Easy Pupil Finder Estimation Code 1.02 – README

OVERVIEW

This package includes a collection of MATLAB files which are designed to:

1. Given a calibration scan of the image of a point emitter with an engineered point spread function (PSF),
2. Perform a phase retrieval algorithm based on maximum likelihood estimation (MLE) of a phase aberration term which is added to the theoretical pupil function of the imaging system.
3. Use the phase-retrieved pupil function to perform single-emitter localization.

Files included:

- ESTIMATE_PUPILFUNC.m
- LOCALIZE_PSF_DATA.m
- localize_emitter_mleInterp.m
- plot_field.m
- calc_img_fromCoeffVec.m
- calc_nLogLikelihood.m
- calc_meanDarkCounts.m
- choose_roi.m
- find_dicMatch.m
- apply_locBias.m
- dftregistration.m
    - This function is available with permission from Dr. Manuel Guizar-Sicairos. The citation for this algorithm is:
      Manuel Guizar-Sicairos, Samuel T. Thurman, and James R. Fienup, "Efficient subpixel image registration algorithms," Opt. Lett. 33, 156-158 (2008).
- zernike_rad_per_coeff_14orders.mat
- phasemap.mat

This document explains the usage of the two main programs, ESTIMATE_PUPILFUNC (which performs #2 above) and LOCALIZE_PSF_DATA (which performs #3 above).

Please direct correspondence regarding this package to Petar Petrov (**petar@stanford.edu**).

1. DETERMINING A PUPIL FUNCTION

In order to determine a pupil function, the user runs a single MATLAB function, ESTIMATE_PUPILFUNC.m. This function goes through several prompts, explained below, and ultimately produces and saves an estimate of the pupil function. The use of the resulting pupil function to perform 3D localization microscopy is described on page 5.

Files needed:

- **PSF scan data in units of camera ADC counts (.TIF)**

  - This data can be the entire field of view. There is a cropping step within the code (Section 4 below), in which a single emitter will be chosen.
- **Dark current data in units of camera ADC counts (.TIF) – optional**
  - This data must be the same size as the PSF data in x-y, but can be a different number of frames. Within the code, the average value across all frames is taken for each pixel, and this number is subtracted from all frames of the PSF data file.
  - Pressing 'Cancel' when prompted for the location of this file will skip ahead and no dark current subtraction will be applied.
- **Phase mask pattern in units of meters (.MAT)**
  - This should be a square array, with each pixel representing a local thickness of the phase mask medium (for transmissive masks) or the effective optical path delay of a spatial light modulator or deformable mirror (for reflective masks). Recommended array size ~500x500. Phase masks are often circular, in which case it is recommended to make the diameter of the circle smaller than the side length of the array (e.g. 400 pixels).
  - The scale of this array must be known (i.e. pixels per meter).
  - The name of the array must be *physMask*
  - If the user desires to use the tetrapod phase mask, this mask is available for non-commercial, research use only from wmoerner@stanford.edu .

Anatomy of **ESTIMATE_PUPILFUNC.m**:

1. **GET FILES AND LOCATIONS FROM USER FOR THE CALIBRATION SCAN**
   In this section, the user identifies the location of necessary files and chooses a directory in which to save results.
   - When prompted, locate the three files listed above.
   - If dark current file is unavailable, press 'Cancel' when prompted for its location.
   - The directory identified for saving results (the fourth prompt) will have several FIG files and a MAT file saved to it.
2. **GET MASK PARAMETERS**
   In this section, the user identifies the type of phase mask and its parameters.
   - Transmissive mask (assumed to be placed in air)
     - Specify scale of MAT file (pixels per meter)
     - Specify refractive index of mask
   - Reflective mask (assumed to be placed in air)
     - Specify scale of MAT file (pixels per meter)
     - "Effective" refractive index is set automatically to -1 in accordance with the helper function *calc_phase_fromPhysMask* (located within the main file).
3. **PREPARE DARK COUNTS FOR SUBTRACTION FROM DATA**
   In this section, the mean dark current image is computed. If no such TIF file is identified in Section 1, the mean dark current image is set to 0 in all pixels.
4. **CHOOSE EMITTER AND REPRESENTATIVE BACKGROUND REGION**
   In this section, the user identifies the region within the image data from which the PSF will be cropped and from which the background will be computed.

