STROMULE BRANCH TIP DETECTION BASED ON ACCURATE CELL IMAGE SEGMENTATION

Guoyu Lu[⋆]*, Li Ren*[†]*, Jeffrey Caplan*[†] *and Chandra Kambhamettu*[†]

[⋆]Rochester Institute of Technology, USA, [†]University of Delaware, USA

ABSTRACT

Based on the dynamic structure, we design a system that can perform accurate stromule image segmentation, branch tip detection and tracking automatically. We substitute the user constraints in active contour segmentation by spatial fuzzy c-means clustering for providing more precise segmentation result. Based on the segmented contour after smoothing, we create a surface normal based feature that can accurately detect the branch tips. We further combine normal information together with tip position coordinate to apply ICP to track the branch tips moving path.

I. INTRODUCTION

As stromules are stroma-filled tubules that connect the plastids and support the exchange of stromal components between plastids [1], stromules are important to learn the mechanism of plant cells. Plenty of efforts is put in learning the biological functions of stromules in recent years [2] [3]. Stromules are highly dynamic structures which result in special characteristics for the shape. First, stromules can grow from the plant cells which form the shape as a branch. Second, the shape of stromules can dynamically change from time to time. Third, the number of stromules existing on one cell is not fixed. There might be several stromules on one cell and the number may change with the time moving. Also, stromules can only be seen in microscopy cameras and the stromules may lie in different depth layers, for which the branch shape is not always clearly observed on one image at a certain depth. These characteristics make stromules segmentation, detection, and tracking differ from normal objects that mainly rely on the local features.

For image segmentation, superpixel [4] and Min-cut [5] segment the images into many small regions based on pixels' RGB value, which may segment too many regions and consume relatively long time. Deep learning is explored to segment complex images [6]. In 3D space, object shapes are also learned to segment the object boundary [7], [8], [9]. However, deep neural networks usually result in complex models and require a large number of training samples, which is not suitable for our task. We apply fuzzy c-means and active contour segmentation algorithm (Snake) [10] to segment the stromule microscopy images. Instead of traditionally users providing external constraints for active contour segmentation, we use the fuzzy c-means clustering result to initialize the external constraints, which reduces manual efforts and provides more accurate initialization. We also apply different depth layers microscopy images to realize the accurate stromule image segmentation. In order to accurately detect the stromule branch tip later, we smooth the boundary to avoid the misdetection of branches, which is achieved through computing a regression line on the points inside a window and project the center point onto the line.

A key component for stromule analysis is detecting the branch tips from which we can further calculate the stromule length. Many features have been designed to detect salient points, for examples Harris [11], Shi-Tomasi corner [12], FAST [13], SIFT [14], and SURF [15], which are also used for tracking [16]. For microscopy image processing, Ducroz et al. [17] et al. utilize spherical wavelets to model the cells and detect the thresholded points with high curvature variations as the branches' tips. Tsygankov et al. [18] connect the boundary points with the cell main body line and detect the boundary local maxima as the branch, which is sensitive to the distance threshold. However, those methods largely rely on the rich texture information. As our images contain few textures, we design our own surface normal feature to detect the stromule tips. Based on the detected tips and segmented cells, we perform morphological operations to obtain the complete stromule branches. For tracking, we apply ICP algorithm [19] to track the branch tip movement due to the few textures on our images. The distance measurement of ICP is the combination of the Euclidean distance of both the point position coordinate and the surface normal. We then can find out the stromule movement path.

II. STROMULE SEGMENTATION

II-A. Fuzzy C-means Guided Active Contour Segmentation

We apply fuzzy c-means (FCM) clustering which is an unsupervised learning technique with spatial information analysis. This algorithm classifies the components of the image by grouping similar data points in the feature space into clusters. The result can be referred as a rough segmentation where the majority foreground pixels can be separated from the background. Then, guided by the result from the fuzzy clustering, we apply the Snake method to refine the segmentation result.

Generally, the FCM algorithm assigns pixels to each category with fuzzy coefficients. Let $X = (x_1, x_2, ..., x_n)$ represents an image with spectral values (RGB) of n pixels to be clustered into C clusters. The goal of the algorithm is to optimize the following cost function iteratively:

$$Cost = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - v_j \right\|^2 \quad (1)$$

where u_{ij} is the fuzzy coefficient that represents the membership of pixel x_i in the j th cluster and v_j is the center of the j th cluster. The fuzzifier m could be any real number greater than 1, determining the cluster fuzziness. A larger m leads to fuzzier clusters. We set 5 as the m value. The coefficient can be regarded as the probability that the pixel i belongs to the cluster j. The coefficients and the cluster centers are updated iteratively by the following functions:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} (\frac{\|x_i - v_j\|}{\|x_i - v_k\|})^{\frac{2}{m-1}}} \quad (2)$$

$$v_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \quad (3)$$

The updating process will be terminated when the values of membership coefficients and cluster centers converge. To further consider the spatial information of the image, a spatial function is defined below:

$$h_{ij} = \sum_{k \in SW(x_i)} u_{ik} \quad (4)$$

$SW(x_i)$ is a square window in a spatial domain, $5 * 5$ window as the window size in our setting, centered on the pixel x_i. The idea behind Eq. 4 is that the value of the spatial function is large when most neighboring pixels belonging to the j th cluster and small when few neighboring pixels residing in the same cluster. Similar to the coefficient function, spatial function represents the probability of a pixel belonging to a cluster. Thus, after the initialization of the centers, in each iteration, the coefficient function combining both the original color based membership function and the spatial function is updated by the following equation:

$$u'_{ij} = \frac{u_{ij}^{p} h_{ij}^{q}}{\sum_{k=1}^{C} u_{kj}^{p} h_{kj}^{q}} \quad (5)$$

where p and q are parameters to control the importance of relativity over spectral and spatial functions. Eq. 4 and Eq. 5 capture the characteristic that the pixels closing to each other in the spatial domain always belong to the same cluster. In addition, the misclassified pixels from the noisy regions can easily be corrected.

II-B. Active Shape Contour Segmentation

In order to further segment the contour, we apply active contour segmentation on the detected foreground images, which is an energy-minimizing spline forced by both internal and external constraints. The active contour detection function [10] is defined by a set of points near the boundary, as well as the internal energy terms and external energy terms, shown as below:

$$E = \int_0^1 E_{int}(v_i) + E_{img}(v_i) + E_{con}(v_i) di \quad (6)$$

where v_i is the i th point in a point set. $E_{img}(v_i) + E_{con}(v_i)$ represents the external energy which is composed by the image forces E_{img} and the constraint forces E_{con} provided by users. The image forces contain a number of functions representing features extracted from the image, such as the lines, edges and corners. The internal spline can be further written as,

$$E_{int} = (\alpha(i)|v(i)|^2 + \beta(i)|v(i)|^2)/2 \quad (7)$$

where $\alpha(i)$ and $\beta(i)$ are pre-defined weights that control the sensitivity between the first-order terms (stretch) and the second-order terms (curvature) respectively. To optimize the energy functions of the contour detection, the gradient descent minimization method is applied to move the contour towards the direction of negative gradient in each iteration. In our method, we apply the rough segmentation result from fuzzy c-means clustering to substitute user defined energy functions. In fuzzy c-means clustering, the pixels with a coefficient of over 70% probability belonging to foreground compose the initial force E_{con}. Comparing with the original active shape contour segmentation, our method is automatic without pre-defined initial contour, which largely saves manual efforts. At the same time, initial constraints from cluster selection are more robust than initialization from gray threshold method. Because selecting clusters from the fuzzy clusters separate foreground with dynamic thresholds updated pixel by pixel which capture both spectral and spatial information.

II-C. Contour Smoothing and Segmentation Based On Image Stack

We smooth the boundary that can facilitate to detect the branch tips, avoiding confusion due to the coarse boundary effect. We realize boundary smoothing by projecting all the contour points onto the local regression line. Specifically, we apply a sliding window along the contour points. For each point, N neighboring points in the window are sampled on each side and a local regression line is estimated. Thus, $2N + 1$ is the number of total points contributing to the computation of the local regression line. We choose N equaling to 10 in our setting. Then the current point is projected onto this line. Applying this algorithm to all the points smooths the contour and in a way brings the points closer. The larger the number of points, the smoother the curve. When there are too many points in the smoothing window, some important features such as corners will be lost. To avoid this situation, we adopt the weighted orthogonal least squares fitting. For a better smoothing effect, we also assign Gaussian weight on the points within the window.

Usually, when framing microscopy images, a specific focal length can only capture a part of the cells clearly

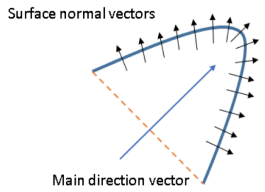


Fig. 1: The feature we use for stromule tip detection. We keep a window size 15 points along the boundary. For each point, a normal is calculated based on its adjacent points (small arrows). We also calculate the main direction for the entire window (the big arrow), which is perpendicular to the connection line of the starting point and the ending point (orange line). Each point's normal is projected to the main direction of the window, resulting in a projection distance along the main direction. A 15-dimensional vector with each dimension representing a projection distance is used as the feature of the window.

because the cell is located in different depth. In order to capture the entire cell clearly, we need to capture a series of microscopy images with different focal length. In our data, there are 8 depth layers, which we call z-stack. The segmentation result of each image in the stack is incomplete. To overcome this issue, for each image in z-stack, we apply the segmentation subroutine and project the segmentation results from different layers together onto one image.

III. STROMULE BRANCH TIP DETECTION AND TRACKING

III-A. Surface Normal Feature

We detect and count the branches by detecting their tips. To detect the tips, we propose our surface normal feature that describes the shape of the contour within a window. Specifically, we apply a sliding window to search along the contour. For each position, we apply the following extraction routine: (1) Find a main direction that is perpendicular to the line connecting the first point and the last point in the window; (2) Project every point's normal within the window to the main direction; (3) Record the length of the projection as a feature. If the length of the window is L, the feature has L dimensions.

