Toolboxes User's Guide for "Systematic Characterization of Optical Aberrations Reveals Cryo-FLM Localization Fidelity"

Hongjia Li[1,2], Lauren Ann Metskas[2, 3, 4*], Fang Huang[1, 4, 5*]

Address:

[1] Weldon School of Biomedical Engineering, Purdue University, West Lafayette, IN, USA

[2] Biological Sciences, Purdue University, West Lafayette, IN, USA

[3] James Tarpo Jr. and Margaret Tarpo Department of Chemistry, Purdue University, West Lafayette, IN, USA

[4] Purdue Institute of Cancer Research, Purdue University, West Lafayette, IN, USA

[5] Purdue Institute for Integrative Neuroscience, Purdue University, West Lafayette, IN, USA

*Corresponding authors. Email: metskas@purdue.edu, huang892@purdue.edu

Outline

1. General Information

We provide **three MATLAB toolboxes** as supplementary software for this manuscript:

1. **MLE-based Phase Retrieval Algorithm** – For retrieving point spread function (PSF) models from bead image stacks.
2. **Data Simulation for Biplane Setup** – For generating simulated biplane datasets.
3. **3D Localization for Biplane Setup** – For high-precision single-molecule localization in biplane microscopy.

1.1 Installation Requirements

- **Operating System:** Windows 7 or later (64-bit).
- **MATLAB:** R2023b (64-bit) or newer. Download from MathWorks.
- **GPU & Drivers:** CUDA 12-compatible graphics driver. Download from NVIDIA CUDA 12.0 Archive.
- **DIPimage Toolbox:** Version 3.5.2. Download from MATLAB File Exchange.

1.2 Toolbox Updates

All three toolboxes are available as Supplementary Software with the manuscript. Future updates and maintenance will be freely available at: https://github.com/HuanglabPurdue/

2. MLE-Based Phase Retrieval Toolbox

2.1 Overview

The MLE-based phase retrieval toolbox reconstructs the PSF model from bead image stacks acquired at multiple axial positions (e.g., –1 μm to +1 μm, with a 200 nm step size).

Key Features:

- Based on **scalar diffraction theory**, where the PSF is computed as the **Fourier transform of the pupil function**.
- The pupil function is parameterized using **Zernike polynomials**.
- Includes example bead stacks acquired under **room-temperature (RT)** and **cryogenic (CryoT)** conditions for testing.

Figure 1. Beads stack example

2.2 Step-by-Step Instructions

Step 1. Install the Toolbox

- Unzip the file **1. Phase Retrieval.zip**.
- It contains three main folders:
    - 1. Crop Single Beads
    - 2. Phase Retrieval
    - 3. Test Data

Step 2. Crop Single Beads

Navigate to the *1. Crop Single Beads* folder and run:

- Crop_single_beads_1p.m – Extracts bead subregions across Z-steps.
- Crop_single_noise_1p.m – Extracts background subregions for noise estimation.

Inputs:

- 3D image stack of beads at multiple axial positions from *3. Test Data* Folder.

Outputs:

- subregion_ch1_s and subregion_ch1_n – Subregions of beads and noise, respectively.

