Measuring Length of Sperm Cell of Fruit Flies

CSE 554A Final Project Report
Finley Li (465060)
Instructor: Prof. Tao Ju

Introduction

The longest sperm ever recorded comes from the fruit fly Drosophila bifurca, which is a total of 5.8 centimeters long, about 20 times longer than the fly itself. For years, scientists and researchers have been interested in measuring the length of the sperm cells from these fruit flies, as it serves as the key to further understanding their genetic makeup and lays the foundation for more interesting and impactful research on the topic. Sperm cell lengths are usually obtained through manual measurement, usually with the help of tools such as ImageJ's "Measure" feature, which could often be a laborious process. In this project, my goal is to design an interactive tool using Matlab that can not only help to automate the task for researchers but obtain an accurate measurement result as well. To that end, I will be utilizing the power of geometric computing algorithms such as flood-fill and cell complex thinning.

Methods

Algorithm:

Image Preprocessing

- *Rolling ball background subtraction*

  The rolling ball background subtraction corrects for unevenly illuminated background by implementing a rolling ball algorithm. For every pixel on the image, its pixel value is determined by taking the average of its neighboring pixel lying within a specified radius. The pixel value is then subtracted from the original image. The rolling ball background

subtraction effectively cleans up the debris and artifacts in the image while preserving the main objects. The algorithm is borrowed directly from Python's OpenCV library and called by the application.

- *Image brightening*

  The image brightening algorithm increases the contrast of a grayscale image. In specific, it saturates the bottom 1% and the top 1% of all pixel values. The method significantly increases the visibility of the sperm cells in dark images. The image brightening method implements Matlab's Imadjust() method.

- *Image thresholding*

  The image thresholding algorithm performs a simple value-based thresholding on grayscale images. The algorithm replaces each pixel in an image with a black pixel if it is less than the specified value while preserving those that are higher than the value. The algorithm successfully segments the images, separates the sperm cell from the background noises, and outputs a clean binary image that the main algorithms could run on.

Main Algorithms

- *Flood fill*

  The flood-fill algorithm is implemented to find the connected component containing an object pixel, which would help us identify the cell body since it is essentially a string of connected pixels. The algorithm detects the connected components to the specified pixel by performing depth-first search using a stack.

- *K largest components*

  The algorithm begins by finding all the connected components in the binary image by running the flood-fill algorithm on every object pixel. Each connected component is given an integer label and the K largest components are returned as a binary image in the end.

- *Cell Detection*

  The cell detection algorithm simply calls the K largest components method with the user-specified parameter K that indicates the number of sperm cells they want to extract

from the image. The resulting binary mask is then overlaid on top of the original image in red for better visualization.

- *Thinning*

  The algorithm performs 2-dimensional thinning on cell complex. The cell complex is first built from the extracted largest components returned by the Cell Detection method. A 2-dimensional array is returned with all the 0-cells representing the points stored on the first level, and all the 1-cells representing the edges stored on the second level. The algorithm then applies thinning on the cell complex, removing all the simple cells (noisy branches) while preserving the medial cells (the main structure of the sperm cells). The resulting cell complex is returned.

- *Length calculation*

  At the final step of the pipeline, the number of 0-cells in the thinned cell complex is determined and the final length of the sperm cell in micrometers is calculated by dividing the total number of 0-cells by the conversion factor 3.06.

GUI

The GUI is implemented with the help of Matlab's GUIDE graphical user interface designer. All the required features outlined in the project proposal are accomplished and all but one wishlist feature are successfully implemented. The omitted wishlist feature is the ability to click on a pixel of the image and obtain the intensity value. The feature was not implemented as I did not find it to be a useful and informative feature to be included in the GUI.

The GUI contains a total of 10 buttons that the users can interact with, all divided into three main groups. The first group on the upper left side of the main interface allows the user to perform basic operations such as uploading the image, resetting, or confirming the changes they make on the images.

