MICROTUBULE FILAMENT TRACING AND ESTIMATION

Rohan Chabukswar

Department of Electrical and Computer Engineering
Carnegie Mellon University

ABSTRACT

The project explores preprocessing of images to facilitate tracing of microtubule filaments in three dimensional images accrued from fluorescence microscopy. The next step of the project is tracing of the filaments themselves, which is hoped to be completed shortly.

1. INTRODUCTION

Microtubules are filamentous cytoskeletal structures composed of tubulin protein subunits. These subunits can add on, or dissociate from, the tubulin polymer rapidly, making them highly dynamic. Microtubules are critically involved in many essential cellular functions, such as chromosome segregation at mitosis and intracellular cargo transport.

Motivation. Microtubules are generally studied using three dimensional fluorescence microscopy. The output of such a system is a 3D image of the microtubules, blurred due to factors like the lenses, the imaging device, sampling and digitization, and the finite size of the microtubules themselves. This blurring makes it difficult for automated analysis of the microtubules. Manual tracing of the tubules is a very inconvenient option, due to large size of images and visualization problems due to the 3D nature of the images.

Th aim of the project is to automatically trace each microtubule filament in the 3D microscope image. The traced image will be used to estimate the statistics of the filaments, like number of filaments, average length, distribution of length. The traces in the image should be as close as possible to the actual filaments for the statistics to have appropriate confidence intervals.

Related Work. Very little work has been done in the area till date. Sargin et al [1] have presented a microtubule body tracing algorithm that addresses the clutter without imposing directional constraints. Others have also presented algorithms that work in different cases. Most of the previous work has been done in two dimensions.

Organization of the Paper. This paper describes the different approaches that were implemented to thin the image. The actual tracing could not be completed due to time constraints, however the implementation is only a short step away from the results obtained at the end of the project. The input image is a simulation of the microscopy image, without noise. The point spread function is also given. Initial trials were also run on images of a smaller size mainly due to the time complexity of the algorithms involved. The smaller images were also less complex, contaning one or two filaments, which facilitated the development of the algorithms and choosing of specific parameters. This paper first describes the analysis on images contining one filament, then two intersecting filaments and then the simulation image. Due to difference in the basic characteristics of the images, the approaches used were different in each case.

2. BACKGROUND

The microtubule filaments originate from a common center and grow outwards. This results in the density of the filaments being less towards the periphery of the image than the center. The mechanism of the formation of these filaments dictates that the filaments grow in a straight line unless an obstacle exists, like cell organelles. Thus while tracing the image, a minimum curvature constraint can be imposed to prevent wrong tracing of the tubules. This is especially true at points of intersection of two microtubules. On the bright side, due to the nature of fluorescence microscopy, points of intersection of microtubules glow twice as bright as any other point on a single microtubule. Thus, intensity information has to be preserved to isolate the problematic points and tread carefully around them while performing the tracing.

3. THEORY

Deblurring. The input image is noise free, as are the smaller images. However, they still need to be deconvolved with the PSF so that the deconvolved image is positive, and the noise is reduced, as the actual images will have noise. For such a system, the noise is usually Poisson in nature, so the Richardson-Lucy deconvolution algorithm can be used. It, of course, has to be modified to 3D, the implementation of which already exists in Matlab. **Direction of Extension.** However, after deconvolution, we are not guaranteed a thin image. That is, even after deconvolution, the thickness of

the individual filament can be more than 1 pixel. To get rid of this, we need to thin the image.

While thinning as well as tracing filaments, it is essential to know the direction that the filament at that point has grown from, and the direction it is growing in. In two dimensions, the Hessian is often used to determine this. This concept can be extended to n-dimensional space very easily.

The Hessian is a matrix of second order derivatives. If $I(x, y)$ is a 2D image, the Hessian H is given as

$$H = \begin{pmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} \end{pmatrix} \quad (1)$$

For an n-dimensional image $I(x_1, x_2, \ldots, x_n)$, it is

$$H = \begin{pmatrix} \frac{\partial^2 I}{\partial x_1^2} & \frac{\partial^2 I}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 I}{\partial x_1 \partial x_n} \\ \frac{\partial^2 I}{\partial x_2 \partial x_1} & \frac{\partial^2 I}{\partial x_2^2} & \cdots & \frac{\partial^2 I}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 I}{\partial x_n \partial x_1} & \frac{\partial^2 I}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 I}{\partial x_n^2} \end{pmatrix} \quad (2)$$

For a continuous, well-behaved image (as is the case always), the second derivatives have the property that $\frac{\partial^2 I}{\partial x_i \partial x_j} = \frac{\partial^2 I}{\partial x_j \partial x_i}$. Thus the Hessian is symmetric. The eigensystem of the Hessian of the image at a particular point gives information about the direction of the tubule at that point. The direction of extension of the filament is given by the eigenvector of the Hessian corresponding to the eigenvalue of the minimum magnitude. As the eigenvalues are all real, the singular value decomposition can be used instead, as it inherently orders the (positive) singular values in the order of decreasing magnitude. Thus we only need to pick the last eigenvector (and its negative) to find the directions of extension of the filament at that point.