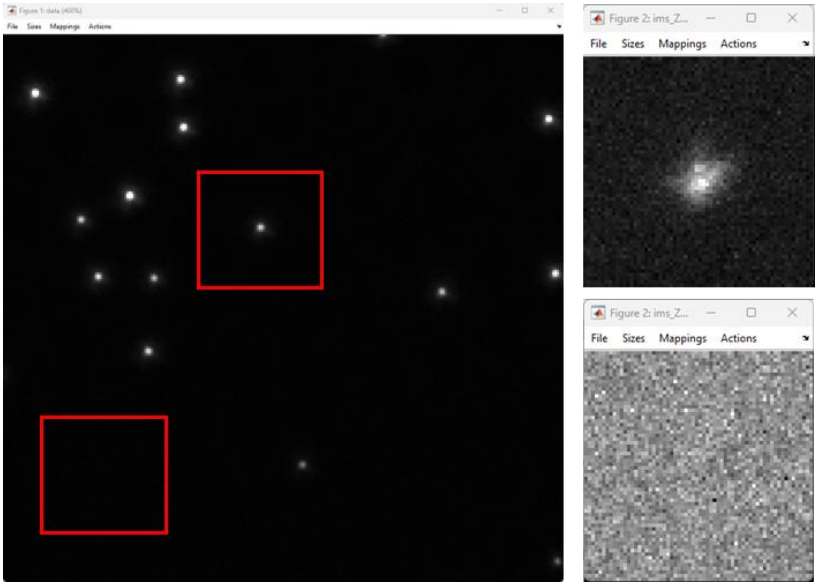

Figure 2. Example of select bead subregion and noise subregion

Step 3. Phase Retrieval

- Navigate to *2. Phase Retrieval*.
- Check and update support_path in main.m.
- Run main.m.

```
%% ------------------------------------------------------------------------
% Add Required Paths
% ------------------------------------------------------------------------
gitpath = '.\Support\Github';
addpath(fullfile(gitpath, 'Live3DSMSN\SRsCMOS'));
addpath(fullfile(gitpath, 'CUDAmex'));
addpath(fullfile(gitpath, 'mex'));
addpath(fullfile(gitpath, 'helpers'));
addpath('.\Support\tag\RC1\PSF Toolbox');
```

Required Parameters

Camera and System Parameters:

- **CCD Offset:** Baseline intensity of the camera (in counts).
- **Gain:** Conversion factor between electrons and digital counts (e^-/ADU).
- **NA:** Numerical aperture of the objective lens.
- **λ (Lambda):** Emission wavelength of the fluorophore (in nm).
- **Refractive Index:** Immersion medium index (e.g., 1.00 for nitrogen vapor).
- **Pixel Size:** Pixel size in the object plane (in nm).

```
% CCD camera properties
probj.CCDoffset = 100;
probj.Gain = 7;

% Optical system configuration
probj.PRstruct.NA = 0.9;
probj.PRstruct.Lambda = 0.515;  % Wavelength in µm
probj.PRstruct.RefractiveIndex = 1;
probj.Pixelsize = 0.130;        % µm/pixel
```

Phase Retrieval and PSF Settings:

- **PSFsize:** Image size for PSF modeling (default: 256 × 256).
- **Subroisize:** Crop size around the bead center.
- **ZernikeorderN:** Zernike polynomial order (default: 7).
- **PSF_size_forMLE:** PSF size for MLE fitting (e.g., 16 × 16 for noisy datasets).
- **Z_pos_plane1:** Z positions for input stacks (adjust based on dataset).
- **Automatic Data Loading:** Bead and noise data are auto-loaded.

```
% Phase retrieval and PSF settings
probj.PSFsize = 256;
probj.SubroiSize = 40;
probj.ZernikeorderN = 7;
probj.PSF_size_forMLE = 16;
probj.resizeFactor = 1;
```

```
%% ------------------------------------------------------------------------
% Set Z positions (in microns) for the focal stack
% ------------------------------------------------------------------------
Zpos_plane1 = -1.4:0.2:1.4; %for cryoT
% Zpos_plane1 = -1:0.2:1; %for rt
probj.Zpos = Zpos_plane1;
probj.Zindstart = 1;
probj.Zindend = numel(Zpos_plane1);
probj.Zindstep = 1;
```

```
%% ------------------------------------------------------------------------
% Load Bead Data and Noise Data
% ------------------------------------------------------------------------
% Replace ims_Ztrue_plane1_s and ims_Ztrue_plane1_n with your actual variables
input_beadData = ims_Ztrue_plane1_s;   % Signal
input_noiseData = ims_Ztrue_plane1_n;  % Background

probj.BeadData = input_beadData;
probj.NoiseData = input_noiseData;
```

Intensity and Background Coefficients:

Two additional coefficients (intensity and background) are initialized to **0.2** and **0.1**, respectively, and automatically scaled during retrieval.

```
%% ------------------------------------------------------------------------
% Set Initial Phase Coefficients
% ------------------------------------------------------------------------
num_terms=(probj.ZernikeorderN + 1) * (probj.ZernikeorderN + 2) / 2+2; %aberrations and Intensity&bg
probj.phase_coefficients = zeros(1, num_terms);
probj.phase_coefficients(end-1) = 0.2;
probj.phase_coefficients(end) = 0.1;
```

Gaussian Blur Kernel Size:

Automatic kernel estimation works well for RT datasets but may over-blur cryo-FLM data. We recommend **manual values of 1.0 or 1.1** for cryo-FLM datasets.

```
% Estimate PSF width (sigma) using the updated phase model
% probj.precalsigma_new();
% disp(['Calculated Gaussian blur kernel size: ', num2str(probj.gBlur), ']']);

% Set Gaussian blur parameter manually
probj.gBlur = 1; %for cryoT beads, use 1.1 for more diffused pattern
```

Iterations and Output Directory:

- **Max_iterations:** Multiple iterations can be run due to noise and aberrations.
- NCC-based metrics help in selecting the best result.

