CSE554 Final Project Report
Tracking Microtubule

Weijie Gan

December 10, 2018

Contents

1 Introduction

1.1 Project Requirement

In this project, we was required to detect and track one microtubule form video with many microtubules. The microtubule in video grows, which means its length will change and position will be moved slightly.

The tracking result can be shown as a mask binary image or a skeleton. A demo output is shown as figure [1], where the left image is one frame image extracted from video and the right image is the output mask for center microtubule in the original image. What we need to do is to produce a similar mask video.

Based on the professor' s slide, my own wishlist in project proposal is:

- A GUI can load and show video contain moving microtubules.
- The user is allowed to select a certain microtubule.
- Selected microtubule can be shown separately in each frame of the video as a mask binary image.
- The program works for highly dynamic microtubules.
- Recording growth event, duration, and rate of selected microtubule.

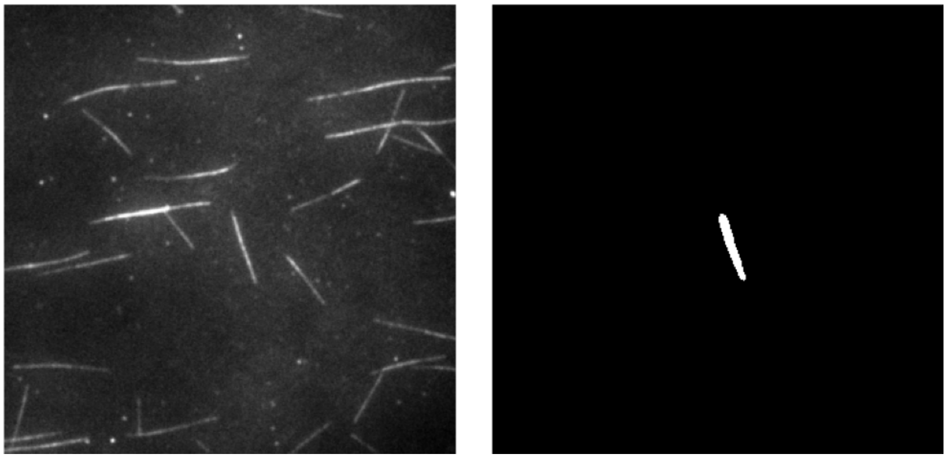

Figure 1: Demo Output Mask

1.2 Project Completion

Compared with my wishlist, the FINISHED wishlist of this project is:

- I designed GUI can load and show video.
- The user is allowed to choice a certain microtubule by Clicking a Point in the video.
- Selected microtubule can be shown separately in each frame of the video as a mask binary image in the GUI.
- I record growth event, duration, and rate of selected microtubule as a png file.

The UNFINISHED wishlist of this project is:

- My algorithm only works for limited amount of microtubule in video.
- Sometime the programme is not stable.
- The separation is not perfect.

1.3 Project Implement

Implement details is introduced as follow:

- Programme language: Python
- GUI Interface: PyQt5(Qt5 in Python API)
- Image Process Library: cv2(OpenCV in Python API) and PIL (Python Image Library)
- Machine Learning Library: sklearn

2 Algorithm

In this part, the process of our project and core algorithm will be introduced.

2.1 Preprocess

At the beginning, the contrast of video frames will be enhanced, for make microtubule more obvious in image. However, if we directly enhance contrast, noised data will also be enhanced in image. So, we need to do Gaussian blur method to reduce noise in image. The demo result is shown as figure [2], where the left image is original image, median image is image processed by Gaussian blur and the image processed by both Gaussian blur and contrast enhance is shown in the right place.

