

This MATLAB code functions to three-dimensionally track non-fluorescent beads using quadrant interpolation and a look up table of off-focus images¹. This code is set up to process multiple videos back-to-back with frames stored as grayscale tif images. The images can be stored in a different file from this code.

This code requires the 'Computer Vision Toolbox' add-on.

Image names

It processes images with the following name convention: Basenames, video number, _, 3 digit frame number. The first frame in a video does **not** have to be labeled as 001.

For example: beads_2_034.tif

Where 'basenames' = 'beads_'

Video number = 2

Frame number = 34

Reference images are assumed to have a similar naming convention but without the video number.

For example: Z_stack_000.tif where basename (for reference images the corresponding variable is 'Z_stack_name') = 'Z_stack_'

The user can specify the above image information in the first section of the function

'TrackManyVids.mat' using the following variables:

- 'basenames' is a cell array containing all basenames of interest entered as strings. Multiple basenames can be entered within the curly braces as separate strings.
- 'strtvid' is first video number to analyze.
- 'endvid' is last video number to analyze. The function will create an array ranging from strtvid to endvid and look for frames with a video number for each value in the array.
- 'startframe' indicates the first frame of the video(s) to analyze ('1' for first the frame of video, even if the first frame of the video has a frame number of 0).
- 'endframe' indicate last frame to analyze video(s)
- 'frame_val_frstframe' corrects the frame numbering in case the first frame of a video is numbered '000'.

Model Matrix (Look up table of reference images)

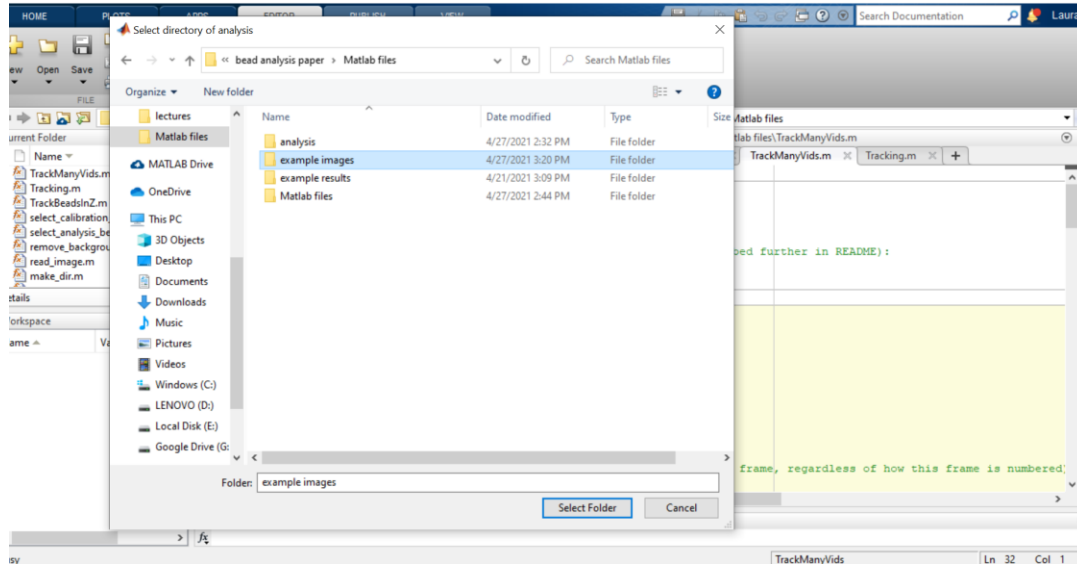
In line 28 of 'TrackManyVids.mat', the user enters a path for the location of a model matrix. If a model matrix can be found in this location, then this matrix will be used as the look up table for estimating axial positions of analysis beads. If no matrix is found, one will be created and saved at the location initially entered.

The format of the model matrix is a 3-dimensional array of doubles. Each row contains the radial profiles for a bead at each axial position as defined by 'distanceVector'. Each column contains the pixel intensity at each radial step. The range of radial steps spans the centermost pixel to the value of 'end_raxis' with step sizes equal to 'rstep'. The third dimension stores the corresponding values for each reference bead. During analysis, the best n beads (n defined by 'modelMat_num2use') will be averaged together along the 3rd dimension to create an optimized model matrix for each analysis bead.