Output:

- **Figure:** Comparison of measured vs. fitted PSFs.
- **MAT File:** Saved as PR-PSF_iteration_#.mat with PSF, pupil function, and Zernike coefficients.

```
%% ------------------------------------------------------------------------
% Output Directory Configuration
% ------------------------------------------------------------------------
save_dir = 'F:\cryoFLM\codes\MLE_publish_version\1. Phase Retrieval\3. Test Data\Cryo Temperature Bead Stack\';

if ~exist(save_dir, 'dir')
    mkdir(save_dir);
    disp('Folder created successfully!');
else
    disp('Folder already exists.');
end
```

```
%% ------------------------------------------------------------------------
% Phase Retrieval Loop (Iterative Optimization)
% ------------------------------------------------------------------------
max_iterations = 1;

for iteration = 1:max_iterations
    disp(['Iteration ', num2str(iteration), ' of ', num2str(max_iterations)]);
```

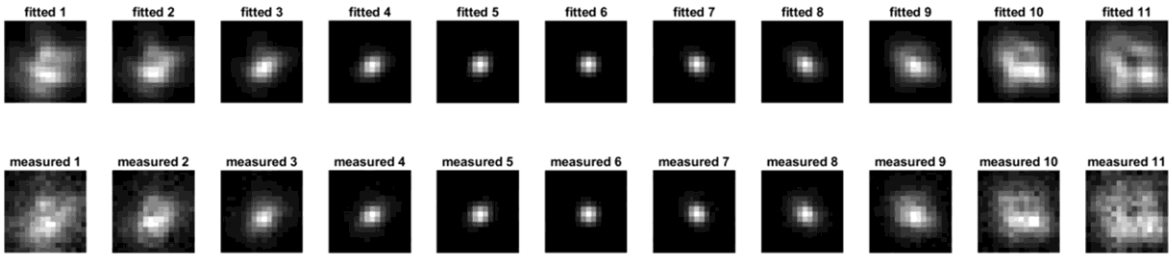

Figure 3. Example of fitted (retrieved) PSF and measured PSF

Step 4. Visualization of Results

To visualize the phase retrieval outcomes:

1. Navigate to the *2. Phase Retrieval* folder.
2. Run the visualize_all_results.m script.

Notes:

- All preset parameters (e.g., NA, λ (Lambda), RefractiveIndex, PixelSize, save_dir) remain consistent with those defined in the phase retrieval step.
- The script generates multiple outputs, including:
    - **Measured vs. Fitted PSFs** for qualitative comparison.
    - **Optimization Metrics** such as Log-likelihood, Mean Squared Error (MSE), and Normalized Cross-Correlation (NCC).
    - **Pupil Phase** visualization to assess aberrations.

```
%% Add necessary paths
gitpath = '.\Support\Github';
addpath(fullfile(gitpath, 'Live3DSMSN', 'SRsCMOS'));
addpath(fullfile(gitpath, 'CUDAmex'));
addpath(fullfile(gitpath, 'mex'));
addpath(fullfile(gitpath, 'helpers'));

addpath('.\Support\tag\RC1\PSF Toolbox');
```

```
%% Initialize PRPSF object
disp('Visualize the pupil and metrics from phase retrieval...');
probj = PRPSF();

% Define imaging system and acquisition parameters
probj.PRstruct.NA           = 0.9;
probj.PRstruct.Lambda       = 0.575;
probj.PRstruct.RefractiveIndex = 1.0;
probj.Pixelsize             = 0.130;   % in microns

% Precompute FFTs and constants needed for optimization
probj.precomputeParam();
```

```
%% Phase retrieval iterative loop
save_dir = '..\3. Test Data\Cryo Temperature Bead Stack\';

num_iterations=1;
for iteration = 1:num_iterations
    disp(['Iteration ', num2str(iteration), ' of ', num2str(num_iterations)]);
```