The second group titled the "Image Processing Toolbox" provides the user with a variety of image preprocessing functionalities. "Subtract Background", as detailed in the algorithm section, implements rolling ball background subtraction and allows the users to clean up a noisy image tainted with debris and artifacts. "Brighten Image", when clicked, will increase the contrast of the image. The "Eraser" is a draggable rectangle that the users could use to further clean up an image by blackening out the irrelevant part of an image. Finally, the "Pen Tool"

enables free-hand drawing on the image. The tool, upon confirmation from the user, will burn the line drawn on the image and display the updated image. The pen tool allows users to explicitly trace out the body of the cell for better cell detection, which is helpful for low light or extremely noisy images where the cell bodies remain obscured even after being preprocessed.

The third group consists of two main buttons. The “Detect Cell” button displays the cell overlaid in red on top of the original image, provided that the user has preprocessed and thresholded the image correctly. Additionally, they can change the number of cells to be detected by entering different values for the parameter K. Lastly, the “Report Length” button returns the length of the highlighted sperm cells from the previous step.

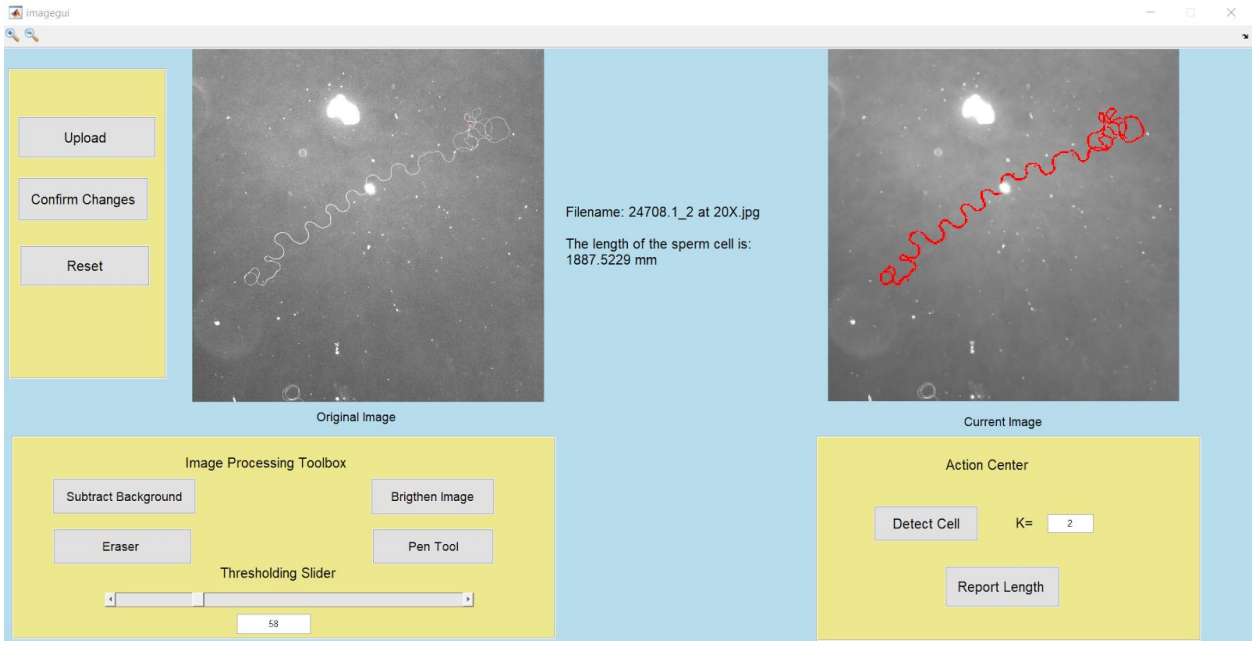

Figure 1. The main interface of the GUI

Data and experimental results

The application is tested with 3 classes of images (easy, medium, hard) and the measurement results were compared with the ground truth measurements by Prof. Ju. The comparison is shown below along with the calculated percent error. Note that all the results were calculated without the aid of the “Pen Tool” feature.

