



APR 10, 2024

OPEN ACCESS



DOI:

dx.doi.org/10.17504/protocols.io.n2bvjny65gk5/v1

Protocol Citation: Katie A. Bunde, Weiyuan Fan, Dimitrije Stamenovic, Paul E. Barbone, Michael L. Smith 2024. An improved image analysis method for micropattern traction microscopy: dot tracking and traction force calculation script protocols. **protocols.io** <https://dx.doi.org/10.17504/protocols.io.n2bvjny65gk5/v1>

License: This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

An improved image analysis method for micropattern traction microscopy: dot tracking and traction force calculation script protocols

Katie A. Bunde¹, Weiyuan Fan², Dimitrije Stamenovic^{1,3}, Paul E. Barbone², Michael L. Smith¹

¹Department of Biomedical Engineering, Boston University, Boston, Massachusetts, United States of America;

²Department of Mechanical Engineering, Boston University, Boston, Massachusetts, United States of America;

³Division of Materials Science & Engineering, Boston University, Boston, Massachusetts, United States of America



Katie A. Bunde
Boston University

ABSTRACT

The dot tracking script takes timelapse images of fluorescent micropatterns and determines the frame-by-frame displacement of each fluorescent dot. From these displacement values, the traction force calculation script calculates the corresponding traction force values using the known material properties of the gel being imaged, as described in Equation 2 of the main text. It can also plot either traction or displacement maps, user's choice.

Final x and y displacement vectors (taken from dot displacement script and thresholded based on the chosen displacement threshold value) are saved to the variables "x_dis_final_thresh" and "y_dis_final_thresh," and the magnitude of these vectors is saved to "magD." Unbalanced traction force vectors are saved to "T1" and "T2," while balanced traction force vectors (for static equilibrium) are saved to "feX" and "feY," respectively. The magnitudes of the unbalanced and balanced traction force vectors are saved to "magT_UB" and "magT_B," respectively.

Protocol status: Working

We use this protocol and it's working

Created: Apr 10, 2024

Last Modified: Apr 10, 2024

PROTOCOL integer ID: 98039

Funders Acknowledgement:

National Science Foundation

Grant ID: CMMI-1910401

Dot Tracking Script

- 1** Name all timelapse images you plan to analyze in order according to the format "filename01, filename02, filename03, etc." for as many images as you have. Our group typically has a stack of 25 images, so the number of images would go up to "filename25," but any number of files can be used. Save these images to a designated folder, and copy and paste the file location for that folder onto *line 12* of the Dot Tracking Script; this is where the results of the dot tracking script will be saved. The location of the folder where the Dot Tracking Script and other Matlab files are located should be copy and pasted onto *lines 11, 36, 420, 490, and 505*.
- 2** Threshold and binarize your chosen undeformed reference pattern in ImageJ, making sure to not threshold so high that dot size is significantly reduced or any dots are thresholded away. Make sure the final image has a dark background with the dots being white. Save this file into the same folder as your experimental images ("__01, __02, __03, etc.") and name the file accordingly, but with the numbers "00" following the name of the file (i.e. "filename00" added to the other files from Step 1)
- 3** Enter in the name of your reference image file (i.e. "filename00") and the first of the deformed experimental images ("filename01") into *lines 18 and 14*, respectively. Enter "*filename*" into *line 25* to designate the filename the code should be looking for, that way it will analyze all images with that name in the order of the numbers following it (i.e. filename00 > filename01 > filename02 > ... > etc.).
- 4** Enter your desired starting threshold value into *line 34*; this value can be changed later while the code is being run, so just make a starting estimation here.