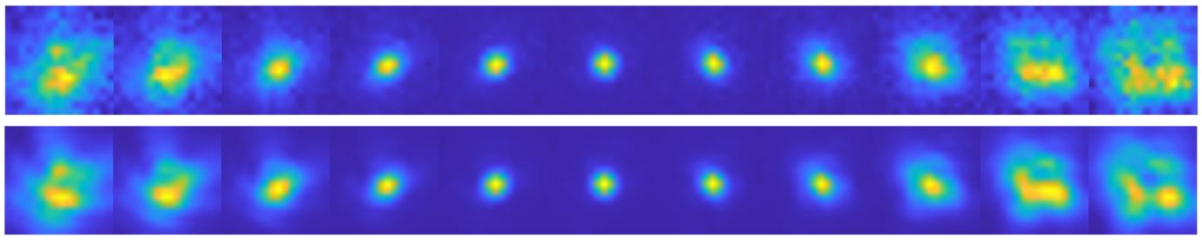

Figure 4. Example of measured PSF and fitted (retrieved) PSF

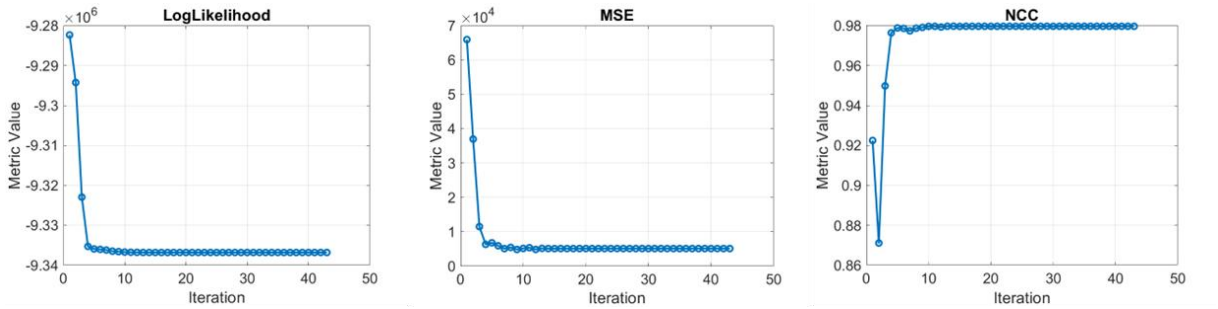

Figure 5. Example of metrics during the MLE optimization

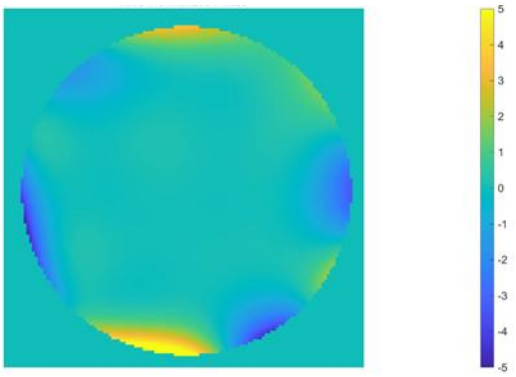

Figure 6. Example of the retrieved pupil phase

3. Data Simulation Toolbox

3.1 Overview

Generates biplane datasets with defined aberrations for evaluating localization accuracy. This toolbox is adapted and refined from our previous work[1].

Key Features:

- Simulates datasets with optical aberrations.
- Varying photon counts for performance benchmarking.
- Multiple PSFs at identical axial positions for statistical averaging.

3.2 Install

- Unzip the file **2. Data Simulation Biplane.zip**.
- Navigate to *2. Data Simulation Biplane* folder.
- Run gen_simuPSFs_biplane_cryofm_multiphotons.m.

3.3 Parameters

Basic PSF Parameters:

- **Imsz:** Image size per PSF (pixels).
- **Numimage:** Number of Z-slices.
- **Frame_per_z:** Frames per Z position.
- **Photon_numbers:** List of photon counts (one .mat per photon value).
- **Bg_num:** Background photon count.
- **Dist_Bi:** Biplane channel separation (μm).
- **Startz / Step_size:** Axial range and step size (μm).
- **Pixel_size:** Pixel size in object plane (μm).

```
%% === Initial Parameters ===
imsz = 32;                          % Image patch size (pixels)
Numimage = 15;                      % Number of Z-slices
frames_per_z = 100;                 % Frames per Z position
photon_numbers = [2000, 5000, 10000, 20000, 50000, 100000, 200000, 400000]; % Photon counts
bg_num = 400;                       % Background photons (total)
dist_Bi = 1;                        % Biplane separation (μm)
start_z = -0.7;                     % Starting axial position (μm)
step_size = 0.1;                    % Z-step size (μm)
pixel_size = 0.13;                  % Pixel size (μm)
```

Additional Parameters:

- **Aberrs:** Path to phase-retrieved .mat file.