- The user is prompted to select the PSF from the first frame of the data. The box used to select the PSF can be rescaled to the appropriate size, but its aspect ratio is fixed to 1 by the last argument provided to *choose_roi.m*
- All frames of the data are added together, and the user is prompted to select a representative background region from the resulting image.
    - The background region should not include any PSFs and should have counts representative of those expected near the PSF identified with the previous selection.
    - The background region is a rectangle with any aspect ratio or size.

5. **GET EXPERIMENTAL PARAMETERS FROM USER, IMPORT DATA**
   In this section, the user specifies some experimental parameters required by the algorithm.
    - Parameters include:
        - initial z0 [meters] – distance of the emitter away from the focal plane (positive z0 is further away from the objective)
        - z step size [meters] – step size of the scan (recommended: 250 nm)
        - final z0 [meters] – distance of the emitter away from the focal plane at the end of the scan
            - The vector produced, going from the initial z0 to the final z0 with the z step size indicated, should be of a length equaling the total number of frames. That is, each frame of the PSF data should have a unique z value within the vector, and the vector should be arranged in the same order as the PSF data.
        - measurement EM gain – electron multiplication gain used during data acquisition
        - emission wavelength [meters] – single representative emission wavelength (recommended: mean of emission spectrum)
        - imaging medium refractive index – in this code, the imaging/immersion/sample media are all assumed to be index-matched, so this one refractive index should describe the entire system
        - objective lens NA
        - microscope magnification – total microscope magnification
        - 4f lens focal length [meters] – focal length of 4f lenses is used to calculate the radius of the electric field in the Fourier plane
        - pixel size [meters] – pixel size in object space (should correspond to the physical dimension of the detector pixels divided by total magnification)
        - camera conversion gain – conversion gain of the detector (electrons per ADC count)
    - The physical mask pattern is used, in combination with the emission wavelength, to calculate the phase pattern applied in the Fourier plane via *calc_phase_fromPhysMask*
    - The calibration data is imported from the TIF file containing the PSF images. Dark counts are subtracted, and EM gain and conversion gain are used to convert counts to photons.
        - Pixels with negative values (which occur after dark counts subtraction) are set to 0 to avoid biasing the MLE algorithm.
        - The region imported is the one selected in section 4.

6. **ESTIMATE BACKGROUND AND SIGNAL**
   In this section, the background is estimated using the region identified in section 4 by taking the mean of the photons detected in the first frame. The signal is estimated by computing the sum of photons in the frame of PSF data nearest to z=0 and subtracting the background photons expected in the same size region.
7. **GET ALGORITHM FEATURES FROM USER**
   In this section, the user specifies some algorithm parameters.
   - Parameters include:
     - mask angle [degrees] – rotation angle of the physical phase mask used to take the data relative to the phase mask file used.
       - If 'unknown' is entered and the tetrapod mask is used, the helper function *calc_tetrapod_angle.m* will be used to calculate the angle from the position of the lobes formed by the tetrapod PSF with the largest z0 (e.g. +3 microns). If another phase mask is used, the exact angle must be entered.
     - number of orders of Zernike polynomials – specifies number of Zernike polynomials to use, which equals $\frac{(n+1)(n+2)}{2}$ for n orders. Recommended n=4 or 5.
     - allow lateral misalignment of mask? [yes/no] – enter 'yes' if lateral misalignment of the phase mask should be included as a parameter of the optimization.
   - Two additional parameters are hardcoded here:
     - gBlur is the standard deviation of a Gaussian blur filter applied to images created by *calc_img_fromCoeffVec.m*. The recommended value for 100 nm fluorescent beads is ~0.5-0.7.
     - resizeFactor is the reciprocal of the oversampling factor in the image plane. The recommended value is ~1/4. Lower values increase computation time; higher values increase sampling errors.
8. **CALCULATE COORDINATE SYSTEMS**
   In this section, the experimental parameters are used to determine the size of the electric field, and polar coordinates of corresponding size are created.
9. **REGISTER X-Y**
   In this section, a coarse X-Y registration of the model with the PSF data is performed using the dftregistration.m *function.*
   - Each frame is registered with the unaberrated model individually, and then the mean shift in each axis is found. This leads to a single XY shift vector which is used for the rest of the algorithm (mean_x0y0).
10. **NORMALIZE ALL FIT PARAMETERS**
    In this section, coefficients are normalized so they can be in a more useful format for the fmincon *optimization routine.*
11. **SET OPTIMIZATION OPTIONS**
    In this section, options are set for the fmincon *routine.*

