This code is distributed as accompanying software for the article C.S. Smith, N. Joseph, B. Rieger, K.A. Lidke, *Fast, single-molecule localization that achieves theoretically minimum uncertainty*, Nature Methods, 2010, 7(5):373-375

1 Overview

This code requires the NVIDA CUDA Driver and Toolkit to be installed. The code has been compiled for various OS and CUDA versions. Calling syntax is identical for all cases. See the companion note in the 'ExampleCode.pdf' for Windows Vista about allowing longer run times on the GPU without a timeout error.

The method of fitting is given by the $fittype$ variable.

$fittype = 1$: Fits $(x, y, Photons, Bg)$ given PSF_σ.

$fittype = 2$: Fits $(x, y, Photons, Bg, PSF_\sigma)$ and uses PSF_σ as a starting guess.

$fittype = 3$: Fits $(x, y, Photons, Bg, z)$ using the model given below and PSF_σ is taken as σ_0.

$fittype = 4$: Fits $(x, y, Photons, Bg, PSF_{\sigma_x}, PSF_{\sigma_y})$ using PSF_σ as a starting guess.

The calling syntax from Matlab is:

$$[P\ \ CRLB\ \ LL] = \text{gaussmlev2}(data, PSF_\sigma, iterations, fittype)$$

or

$$[P\ \ CRLB\ \ LL] = \text{gaussmlev2}(data, PSFSigma, iterations, fittype, A_x, A_y, B_x, B_y, \gamma, d)$$

where the inputs $A_x, A_y, B_x, B_y, \gamma, d$ are required and used only for z-fitting. $data$ must be data type of 'single' and all other inputs can be scalars of any data type. See 'help gaussmlev2' for more information.

2 Changes from GPUgaussMLE (version 1)

2.1 Output Variables

The output structure is now $[P\ \ CRLB\ \ LL]$. P is the $N \times M$ matrix of found parameters where N is the number of fits and M is the number of fitted variables (i.e. 4 for $fittype = 1$). $CRLB$ is the equivalently sized matrix of Cramer-Rao Lower Bound calculated variances for each parameter. LL is the log-likelihood calculated with the found parameters and uses the Stirling approximation. LL is mathematically equivalent to the log-likelihood ratio and therefore $-2LL$ is approximately Chi-Square distributed with $(k - M)$ degrees of freedom, where k is the number of pixels in the fitting sub-region.

2.2 Functionality

The CRLB is now calculated internally using an LU decomposition method for inverting the Fisher information matrix and now also returns CRLB values for $fittype = 4$. The center of mass estimation is now used for all models as a starting guess for the x, y positions. This should result in fewer iterations to achieve convergence. The $CRLB$ outputs are now the variances, not standard deviations.
z-fitting is now performed using the expressions:

$$\sigma_x(z) = \sigma_0\sqrt{1 + \frac{(z-\gamma)^2}{d^2} + A_x\frac{(z-\gamma)^3}{d^3} + B_x\frac{(z-\gamma)^4}{d^4}}\,, \tag{17a}$$

$$\sigma_y(z) = \sigma_0\sqrt{1 + \frac{(z+\gamma)^2}{d^2} + A_y\frac{(z+\gamma)^3}{d^3} + B_y\frac{(z+\gamma)^4}{d^4}}\,. \tag{17b}$$

to match expressions used elsewhere.

3 Erratum

In the original supplemental material, the expressions below had a typo in the text and were missing the square root in the denominators. Expressions below are corrected.

$$\Delta \mathrm{E}_x(x,y) \equiv \frac{1}{2}\mathrm{erf}\left(\frac{x-\theta_x+\frac{1}{2}}{\sqrt{2\sigma^2}}\right) - \frac{1}{2}\mathrm{erf}\left(\frac{x-\theta_x-\frac{1}{2}}{\sqrt{2\sigma^2}}\right), \tag{4a}$$

$$\Delta \mathrm{E}_y(x,y) \equiv \frac{1}{2}\mathrm{erf}\left(\frac{y-\theta_y+\frac{1}{2}}{\sqrt{2\sigma^2}}\right) - \frac{1}{2}\mathrm{erf}\left(\frac{y-\theta_y-\frac{1}{2}}{\sqrt{2\sigma^2}}\right), \tag{4b}$$

In the original supplemental material, the expressions below had a typo in the text, giving an extra factor of -$\frac{1}{2}$. Expressions below are corrected.

$$\frac{\partial \mu_k(x,y)}{\partial \theta_x} = -\frac{\theta_{I_0}}{\sigma^2}\int_{A_k}(\theta_x-u)\mathrm{PSF}(u,v)\mathrm{d}u\mathrm{d}v \tag{10a}$$

$$\frac{\partial \mu_k(x,y)}{\partial \theta_y} = -\frac{\theta_{I_0}}{\sigma^2}\int_{A_k}(\theta_y-v)\mathrm{PSF}(u,v)\mathrm{d}u\mathrm{d}v \tag{10b}$$