimate the correct label probability of the voxel. Instead, the network estimated that a molecule existed in the neighboring voxel closest to the true molecule location.

Additional details of the FP and FN cases are listed in Table 5. Adjacent1 lists the number of FP and FN voxels for which an FN voxel and its adjacent FP voxel had a one-to-one correspondence. In this case, a molecule was localized from the adjacent FP voxel. Similarly, Adjacent2 for FP indicates the number of FP voxels that were adjacent to TP or FN voxels. In this case, FP voxels could be adjacent to multiple FN or TP voxels. In addition, Adjacent2 for FN presents the number of FN voxels that were adjacent to voxels whose labels were estimated to be positive. Then, Non-adjacent for FP indicates the number of FP voxels that were not adjacent to TP or FN voxels, and Non-adjacent for FN indicates the number of FN voxels that were not adjacent to TP or FP voxels. As illustrated in Table 5, there were no FP voxels that were not adjacent to voxels that contained a molecule. This result suggests that because the existence probabilities of molecules were estimated from the intensity of molecules, FP voxels only appeared close to voxels that contained a molecule.

The processing speed of the network is displayed in Fig. 9. The computation time consisted of the image loading time, image transfer time to a GPU, and computation time on the GPU. As indicated in the figure, the computational speed of our proposed method was as fast as that of DeepLoco and faster than that of the voxel-based method, especially when large images were processed. Our method achieved 200 fps for the localization of molecules including data loading and data transfer; thus, our method is readily applicable to the real-time localization of molecules. In addition, images of various sizes can be processed by the same network.

4.3. Experiments with real data

In this subsection, we present the experimental results of real data of observed microtubules using the microscope described in Section 4.1. Methanol-fixed COS7 cells were used for stochastic optical reconstruction microscopy (STORM) imaging of tubulin molecules expressed inside the cells, as described in a previous study (Cleveland and Sullivan, 1985). Images were acquired at 22 Hz with 20-ms exposure. The observed images were preprocessed by an affine transformation so that the pixel-level translations of the images were removed. However, it should be noted that sub-pixel-level translation remained in the observed images because the lateral drift of the camera positions affected the observation. The size of the voxels in the observed images was the same as in Section 4.2. An example of a frame is presented in Fig. 10. Note that the fluorescence intensity from beads that are used as markers are blighter than the fluorescence from samples; hence, the fluorescence from beads are masked to zeros for better visibility of fluorescence from specimen. The dataset consisted of 30,000 frames, and each frame consisted of 256×256 images obtained from four focal planes. Therefore, the size of each frame was $n_1 = n_2 = 256$ and $n_3 = 4$. Each frame was processed independently to localize the molecules.

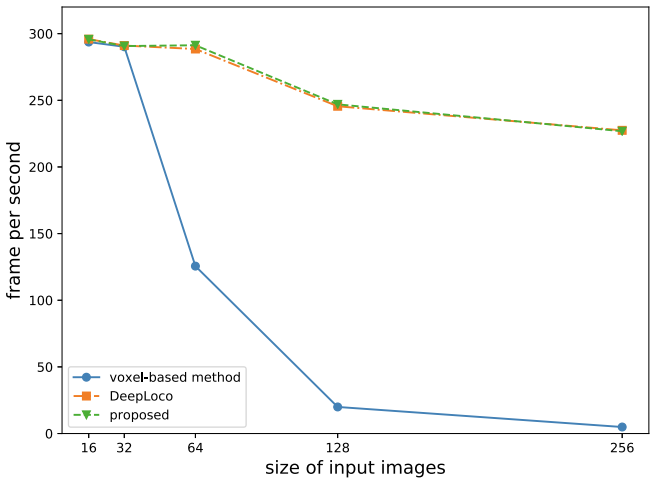


Fig. 9. Comparison of computational speed of deep-learning-based methods for image sizes of $16 \times 16 \times 4, 32 \times 32 \times 4, \ldots, 256 \times 256 \times 4$. The computation time consisted of the time to load the images, time to transfer them to graphics processing units (GPUs), and time to predict the coordinates and confidence.