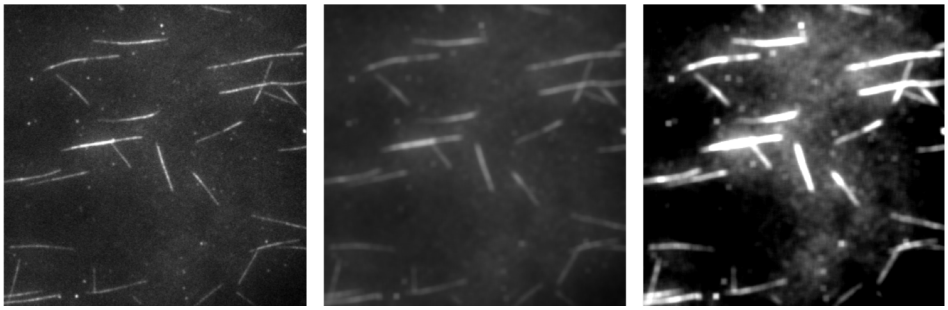

Figure 2: Demo: Preprocess

2.2 Step 1: Choice Point of Interest

The first step of algorithm is to choice a point of interest(POI). In the first frame, this point is selected by user and microtubule contains this point is seen as our target microtubule, which need to be separated in our project. The demo of selecting a POI (Shown as read point) in GUI interface is shown as figure [3],

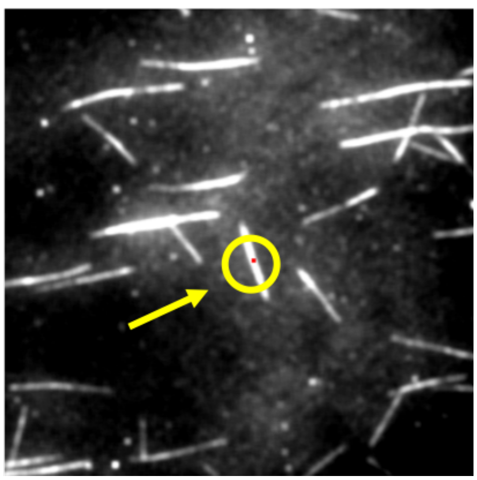

Figure 3: Demo: Choice Point of Interest

In the other frames of video, the POI is updated in the same microtubule (Introduced in Step 6).

2.3 Step 2: Thresholding

In this step, value of thresholding is obtained by the mean value of POI neighbor $3*3$ region. A demo result is shown figure [4], where the right image is threshold binary image.

2.4 Step 3: Compute Connected Component

In this step, we compute connected component for the whole threshold binary image with 4-connected rule. Then, the connected component that have the lowest distance with POI is selected. A demo result is shown

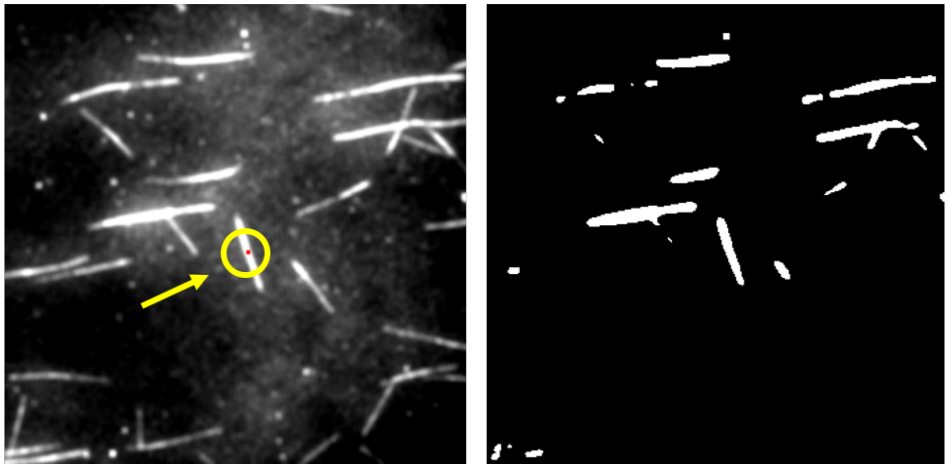

Figure 4: Demo: Thresholding

as figure [5], where the left image in binary image, the median image show all connected component and the right image shown the connected component related with POI. Actually, the result of this step is the mask image that we need to solve for target microtubule. Remaining step is for compute mask image iteratively.