For discrete images as obtained from a computer, the Hessian is replaced by its finite-difference equivalent. Since the image is now not continuous, to preserve the symmetry of the Hessian (for real eigenvalues), the finite difference implemented assumes $\left.\frac{\partial I}{\partial x_i}\right|_{x_1=X_1, x_2=X_2, \ldots, x_n=X_n} = \frac{I(x_1, x_2, \ldots, x_i+1, \ldots, x_n) - I(x_1, x_2, \ldots, x_i-1, \ldots, x_n)}{2}$. Using this definition, the property of the partial derivatives is preserved, and the Hessian remains symmetric. **Thinning.** The thinning of the image is achieved by a method called non-maximal separation, best known as one of the steps in the Canny edge detection algorithm. This method basically checks if a point is a local maximum along directions perpendicular to the direction of extension, and puts it to zero if it isn't. This results in thinning of the filament. The first step in this algorithm is to quantize the (bidirectional) direction vector to one of 13 possible (bidirectional) vectors which can be constructed in a 3×3 pixel cube. This is done by taking the inner product of the direction vector with each of the 13 vectors, and taking the one which results the maximum absolute value of the inner product. The inner product of this quantized vector with all the 13 vectors is again taken, and the 2 directions which are perpendicular to it are taken (they give a value of 0 for the inner product). The 4 pixels in these directions are taken, and the current pixel value is checked against each of them. If it is the maximum, its value is preserved. If it is not the maximum, the pixel value is set to zero.

This thinning also has the ability to function on the original image giving equivalent results. However this is true in the simulated image because of absence of noise. In the presense of noise, the deconvolution will indeed have to be carrued out.

Non-maximal supression is the slowest of all the steps, as it has to go through all the pixels of the image. **Thresholding.** It can be noticed that the intensity values are preserved along the filament length rather than set to a predetermined value. This is done to ensure that the intensity data along the length of the filament can be used in future to detect intersections. However, the intensity still has to be quantized.

It is a fair assumption that atmost two microtubules can intersect at a point. The probability of three microtubules meeting at one point is considerably low, enough to disregard it completely, especially near the periphery. Thus the intensity value of a pixel can be 0 (for a background pixel), 1 (for a pixel on the filament) or 2 (for the pixel at the intersection of two filaments). The deconvolution operation, however does not result in values which are close to these values. A thresholding step has to be implemented.

Hysteresis thresholding, another step from the Canny algorithm, is used as the basic idea here. For edge detection, where we want the output to be either 0 or 1, 2 thresholding levels are set. Intensities lower than the lower level are set to zero, those higher than the higher level are set to 1. The pixels which fall between the two thresholds are set to 1, if atleast one of their 8-neighbors is a definite 1. This hysteresis also tries to complete broken edges.

In this case, we need 3 different intensity levels, and correspondingly 4 different threshold levels — anything below level 1 is 0, anything above level 4 is 2, anything between levels 2 and 3 is 1. Anything between levels 1 and 2 is 1 if one of its *26*-neighbors is above level 2, 0 otherwise. As we are expecting only a few pixels with intensity 2, surrounded by atleast 2 pixels of intensity 1, anything between levels 3 and 4 is 2 if atleast 2 of its 26-neighbors are 1. The actual threshold parameters are fixed by trial and error.

4. RESULTS

4.1. Single Filament

The original image, Figure 1, is a curve in xyz space. It is convolved with the PSF to get the input image, Figure 2. The input image is put through Richardson-Lucy deconvolution to get the deconvolved image Figure 3. This step is just performed to check the performance of the deconvolution, because since the image is noise-free, nonmaximal separation can be applied directly on the input image to get the thinned image, Figure 4. Hysteresis Thresholding is then applied to the thinned image to get the final image, Figure 5. It can be seen that the image reconstruction is fairly accurate after all the steps have been applied. The pixel count in the final image is quite close to that in the original image.