Initiating analysis

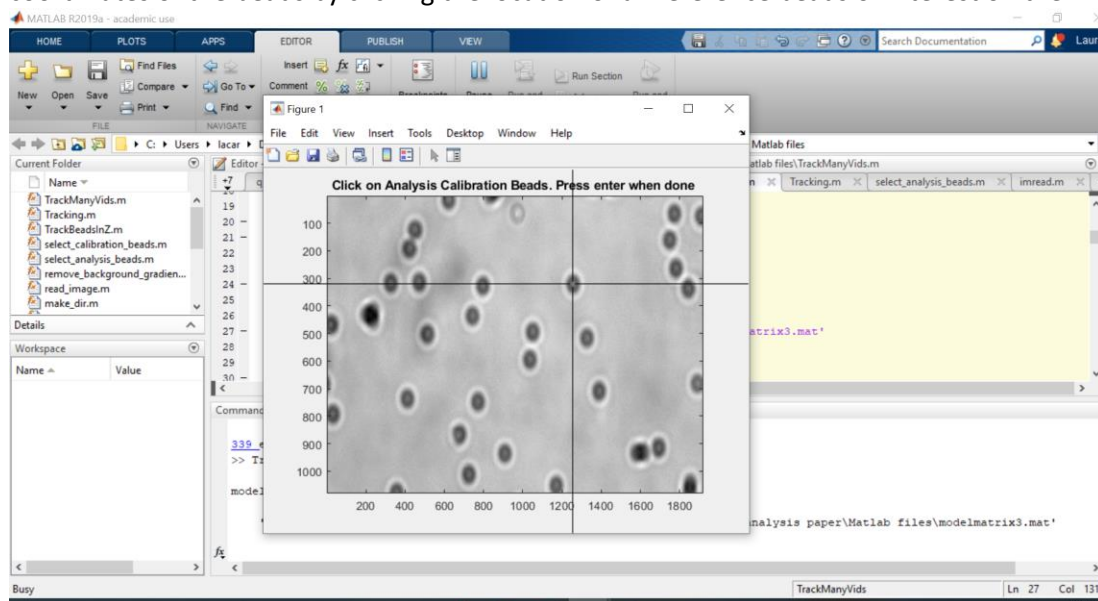
Before running this code, ensure user input parameters described above in 'TrackManyVids' are correct (lines 9-71).

Run 'TrackManyVids' and the first window to appear will be a file selection box where the user chooses the file where all image files are located.



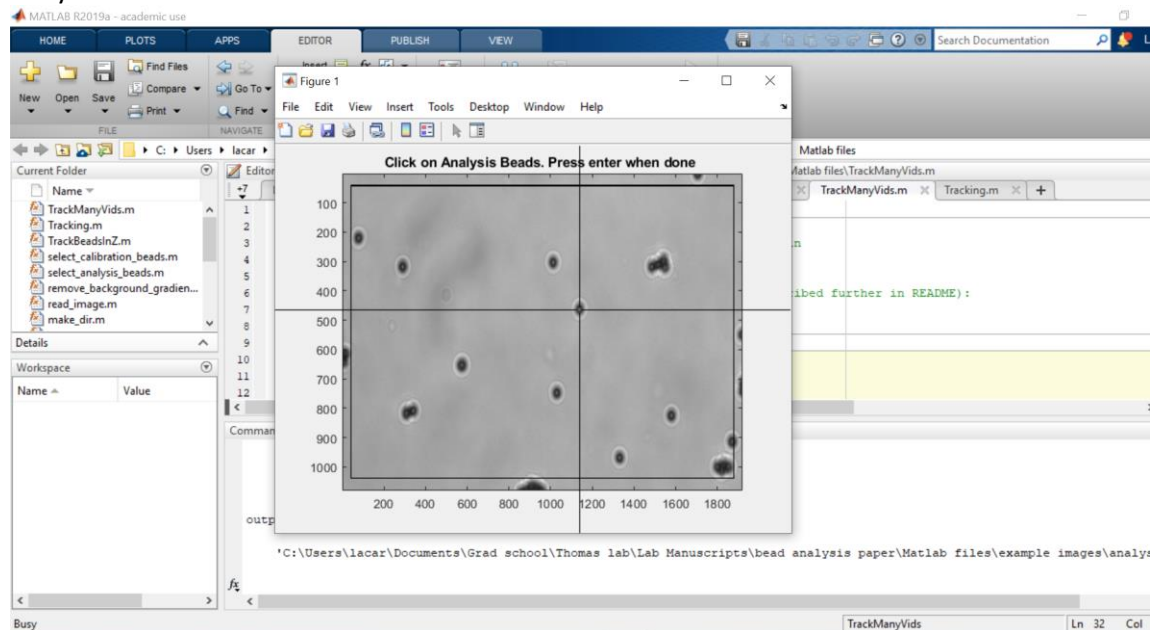
A folder called 'Analysis' will be created inside the chosen folder and all output results will be placed in this folder.

The code will then look for a model matrix in the path specified in the variable 'modelMatrix_path'. If none exists in the location entered the code will create one using image specified with the naming scheme described above. These images must be located in the same folder as the analysis images. The first image of the stack of reference images will be brought up and the user will be asked to input the coordinates of the beads by clicking the location of all reference beads of interest on the image.