	ImageID	Length.Manual.mm	Measured Length (mm)	Error (%)
Hard	472.1A.1_1	1421.724	1101.7	22.50957288
	472.1A.1_2	1721.22	1644.38	4.46427534
	28369.2.6_2	1798.116	2098.08	16.68212729
	28369.2.6_3	1820.409	2179.5	19.72584183
	472.1A.1_5	1827.506	1704.54	6.7286236
	472.1A.1_4	1836.393	1566.72	14.68492855
	42568.b4.7	1840.172	1488.49	19.11136568
	LHM.1B.3_2&3	1847	1851.3878	0.237563617
	LHM.1B.3_7	1849.383	1827.48	1.18434094
	472.1A.1_3	1849.996	2073.03	12.05591796
	472.1B.1_5&6	1870.82	2183.2	16.69749094
	53387.1B.2_7&8	1873.806	2253.49	20.26271663
Medium	24708.1_4 at 20X	1681	1801.4	7.162403331
	24708.1_5 at 20X	1952	2235.95	14.54661885
	24708.1_6 at 20X	1991	2596.91	30.43244601
	WT.C.1	1090	1233.28	13.14495413
	WT.C.2	1847	2313.65	25.26529507
Easy	24708.1_1 at 20X	1951	2115.63	8.438236802
	24708.1_2 at 20X	1787	1902.56	6.466703973
	24708.1_3 at 20X	1786	1900.06	6.386338186

Table 1. Measured cell lengths in micrometers compared with the ground truth measurements.

As shown in the table, the application is able to maintain a percentage error in the range of 6.39% -8.44% for all the easy examples, 7.16%-30.43% for the medium examples, and 0.23%-22.51% for the hard examples. On average, the application obtains a 7.09% percent error on the easy examples, an 18% percent error for the medium examples, and 13% for the hard examples.

Bugs and future work

While the program performs reasonably well on the data given, improvements could certainly be made to improve the accuracy of the algorithms as well as the overall user experience.

One of the biggest issues with this application is its inconsistent calculation results. The successful calculation heavily hinges upon how well the users preprocess and clean up the images. Before the program could correctly detect the sperm cells and calculate their length, the

users would need to adjust the brightness and clarity of the images, threshold them properly so that the sperm cells are preserved, and filter/blacken out the potential distractions (extra noise or irrelevant information such as unwanted cells). If the users fail to perform a proper clean-up on the image, the resulting calculation could deviate significantly from the ground truth.

In addition, the reliance on users' help to preprocess the image also entails that the application is inconvenient to use as it is not fully automated. In the future, I aim to redesign the core algorithms and the main user interface so that fewer actions are required from the users to obtain an optimal calculation result.

Lastly, the length calculation algorithm tends to overestimate the length of the sperm cells, indicating that the method might need to be reworked. Upon close examination, I determined that the inaccuracy could potentially be explained by two main factors: image resizing and cell counting method. First, the 2048 x 2048 input images were resized to 13% of the original dimensions after being uploaded as cell complex thinning could be computationally expensive on large images. After the thinning was completed, the calculated length was converted to the actual length by multiplying to the scaling factor 1/0.13. What I did not take into account was the fact that the number of pixels on the skeleton of the cell may not correspond exactly to the scale. Consequently, more or fewer pixels could have accidentally been included during this extrapolation. Moreover, as part of the length calculation process, the number of remaining 0 cells in the thinned cell complex was counted to determine the final length of the sperm cell. However, it could be the case that the number of pixels is by no means an accurate count of the length of curves. Hence, to precisely measure the length of the curvy sperm cells would require a more sophisticated mathematical formula than a simple count of the number of pixels in the thinned-down skeletons.

Conclusion

Overall, my GUI provides a wide assortment of functionalities for image manipulation and performs length calculation on the sperm cells with reasonable accuracies. Meanwhile, some aspects of my implementation are still in need of reworking to make the tool more accurate, automated and intuitive to use.