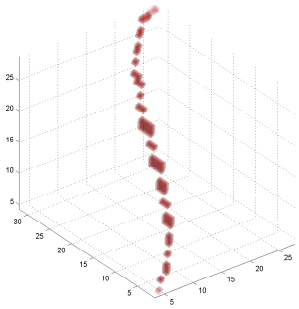

Fig. 1. Single Filament — Original Image

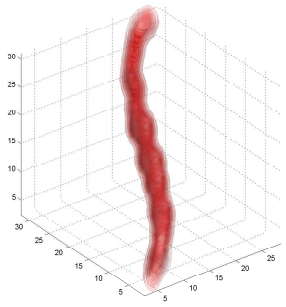

Fig. 2. Single Filament — Input Image

4.2. Intersecting Filaments

The original image, Figure 6, consists of two intersecting curves in the xyz space. The points of intersection have an

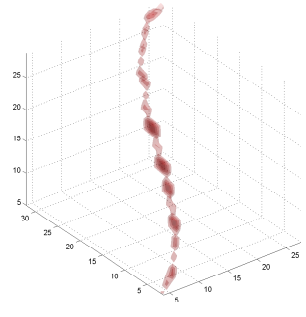

Fig. 3. Single Filament — Deconvolved Image

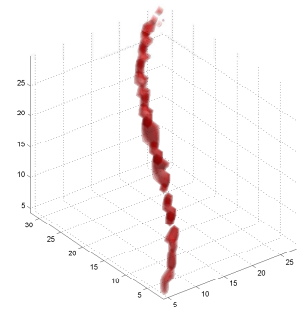

Fig. 4. Single Filament — Thinned Image

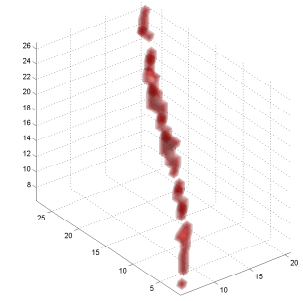

Fig. 5. Single Filament — Thresholded Image

intensity value of two. Similar steps as the single filament case are applied on to the input image Figure 7 to generate the final image, Figure 10. The intermediate steps are shown in Figures 8 and 9. The output matches very well to the input image, apart from slight deviations. The characteristics which was required to be preserved, the pixels with intensity value 2, match perfectly.

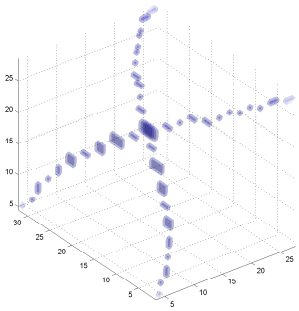

Fig. 6. Intersecting Filaments — Original Image

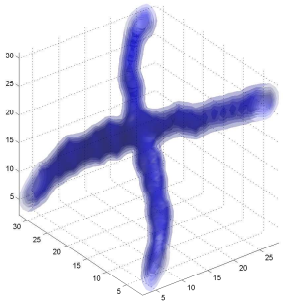

Fig. 7. Intersecting Filaments — Input Image

4.3. Simulation Image

The simulation output is shown in figure 11. The Richardson-Lucy deconvolution results in a clean image, Figure 12. Non-maximal supression gives the output of Figure 13, which is not as good as the ones from simpler images, but when used with the deconvolved image, can give good results. The output is a very clean image, in which tracing of the filaments should be an easy task. The procedure proposed to accomplish this is described later.