Figure 5: Demo: Compute Connected Component

2.5 Step 4: Compute Supposed Line Function

The core algorithm for this project is to suppose growing direction of microtubule is in a line. What we need to do in this step is compute this supposed line function with Linear Regression. We firstly, compute non-zero points in mask image, shown as $(x_1, x_2), (x_2, y_2), ..., (x_n, y_n)$ where n means amount of non-zero points. Then the linear function is formulated as $y = ax + b$. The optimization problem will be:

$$a, b = \arg\min_{a,b} \sum^{n} ||y - x||^2$$

Concretely, we solve it by sklearn, which is introduced in the first part.

Especially, if we have line function(Introduced in the next step), the connected component result can only lie on place have limited distance with the line.

A demo result is shown as figure [6], where the left image is the mask image, and blue line in the right image is what we need to obtained in this step.

2.6 Step 5: Get Value from Frame along this Line Function

In the next frame, we compute value along line obtained in last step. A demo result is shown as figure [7], where the left image is the combination of image in next frame and the line, and the right image plot values

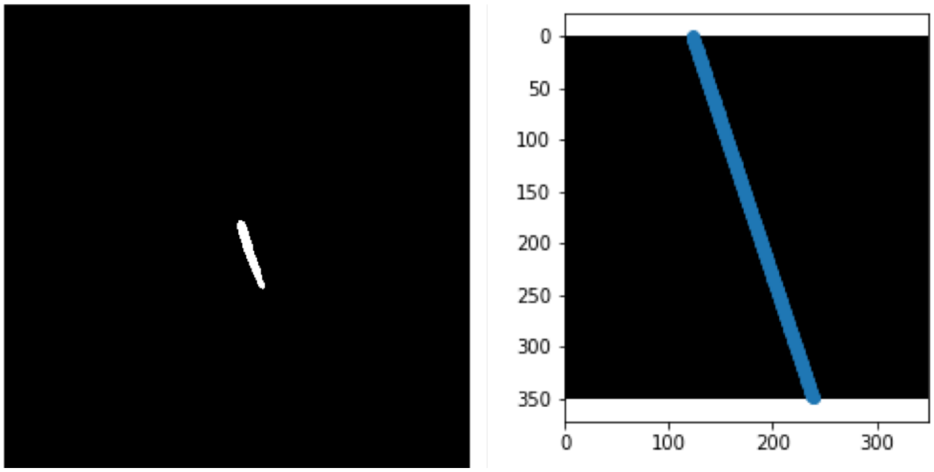

Figure 6: Demo: Compute Supposed Line Function

along the line in the image.

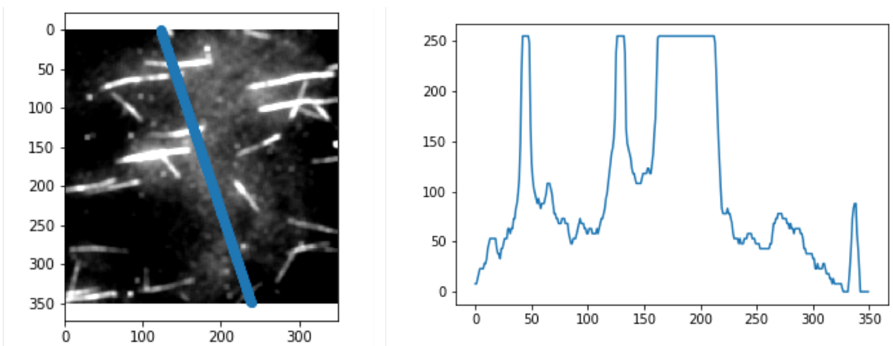

Figure 7: Demo: Get Value from Frame along this Line Function

2.7 Step 6: Find Length and Update a new ROI point

In this step, we used ROI point as starting point and search edge of microtubule in the plot. obviously, the edge will be the huge change of value in the plot. Based on this rule, we can find the length of microtubule as pixel distance between two edge in plot. Then, we updated new ROI point as the median position of two edge in plot.

A demo result is shown as figure [8], where the left image is the combination of image in next frame and the line, and the right image plot values along the line in the image. Especially, I draw three circle with different color, their positions in this two image are related.

After this step, we have updated a new POI and started process next frame, then return to Step 1 until all frames have been processed.