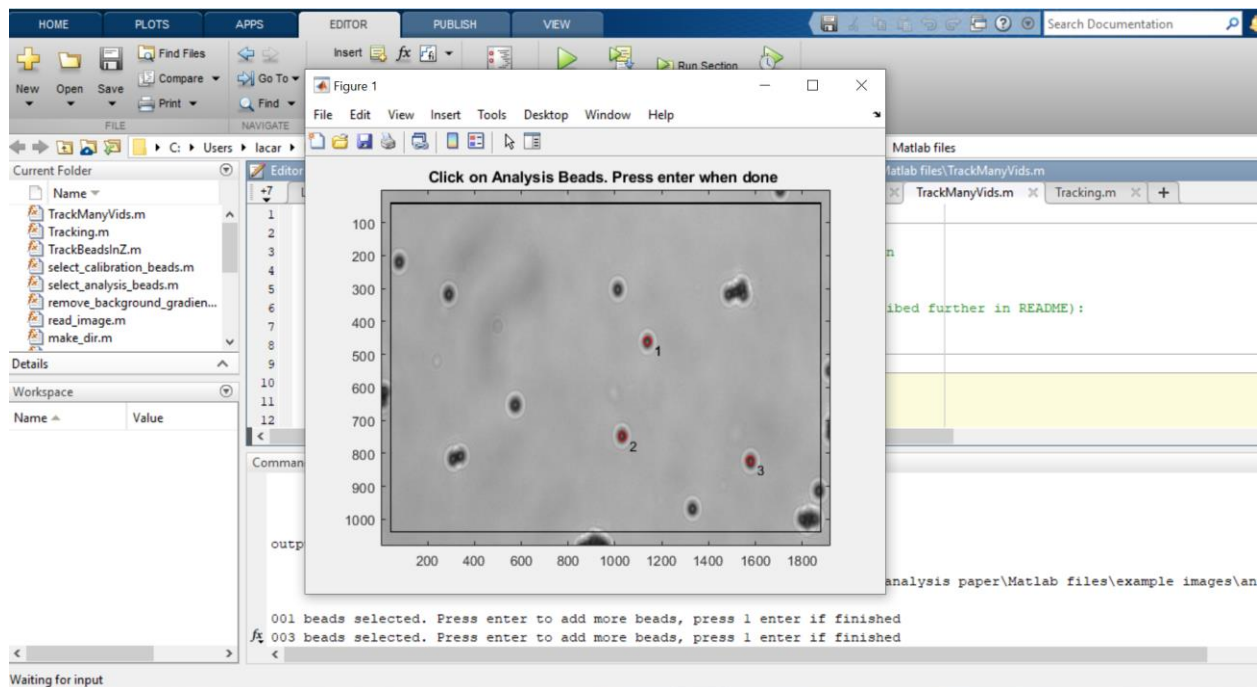


After selecting all beads press 'enter' to indicate you are done. Immediately after clicking a location on the image there will be no visible change to the image. After pressing 'enter' an image will be saved with numbered red circles over the location of the input coordinates will be stored in the same folder the model matrix. The open figure in which the user input the bead coordinates will automatically close. The code will process the reference images and create the model matrix before continuing to the next step. This may take, with a 100 frame z-stack, about 10-15 seconds per selected bead.

After the model matrix is created, the code will loop through each video and bring up the first image of each, one at a time and ask the user to click on the center of each bead. At this point Matlab will be 'busy'.



To help the user keep track of the beads selected in each video, the user can hit 'enter' and red circles will appear over all selected beads. At this point control will move from the figure to the Command Window and Matlab will be 'waiting for input'.



Unlike when selecting beads for the model matrix, the user has the option to continue selecting beads by hitting 'enter' again in the command window. When the user is done and wants to go onto the next video, they need to hit enter so the latest input coordinates appear as red circles on the image and type '1' 'enter' in the command window. The current figure will close and be replaced by the first frame of the next video.

If the code cannot find an image with the combined base name and video number, it will skip the video name, display an error, and move onto the next without erroring out. The user may need to scroll in the command window to find the error.

Below is an example of the Command Window output in which the code was unable to find the location of the first frame of a video. This may be due to selection of the incorrect folder, incorrect typing of the base name, or incorrect identification of the video number or first frame number.

MException with properties:

```

    identifier: 'MATLAB:imagesci:imread:fileDoesNotExist'
    message: 'File "C:\Users\lacar\Desktop\8.3_2_000.tif" does not exist.'
    cause: {0x1 cell}
    stack: [4x1 struct]
    Correction: []

```

skipping to next video

```
total_vids =
```

```
0
```

When the user has input the initial locations of beads in all videos, all figures will close and Matlab will automatically begin calculating the positions of beads through all videos. Depending on the number of beads and size of videos, this may take several minutes.

```
005 beads selected. Press enter to add more beads, press 1 enter if finished 1  
Initial bead positions recieved successfully. Beginning analysis of 2 videos
```

```
vid_num =  
  
1
```

Expected Output

A new folder will be created for each video analyzed. The folder will contain a jpeg of the first image of the video with numbered red circles over each bead, a text file containing the user input parameters used to analyze the video, and the bead positions in x,y, and z in both an Excel file and a MATLAB data file. For both data files, columns correspond to different beads with each row corresponding to a frame.