- **Resultfolder:** Output directory (separate subfolder per photon count).

```
%% === Load Aberration Model ===
% Choose the aberration model from a phase retrieval output.
aberrs = load('../1. Phase Retrieval/3. Test Data/1. Bead Stack RT/PR-PSF_iteration_1.mat');

%% === Main Loop Over Photon Numbers ===
for photon = photon_numbers

    %% Create Output Folder
    resultfolder = sprintf("./test_data_simulation/biplanedis%.2f_bg_%d_intensity_%d", ...
        dist_Bi, bg_num, photon);
    if ~exist(resultfolder, 'dir')
```

Outputs:

- .mat file containing biplane stacks:
  - subregion_ch1 (e.g., 32 × 32 × 1500).
  - subregion_ch2 (e.g., 32 × 32 × 1500).

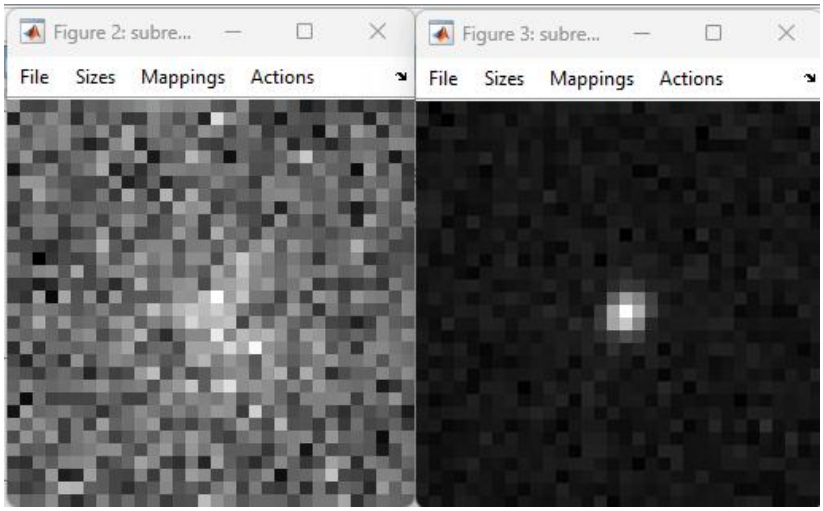

Figure 7. Example of simulated biplane datasets: subregion_ch1 (left), subregion_ch2 (right)

4. 3D Localization Toolbox

4.1 Overview

Performs GPU-accelerated 3D localization on biplane datasets using PSF models. This toolbox is adapted and refined from our previous work[2].

4.2 Install

- Unzip the file **3. 3D Localization Biplane.zip**.
- Navigate to *3. 3D Localization Biplane* folder.
- Run main.m.

4.3 Parameters

Input and Output:

- **Data_file:** Biplane dataset in .mat format.
- **Aberrations_file:** Phase-retrieved PSF .mat file.
- **Bead_idx / Model_idx:** Used for naming outputs (optional).

Initial Guess:

- **I_init, Bg_init:** Initial guesses for PSF intensity and background.
- **X_guess, Y_guess:** Initial lateral positions.
- **Z_values:** Initial axial positions.
- **Frames_per_z:** Number of frames per axial step.

```
% --- Input / Output ---
data_file = ['..\2. Data Simulation Biplane\test_data_simulation\' ...
    'biplanedis1.00_bg_400_intensity_10000\subregions_beads1.mat'];
aberration_file = ['../1. Phase Retrieval/3. Test Data/1. Bead Stack RT/'
    'PR-PSF_iteration_1.mat'];
save_root = fullfile('.', 'loc_results\');

% --- Bead & Model ---
bead_idx = 1;               % Bead index
model_idx = 1;              % Model index

% --- Initial Values ---
I_init = 10000 / 2;        % Initial intensity guess per channel
bg_init = 400 / 2;          % Initial background guess per channel

% --- XYZ Guessing ---
x_guess = 16.5;             % Initial x-position (for imsz=32)
y_guess = 16.5;             % Initial y-position (for imsz=32)
z_values = -0.7:0.1:0.7;    % Z positions for initial guess
frames_per_z = 100;         % Frames per Z position
```

PSF Parameters:

- **Lambda_override:** Emission wavelength (μm).
- **Psf_size:** PSF image size (pixels).
- **Dist_biplane:** Biplane distance (μm).
- **Bin:** Upsampling factor for PSF models.
- **Nzs:** Number of Z-frames (default: [-1.3 μm, 1.3 μm]).

```
% --- PSF Parameters ---
lambda_override = 0.515;    % Wavelength override (μm)
psf_size = 128;             % PSF grid size
dist_biplane = 1;           % Biplane separation (μm)
bin = 4;                    % Binning factor
Nzs = 601;                  % Number of Z-slices

% --- CUDA Localization ---
iterateN = 400;             % Number of iterations
lambda_reg = 0;             % Regularization parameter
```

Outputs:

- **Figure:** Z-localization plot (frame-by-frame).
- **MAT File:** Localization results (x, y, z).