- 5 Run the script; Matlab will then display an image of the first deformed image and ask you to draw a rectangular range of interest (ROI). You can do this by pressing down on the left mouse trigger in one corner of the pattern, dragging it diagonally across the pattern to draw a rectangular ROI, then releasing the trigger. Be sure to leave a bit of space between the edge of the pattern and the ROI, and not to just try and draw the ROI as close to the edge of the island as possible.
- 6 Once the ROI has been chosen, the code will display the thresholded and binarized version of the first deformed image based on the threshold you inputted before running the code. The code will then ask you if you want to change the threshold of your images. If correctly thresholded, the image should show the fluorescent dots (yellow) as close as possible to their size in the original image, with a blue background; in this case, select “No” when prompted if you want to change the threshold. If the threshold is too low, the yellow dots will appear too large; if the threshold is too high, the dots will appear too small or may even not even be visible. In either case, select “Yes” when asked if you want to change the threshold, input a new value into the command window, and click the “Enter” key. Repeat this as many times as necessary until you’ve found the optimal threshold for the images being analyzed.
- 7 Once the proper threshold value has been chosen, the code will then display the first experimental image with all detected dots outlined with a red circle. You will be asked if you want to manually select dots, which you can do for any dots which are present but not detected by the code. To do this, select “Yes” when prompted and click as close to the center of the dot which you want to add as possible. Repeat this as many times as necessary to ensure that all present dots are included in the analysis. Once you are finished – or if no additional dots need to be selected – select “No” when prompted to select additional dots.
- 8 You will then be prompted to de-select dots, which you can do if the code locates dots in areas where no dots are present. If this step is not needed, just select “No” when prompted. If some dots do need to be removed, select “Yes” and click in the center of the red ring which outlines the pseudo-dot. Repeat as necessary, selecting “No” when prompted once you’re finished.
- 9 The code will then ask if you would like to select center dots, which you should select “Yes” to do when prompted. One at a time, draw a rectangular ROI around the centermost dots in a pattern; for a square pattern that would be the four center dots, and for a circular pattern that would be the first ring of dots surrounding the centermost dot. The code will then determine the center point of all the dots you select, giving you as accurate a representation of the center point as possible.
- 10 You then must repeat the same steps in Step 9 for the reference image, for which you should select dots which are in the same locations as those you selected in Step 9. If the code did not determine the locations of the fluorescent dots accurately (i.e. red circles denoting dot location are in the wrong place), you may need to adjust the thresholding of the reference image using ImageJ.
- 11 The code will then go frame by frame and calculate dot displacement between each frame (i.e. reference > frame 1, frame 1 > frame 2, etc.). When it’s finished it will output multiple figures, two of which the code is

written to save in the folder you designated at the start, alongside the frame-by-frame displacement measurements. The two saved figures are fig9, which shows the reference pattern (green) overlaid over the first frame of the experimental timelapse images (purple), similar to what we show in Fig 2; and fig12, which shows the total sum of displacements throughout the entire experiment.

Traction Force Calculation Script

- 12 In *line 6*, input the file location where your dot displacement data was saved following the use of the dot tracking script.
- 13 In *line 8*, input the filename the dot tracking data was saved to; by default, our code saves the data as "Timepoint_" with the corresponding frame number on the end.
- 14 In *line 10*, input the number of files you have total with the name "Timepoint_".
- 15 In *line 11*, input the folder location where you've saved your Matlab files, including those from this article.
- 16 In *lines 14-17* we have listed a few possible values for Young's Modulus (E); comment out all values of E except the correct value for your hydrogel, adding different values if you used a stiffness other than the four values we listed.
- 17 In *line 19* input the Poisson's ratio (ν) of your hydrogel, for us this value is always 0.445 so that's the default value.
- 18 In *lines 24 and 25* respectively input the approximate radius of your dots (in mm) and the pixel ratio of your imaging system ($\mu\text{m}/\text{pixel}$).

- 19** In *lines 31* and *32* are two possible displacement threshold values: 0 μm (no threshold) or 0.3 μm (our chosen threshold). Comment out the value you don't want to use, or enter in a new value if you would prefer to use a different threshold for displacement value.

- 20** In *lines 122 – 134* are options for which type of quiver plot you'd like the code to display (unbalanced traction forces, balanced traction forces, or displacement values); comment out all but the type of plot you want, or comment all out if you don't want to make plots and just want to calculate traction force values.

- 21** Run the code, and it will calculate traction force values and plot the displacement or traction force maps of your choice (one for each frame).