12. **RUN OPTIMIZATION**
    In this section, a constrained optimization is run to determine the optimal Zernike mode coefficients. The resulting pupil function is used to produce a PSF.
13. **UPDATE PUPILFUNC STRUCTURE**
    In this section, the pupilFunc structure is updated with the phase retrieval results.
14. **DETERMINE RESIDUAL XYZ BIASES**
    In this section, the residual biases are determined by fitting the PSF calibration data with the new phase-retrieved model.
15. **PLOT AND FIT BIAS RESULTS**
    In this section, the results of the position bias fits are plotted. Before seeing the plot, the user must choose which order of polynomial with which to fit the results.
    - The user has the option to change the polynomial fit order.
    - The resulting figure is saved as 'xyz_bias_results.fig'
16. **UPDATE PUPILFUNC STRUCTURE**
    In this section, the pupilFunc structure is updated with the bias fit results.
17. **PLOT PHASE RETRIEVAL RESULTS**
    In this section, the PSF data and the final phase-retrieved model PSF are plotted.
    - A plot of all the PSF slices in the raw data is saved as 'raw_psf_data.fig'
    - A plot of all the corresponding phase-retrieved model PSF images is saved as 'mlepr_psf_model.fig'
    - A plot of the final pupil function is saved as 'pupil_field.fig'
    - A plot of the final aberration is saved as 'phase_aberration.fig'
    - A plot of the decomposition of the phase aberration into Zernike polynomials is saved as 'zernike_decomposition.fig'
18. **SAVE PUPIL FUNCTION**
    In this section, the pupil function data is saved.
    - All useful data is stored within the pupilFunc structure, which is saved as 'pupilFunc.mat'

2. LOCALIZING PSF DATA

This portion of the code is intended to demonstrate the how the localization process is carried out using the pupil function obtained in the previous section. The code performs the key steps of single-emitter fitting and bias correction, but requires manual identification of regions of interest (ROIs) containing PSFs. We encourage users to customize the provided function, LOCALIZE_PSF_DATA.m, to suit their analysis needs. Obviously, this code can be parallelized and optimized for special cases. Below, the operation of this function is described.

Files needed:

- **Superresolution or single-particle tracking data to be analyzed, in units of camera ADC counts (.TIF)**
    - This data can be the entire field of view. There is a cropping step within the code (Section 4 below), in which a single emitter will be chosen.
- **Dark current data in units of camera ADC counts (.TIF) – optional**

    - This data must be the same size as the PSF data in x-y, but can be a different number of frames. Within the code, the average value across all frames is taken for each pixel, and this number is subtracted from all frames of the PSF data file.
    - Pressing 'Cancel' when prompted for the location of this file will skip ahead and no dark current subtraction will be applied.
- **Pupil function from ESTIMATE_PUPILFUNC (.MAT)**
    - This should be the output of ESTIMATE_PUPILFUNC.m. The file name is not important but the variable should be called *pupilFunc* (which is the name under which the pupil function is saved at the end of ESTIMATE_PUPILFUNC.m).