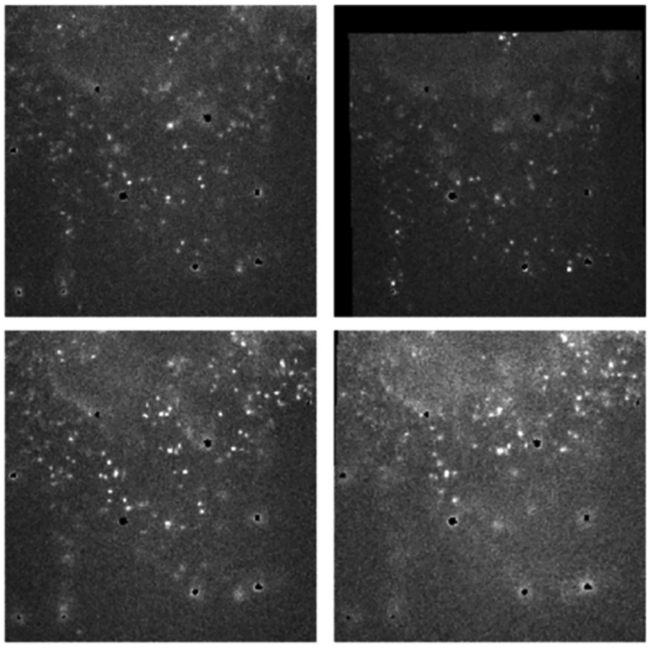
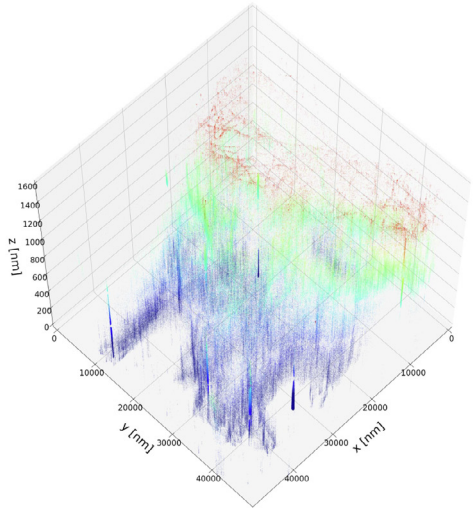
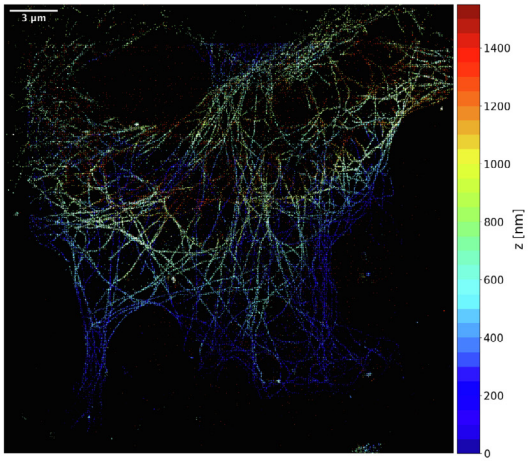


Fig. 10. Example of a frame with observed microtubules. The observed image was a $256 \times 256 \times 4$ image, and the difference in the field of view of the cameras was corrected by affine transformation. In the above figure, the fluorescence from fluorescent beads are masked to zero and images are normalized for better visibility of fluorescence from the microtubules.

Fig. 11(a) presents the 3D coordinates of molecules estimated from all frames. Here, we used the threshold value $\tau = 0.9$. Fig. 11(b) presents a high-resolution image generated by discretizing the estimated coordinates. Here, we set the super-resolution scale factor along each axis to 8 and the voxel size of the high-resolution images to $24 \times 24 \times 50$ [nm]. The difference in depth is represented by colors.



(a)



(b)

Fig. 11. Visualization of localization results for a threshold value of $\tau = 0.9$. (a) Scatter plot of localized molecules; (b) high-resolution image generated from localization results.

Fig. 11 reveals the tubular structure of the microtubules which varies depending on the depth.

5. Conclusion

In this paper, we propose a novel deep learning model for regressing the coordinates of molecules from an observed image obtained by MUM. The key principle of our method is to decompose a regression problem into classification and regression problems for each voxel in an observed image. Then, the localization problem of each voxel is solved as a combination of a binary classification and regression problem for a single molecule. By this decomposition, a fully convolutional neural network can be used to solve the regression problem, and training the network is also simplified. In addition, our proposed method is parameter-free, computationally efficient, and can process images of any size by the same model. The experimental results demonstrate the effectiveness of our proposed method.

In this study, we applied our method to images obtained by MUM. Then, we used the voxels in the input image to decompose the regression problem. However, the layout of the voxels can be modified. Therefore, our method is applicable to a wide variety of optical layouts if a reasonable generative model is provided and if it can be assumed that at most one molecule exists in a voxel. For example, 3D localization from a single image can also be performed by changing the optical layout of the microscope and the layout of the output voxels from the layout of the input image. We plan to study the application to other types of microscopy in future work. The overlapped layout of voxels may also improve the classification accuracy of the voxels.

CRediT authorship contribution statement

Toshimitsu Aritake: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - original draft, Visualization. **Hideitsu Hino:** Formal analysis, Conceptualization. **Shigeyuki Namiki:** Investigation. **Daisuke Asanuma:** Investigation. **Kenzo Hirose:** Supervision. **Noboru Murata:** Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partly supported by JSPS KAKENHI Grant Nos. 17H01793, 18H03291, and JST CREST Grant No. JPMJCR1761, JPMJCR14D7.

References

Aritake, T., Hino, H., Namiki, S., Asanuma, D., Hirose, K., Murata, N., 2020. Fast and robust multiplane single molecule localization microscopy using deep neural network. arXiv:2001.01893.