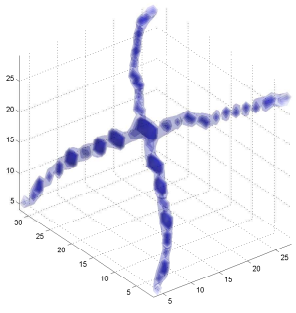

Fig. 8. Intersecting Filaments — Deconvolved Image

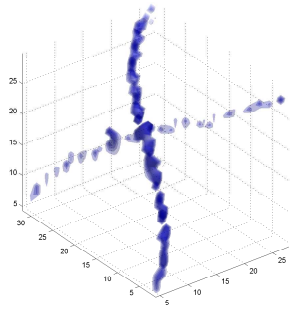

Fig. 9. Intersecting Filaments — Thinned Image

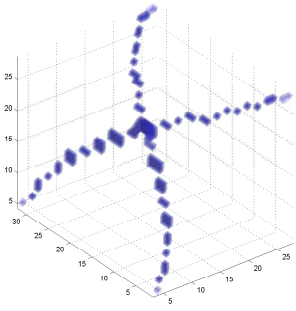

Fig. 10. Intersecting Filaments — Thresholded Image

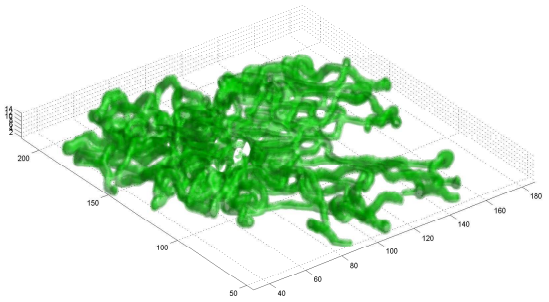

Fig. 11. Simulation — Input Image

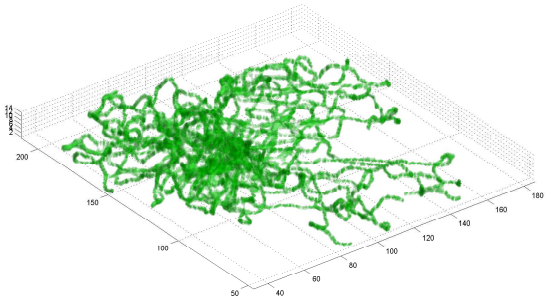

Fig. 12. Simulation — Deconvolved Image

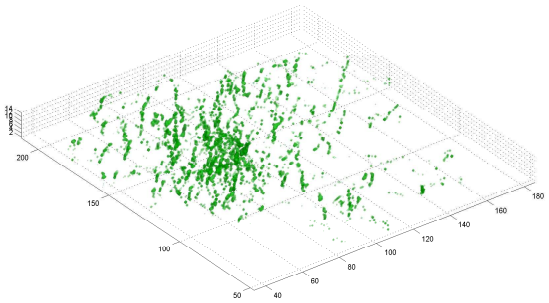

Fig. 13. Simulation — Thinned Image

5. DISCUSSION

The preprocessing of the image appears to be very successful in all three cases. The non-maximal suppression works well on input images without noise, but the results are not so good with deconvolved images. However, since the deconvolution step is essential to remove noise, an additional step of convolving the deconvolved image with the same or different PSF should give an noise-free blurred image, on which the non-maximal supression has been observed to work. This PSF is slightly more elongated in the z-direction than the xy-plane, which will interfere with the eigenvectors, and thus the finding of the direction of extension. Thus another PSF, having the same spread in x-, y- and z- directions will be preferred.

6. FUTURE WORK — TRACING THE FILAMENTS

Tracing the filaments is but a short step after the preprocessing is complete. The key idea used is the same as that of connected components labeling. The algorithm for connected components labeling interates Equation 3 interatively.

$$X_{k+1} = (X_k \oplus B) \cap A. \tag{3}$$

where $\oplus$ denotes dilation, A is the original image, and B is the structuring element, usually the set of 4-neighbors or 8-neighbors in two dimensions. The initialization X_0 is an empty image except for one pixel of the current component. This image is dilated interatively, but the intersection with A ensures that only those pixels which are present in the original image are retained in the component.

To label the next component, the current component is subtracted from the original image, and the algorithm is run again.

It is easy to see the application of this algorithm to trace filaments. Apart from the structuring element being the set of 26-neighbors instead of 4- or 8-neighbors, the dilation operation has to be made conditional. More specifically, the image should be dilated only in the direction of extension of the filament, not isotropically. This will only be needed near the intersection of two filaments, and these areas are already pinpointed in the preprocessed image. Thus, the algorithm is a short extension of the current connected components labeling procedures.

The initial pixels can be found out by searching inwards from the periphery.

7. CONCLUSION

The preprocessing of the image is one of the most challenging aspects of automatization of this task. In this case, the algorithm after pre-processing is a short one, and easy to implement. Due to constraints on time, this procedure could

not be implemented. Future work will involve tracing the filaments and comparing them to the input image, and computing the statistics of the filaments. After implementing on simulated images, the algorithm can be tested on actual images obtained from fluorescence microscopy.

8. REFERENCES

[1] M. E. Sargin, A. Altinok, E. Kiris, S. C. Feinstein, L. Wilson, K. Rose, and B. S. Manjunath, "Tracing microtubules in live cell images," *Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on*, pp. 296–299, 12-15 April 2007.