Rapid, accurate particle tracking by calculation of radial symmetry centers

Raghuveer Parthasarathy
Department of Physics, Materials Science Institute, and Institute for Molecular Biology
The University of Oregon, Eugene, Oregon, USA

Supplementary Software

Code for

- generating simulated images,
- determining particle center locations via various methods including the radial-symmetry-based method introduced in this manuscript, and
- calculating and plotting localization accuracy.

Disclaimer / License

Language requirements. All functions are written in MATLAB and have been tested using version 2011a. The Statistics Toolbox is required for generation of Poisson-distributed noise (in `modelimage.m`), and the Optimization Toolbox is required for numerical fitting to Gaussian functions using nonlinear least squares and maximum likelihood estimation.

Functions

Radial-symmetry-based particle localization algorithm

- **`radialcenter.m`** – radial symmetry based particle localization. This function implements the particle tracking method introduced in this paper

Other particle localization algorithms

- `gaussfit2Dnonlin.m` – nonlinear least squares Gaussian fitting
- `gaussfit2DMLE.m` – maximum likelihood estimation of a Gaussian form
- `fluorobancroft.m` – particle localization using the FluoroBancroft algorithm for a symmetric Gaussian form, as described in Andersson, S. *Opt. Express* **16**, 18714-18724 (2008).

Assessment and plotting functions

- `modelimage.m` – generates model CCD images of single particles, pixelated and with Poisson-distributed noise
- `tracking_tests_RP_Apr2012.m` – assesses the accuracy of various algorithms' localization of model images
- `make_singleSNr_plots.m` – creates plots, like Figure 2a, from the output of `tracking_tests_RP_Feb2012.m` calculated at a single SNr
- `make_multiSNr_plots.m` – creates plots, like Figure 2a, from the output of `tracking_tests_RP_Feb2012.m` calculated at a range of SNr values
- `interleaveplot.m` – called by `make_singleSNr_plots.m`; creates plots with interleaved values.
- `usual_labels_for_tracking_tests.m` – called by `make_singleSNr_plots.m`; sets font sizes, etc.

Other functions

- `fitline.m` – simple line fit
- `psf2d.m` – theoretical 2D point spread function

Functions **not** included in this collection (not written by Raghuveer Parthasarathy)

- `errorbarlogx.m` – properly plots error bars on logarithmic graphs. (F. Moisy, 2006, availalable on the Mathworks FileExchange: http://www.mathworks.com/matlabcentral/fileexchange/9715-errorbarlogx-m)
- `gauss2dcirc.m` – Weighted linearized Gaussian fitting, from the authors of Anthony, S. M. & Granick, S. *Langmuir* **25**, 8152-60 (2009), available at http://groups.mrl.uiuc.edu/granick/software.html

Procedure (examples)

Please see the comments in the code for descriptions of inputs, outputs, and parameters. This document is **not** intended to serve as a detailed set of instructions, but rather as a quick set of examples to examine.

To run various localization algorithms on images at a single signal-to-noise ratio, one can use `tracking_tests_RP_Apr2012.m` as follows:

```
[sigma time bias sigbias toterror] = tracking_tests_RP_Apr2012(1000, 0.5,20, 'single_SNr_output.mat');
```

In this example, 1000 images with SNr=20 are created and localized. Graphs of the output can be made, for example, using: `make_singleSNr_plots('single_SNr_output.mat', 'SNrplots_');`
The "`'SNrplots_`toterror.png" output graph should look similar to Figure 2a.

To run various localization algorithms on images at a range of signal-to-noise ratios, one can use `tracking_tests_RP_FApr012.m` as follows:

```
[sigma time bias sigbias toterror] = tracking_tests_RP_Feb2012(1000, 0.5,[9 120 15], 'multi_SNr_output.mat');
```

In this example, 1000 images each with SNr from 9 to 120 are created and localized. Graphs of the output can be made, for example, using:

```
make_multiSNr_plots('multi_SNr_output.mat', [], 1:5, 'MultiSNrplots_');
```

The "...toterror_x.png" output graph should look similar to Figure 2b.