Anatomy of **LOCALIZE_PSF_DATA.m**:

1. **GET FILES AND LOCATIONS FROM USER**
   In this section, the user identifies the location of necessary files and chooses a directory in which to save results.
    - When prompted, locate the three files listed above.
    - If dark current file is unavailable, press 'Cancel' when prompted for its location.
    - The directory identified for saving results (the fourth prompt) will have a MAT file saved to it containing the localization results.
2. **GET EXPERIMENTAL PARAMETERS FROM USER**
   In this section, the user specifies some needed parameters.
    - Parameters include:
        - maximum z for model [meters] – highest value of z to be simulated (furthest emitter position away from the objective)
        - z step size [meters] – step size between simulated PSF models. Regions between the steps are reached via interpolation of the PSF, so a subdiffraction step size is recommended (e.g. 10^{-7} m).
        - minimum z for model [meters] – lowest value of z to be simulated (nearest emitter position to the objective)
        - measurement EM gain – electron multiplication gain used in acquiring the data selected at the first prompt in the previous section
        - desired ROI size [pixels] – this is the size of the box in which a single-emitter fit will be conducted; it should fit the entire PSF for best results. If an even number is entered, it will be converted to an odd number.
3. **PREPARE DARK COUNTS FOR SUBTRACTION FROM DATA**
   In this section, the mean dark current image is computed. If no such TIF file is identified in Section 1, the mean dark current image is set to 0 in all pixels.
4. **PREPARE PSF DICTIONARY FROM PUPIL FUNCTION**
   In this section, the pupil function selected in Section 1 is used, along with the parameters specified in Section 2, to produce a dictionary of PSF images.
5. **PERFORM LOCALIZATION ANALYSIS**
   In this section, the localization analysis is carried out by going through the data frame by frame. The code periodically updates a figure with the handle frameFig and stores results in a structure called locResults.

- At the start of each frame, the user is prompted to select all emitters that should be localized. All PSFs should be clicked manually, roughly in their centers. Once the 'Enter' key is pressed, the selection step ends and the analysis of the identified ROIs begins.
    - If no PSFs are observed in a frame, 'Enter' may be pressed without selecting any emitters in order to proceed to the next frame.
- Once ROIs are identified, each ROI is individually addressed.
    - Each ROI is checked to make sure that the location of the corresponding click can be the center of a box which is contained within the frame. If this is not possible, this ROI is skipped. ROIs which are contained within the frame are analyzed for a PSF fit.
    - First, *find_dicMatch.m* is used to identify the image within the PSF dictionary that most closely matches the data in the ROI.
    - The output of *find_dicMatch* serves as a "coarse" axial fit, which is used by *localize_emitter_mleInterp.m* to perform a fine localization via interpolation of the PSF dictionary. The fine fit returns the maximum likelihood estimates ("mle") of five parameters: x, y, z, total signal photons, and background photons per pixel.
        - The output optImg is the optimal PSF model image (in units of photons/2 if EMgain>1; otherwise, in units of photons), as calculated from the five parameters identified by the MLE fit.
        - The output boundFlag is 1 if an upper bound is hit in any of the five parameters and -1 if a lower bound is hit in any of the five parameters.
        - bgQuant indicates which quantile of the pixel intensities in the ROI was used for background estimation. High values of this number often correspond to bad fits.
    - Finally, the function *apply_locBias.m* takes in the MLE fits, pupil function, EM gain, and global location of the ROI within the frame, and outputs a bias-adjusted ("adj") vector of the same five parameters. These parameters are stored in the 'fits' field of the locResults structure.
        - To access the results of the i^{th} fit of the j^{th} frame, use:
          `locResults.fits{j}(i,:)`
        - The columns of the 'fits' field are as follows:
            1. Frame number
            2. PSF number (in order of identification within the frame)
            3. Adjusted x [meters]
            4. Adjusted y [meters]
            5. Adjusted z [meters]
            6. Adjusted signal photons
            7. Adjusted background photons per pixel
            8. boundFlag
            9. bgQuant

6. **UPDATE LOCRESULTS STRUCTURE**
   In this section, the locResults structure is updated with some key parameters of the analysis so

they can be recalled easily with the localization data, which is stored within the 'fits' field of locResults.

7. **SAVE RESULTS**

   In this section, the localization data is saved.

   - All useful data is stored within the pupilFunc structure, which is saved as 'locResults.mat'