Babcock, H.P., Zhuang, X., 2017. Analyzing single molecule localization microscopy data using cubic splines. Scientific Rep. 7 (1), 552. https://doi.org/10.1038/s41598-017-00622-w. URL:/https://pubmed.ncbi.nlm.nih.gov/28373678 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5428856/.

Betzig, E., Patterson, G.H., Sougrat, R., Lindwasser, O.W., Olenych, S., Bonifacino, J.S., Davidson, M.W., Lippincott-Schwartz, J., Hess, H.F., 2006. Imaging intracellular fluorescent proteins at nanometer resolution 313(5793), 1642–1645.https://doi.org/10.1126/science.1127344.

Boyd, N., Jonas, E., Babcock, H., Recht, B., 2018. DeepLoco: Fast 3D localization microscopy using neural networks. bioRxiv.https://doi.org/10.1101/267096.

Cleveland, D.W., Sullivan, K.F., 1985. Molecular biology and genetics of tubulin. Annu. Rev. Biochem. 54(1), 331–366. pMID: 3896122.https://doi.org/10.1146/annurev.bi.54.070185.001555.

Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A., 2012. A kernel two-sample test. J. Mach. Learn. Res. 13 (null), 723–773.

Gu, L., Sheng, Y., Chen, Y., Chang, H., Zhang, Y., Lv, P., Ji, W., Xu, T., 2014. High-density 3D single molecular analysis based on compressed sensing. Biophys. J. 106 (11), 2443–2449. https://doi.org/10.1016/j.bpj.2014.04.021.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778.https://doi.org/10.1109/CVPR.2016.90.

Hershko, E., Weiss, L.E., Michaeli, T., Shechtman, Y., 2019. Multicolor localization microscopy and point-spread-function engineering by deep learning. Opt. Express 27 (5), 6158–6183.

Hess, Samuel T., Girirajan, Thanu P.K., Mason, Michael D., 2006. Ultra-high resolution imaging by fluorescence photoactivation localization microscopy. Biophys. J. 91 (11), 4258–4272. https://doi.org/10.1529/biophysj.106.091116.

Ioffe, S., Szegedy, C., 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning – Volume 37, ICML'15, JMLR.org. pp. 448–456.

Kingma, D.P., Ba, J., 2015. Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings.

Li, Y., Mund, M., Hoess, P., Matti, U., Nijmeijer, B., Sabinina, V.J., Ellenberg, J., Schoen, I., Ries, J., 2017. Fast, robust and precise 3d localization for arbitrary point spread functions.https://doi.org/10.1101/172643. URL:/https://www.biorxiv.org/content/early/2017/08/10/172643.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. SSD: Single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), Computer Vision – ECCV 2016. Springer International Publishing, Cham, pp. 21–37.

Lucas, A., Iliadis, M., Molina, R., Katsaggelos, A.K., 2018. Using deep neural networks for inverse problems in imaging: Beyond analytical methods. IEEE Signal Process. Mag. 35 (1), 20–36. https://doi.org/10.1109/MSP.2017.2760358.

Möckl, L., Roy, A.R., Petrov, P.N., Moerner, W.E., 2020. Accurate and rapid background estimation in single-molecule localization microscopy using the deep neural network bgnet 117(1), 60–67.https://doi.org/10.1073/pnas.1916219117.

Nehme, E., Weiss, L.E., Michaeli, T., Shechtman, Y., 2018. Deep-storm: super-resolution single-molecule microscopy by deep learning. Optica 5 (4), 458–464. https://doi.org/10.1364/OPTICA.5.000458.

Nehme, E., Freedman, D., Gordon, R., Ferdman, B., Weiss, L.E., Alalouf, O., Orange, R., Michaeli, T., Shechtman, Y. DeepSTORM3D: dense three dimensional localization microscopy and point spread function design by deep learning (jun). arXiv:1906.09957.

Ouyang, W., Aristov, A., Lelek, M., Hao, X., Zimmer, C., 2018. Deep learning massively accelerates super-resolution localization microscopy. Nat. Biotechnol. 36 (5), 460–468. https://doi.org/10.1038/nbt.4106.

Schermelleh, L., Ferrand, A., Huser, T., Eggeling, C., Sauer, M., Biehlmaier, O., Drummen, G.P., 2019. Super-resolution microscopy demystified.https://doi.org/10.1038/s41556-018-0251-8.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations.

Zelger, P., Kaser, K., Rossboth, B., Velas, L., Schütz, G.J., Jesacher, A., 2018. Three-dimensional localization microscopy using deep learning. Opt. Express 26 (25), 33166–33179. https://doi.org/10.1364/OE.26.033166.

Zhang, P., Liu, S., Chaurasia, A., Ma, D., Mlodzianoski, M.J., Culurciello, E., Huang, F. Analyzing complex single-molecule emission patterns with deep learning. Nat. Methods 11, 913–916.https://doi.org/10.1038/s41592-018-0153-5.