3 Drawback and Future Work

Based on experiment result, the drawback of my work is:

- Algorithm doesn't work when initial POI is in the intersection of two microtubule, due to mis-estimation of initial line function.
- Algorithm doesn't work in low brightness microtubule, due to mistake in value choice for thresholding.

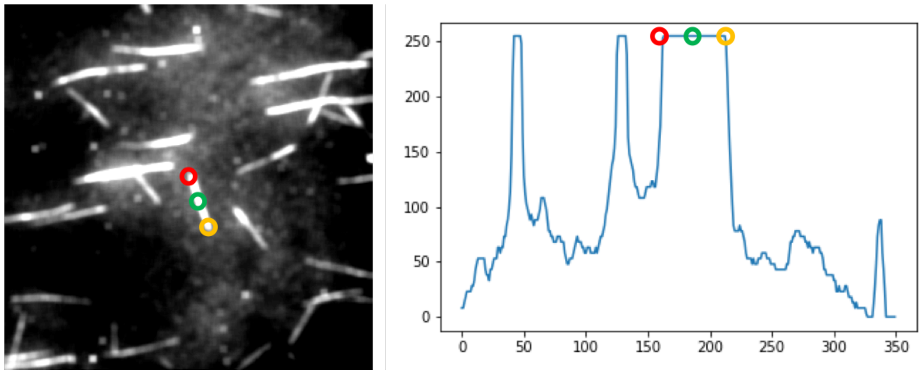

Figure 8: Demo: Find Length and Update a new ROI point

And the future of my work is:

- For the first drawback, to replace Point of Interest with Line of Interest along the target microtubule.
- Fpr the second drawback, to find a good way to balance brightness for every microtubule or do a background separation.

4 Readme for Demo

Firstly, install required python library will be:

- cv2 (OpenCV)
- numpy
- PIL
- sklearn
- PyQt5

Then, run main.py. The GUI will run as figure [9].

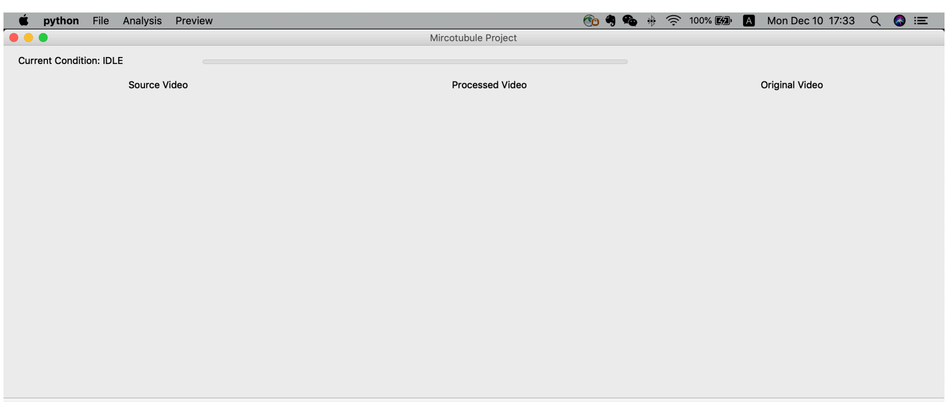

Figure 9: GUI

Click Analysis->SelectROI in menu bar. You can choice the POI in left image in GUI as figure [10].
Click Analysis->Start in menu bar. The algorithm will run one time as figure [11].
Click Preview->Play in menu bar, the process will be finished in all video.