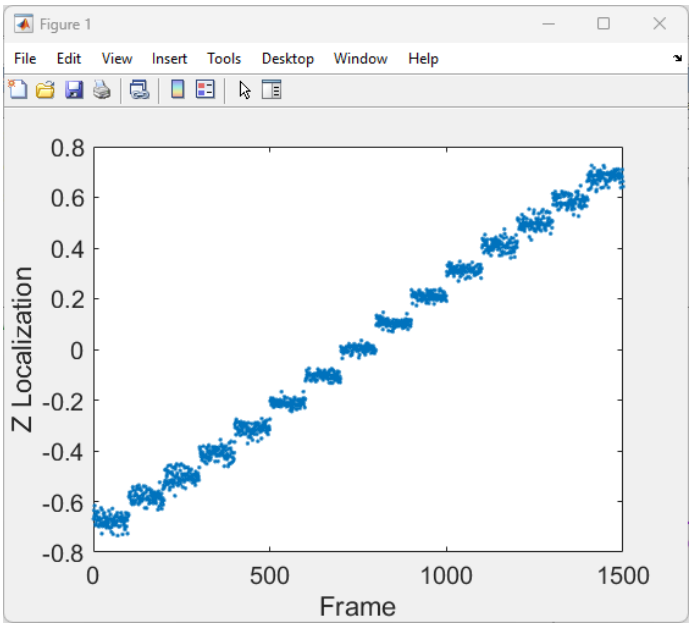

Figure 8. Example of z localization for each frame.

5. Datasets and source codes

5.1 Demonstration Datasets

We provide example datasets for testing and demonstration:

- 1. Phase Retrieval\3. Test Data\1. Bead Stack RT\Beads_stack.mat Bead image stack collected under room-temperature conditions.
- 1. Phase Retrieval\3. Test Data\2. Bead Stack Cryo\Beads_stack.mat Bead image stack collected under cryogenic-temperature conditions.

5.2 Phase Retrieval Source Codes

Folder: 1. Crop Single Beads

- Crop_single_beads_1p.m – Crops subregions (64 × 64 × n) containing a single bead stack for phase retrieval.
- Crop_single_noise_1p.m – Crops subregions (64 × 64 × n) containing only background regions for noise estimation.

Folder: 2. Phase Retrieval

- main.m – Main script for running MLE-based phase retrieval.
- visualize_all_results.m – Visualizes the retrieved results, including measured vs. fitted PSFs, pupil phase, and evaluation metrics.
- calc_nLogLikelihood.m – Calculates the negative log-likelihood used in phase retrieval optimization.
- Support Folder – Contains all auxiliary functions, including PSF generation, pupil modeling, and other helper utilities.

5.3 Data Simulation Source Codes

Folder: 2. Data Simulation Biplane

- main.m – Generates biplane simulated datasets with specified photon numbers and aberrations.

5.4 3D Localization Source Codes

Folder: 3. 3D Localization Biplane

- main.m – Performs GPU-based 3D localization for simulated PSFs.
- genIniguess.m – Estimates initial lateral positions.
- geniniBiplane_z_mat_parfor.m – Estimates initial axial positions.

- genpsf_biplane_real.m – Generates PSF models directly from pupil functions.
- gensamplepsf_biplane.m – Pre-generates channel-specific PSF models.
- cc2.m – Calculates 2D cross-correlation.
- catstruct.m – Concatenates results from the two biplane channels.
- genpsfstruct.m – Computes first-order partial derivatives of a 3D PSF for MLE-based optimization.

Reference

1. Zhang P, Liu S, Chaurasia A, et al. Analyzing complex single-molecule emission patterns with deep learning[J]. Nature methods, 2018, 15(11): 913-916.
2. Xu F, Ma D, MacPherson K P, et al. Three-dimensional nanoscopy of whole cells and tissues with in situ point spread function retrieval[J]. Nature methods, 2020, 17(5): 531-540.