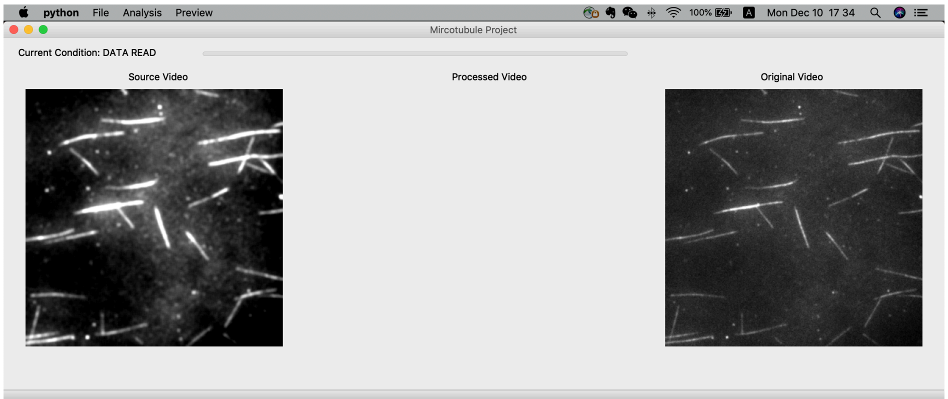

Figure 10: Choice POI

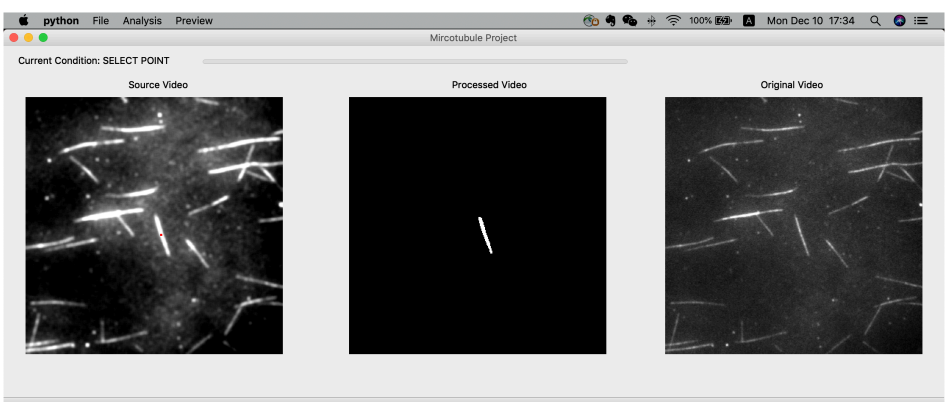

Figure 11: Algorithm Running

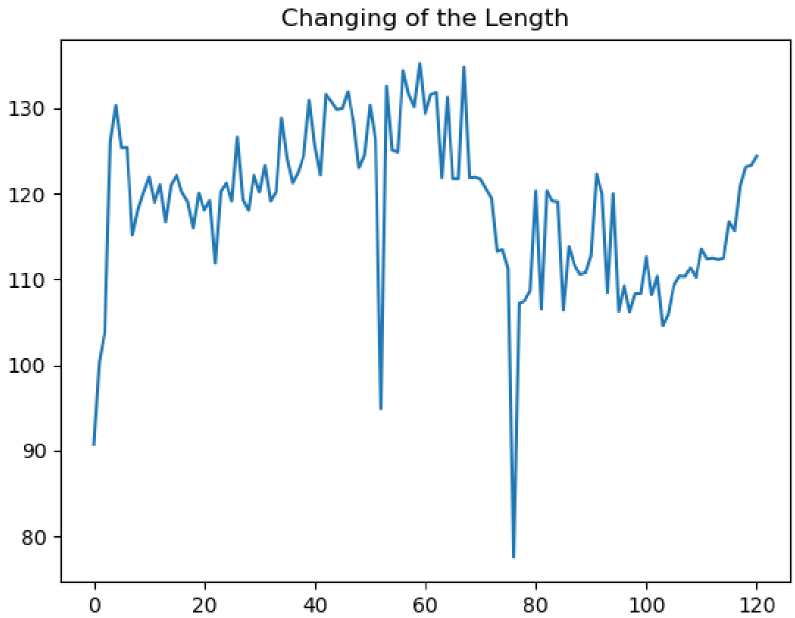

Figure 12: PlotResult

Finally, you can get changing of a plot about changing of length of target microtubule in root path as PlotResult.png as figure [12].

I have a video for this demo in my Google Drive:

https://drive.google.com/file/d/11G9OLbeq1GQM9e_PDTk29YPdt0eHBUIl/view?usp=sharing.

Especially, for success running, please use <MT dynamics example image stack.tiff> and the center microtubule, just like I do in the video.