Fig. 1 shows how the main direction is chosen and the feature is calculated. To discriminate the concave and convex shape, we further constrain the main direction facing towards the position where the tip point locates. To achieve this goal, we first compute the center of the first point and the last point in the window. The computed center point points to the middle boundary point in the window, which we call the "central line". We then calculate the angle between the pre-defined main direction and the "central line". If the angle is larger than $\pi/2$, we change the main direction oppositely.

III-B. Tip and Branch Detection and Stromule Tracking

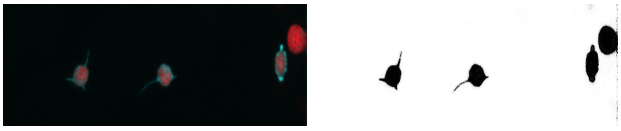
To detect the tip, we pre-defined the ideal feature. We use Gaussian distribution with pre-learned parameters as the ideal feature. For a new shape captured by the sliding window, we measure the distance between the feature extracted from the shape and our ideal feature. Fig. 6 shows an example. We can see that the distances on the tips' positions have significant low values compared with other non-tip places. Based on this characteristic, we can detect the branch by setting a threshold for the distance. Normally we set the threshold as 1.5.

From tip detection, we can get the number of branches each chloroplast cell contains. Meanwhile, the branch length is also an essential factor to analyze the stromule. To achieve this goal, we segment the branches out from the foreground cell images and then measure the length of each branch. To segment the branches, we first apply image opening on the foreground images, which removes the major body (non-branch area) of the cells. We then match the branch tip detection result to the image opening result, which further removes all the non-branch pixels left by the image opening operation. The leaving branch area can be used for measuring the branch size. To measure the length of branches, we apply Dijkstras algorithm to find the shortest path between the ends of the branches.

To understand the movement of stromules, we track the detected branch tips. From tracking, we can obtain the tip moving path. The tracking is conducted by ICP Mean Square Error (ICP MSE). For the distance measure in ICP, we combine the Euclidean distance of both points' position coordinates and their surface normals with equal weights. By combining these two distances, we utilize both the position and shape information for tracking.

IV. EXPERIMENT RESULTS

Our purpose is to design an automatic system that can segment, detect and track stromules accurately, in which there are not much research conducted specifically for stromule image analysis. We show the stromule segmentation, detection and tracking result separately in this section. First, fuzzy c-means segmentation result is shown in Fig. 2.



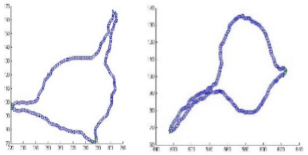
(a) (b)

Fig. 2: (a) Original microscopy image. Some cells contain stromule branches or begin to generate branches; (b) Fuzzy c-means segmentation. All the pixels are clustered into either the foreground or the background based on the pixel values. Cells together with the branches are mainly segmented out from the image. But there are discontinuities existing on the branches and noises on the background.

We can see from Fig. 2 that spatial fuzzy c-means can basically segment the cells out from the background. This result serves as the prior of active contour segmentation, whose result is presented in Fig. 3.

Fig. 3: Active contour detection with image closing. Pixels within the contour are marked as the cells (white) and pixels outside the contour are marked as background pixels (black). Then we perform image closing which completes the whole cell as an entire piece. Discontinuities on the branches are completed and connected with the main cell body.

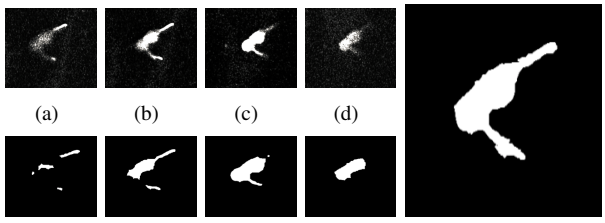
From Fig. 3, the stromule cell is accurately segmented from the background, proving the combination of spatial fuzzy c-means and active contour segmentation methods is suitable for extracting stromule images. In order to detect the branch tip accurately, we perform morphological operations (image closing) to fill in the discontinuities and then smooth the boundary. Fig. 4 shows extracted contours from the smoothed boundaries.



(a) (b)

Fig. 4: Boundary extraction and smoothing. Based on the segmented cells and branches, we extract the boundary of the cells by boundary detection and smooth the boundary to help us later detect the branch tip.

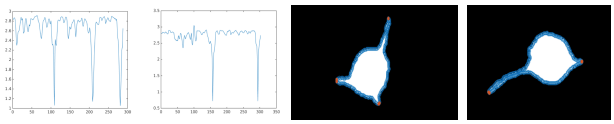
From Fig. 4, we can see that the stromule cell is complete with smooth boundary and branch tips can be easily observed. As a cell usually crosses different depth layers, we make use of images in z-stack to make the image segmentation complete. Fig. 5 shows examples of combining the segmentation result on z-stack images. From Fig. 5, we can see that though each image in different depth results in incomplete segmentation, the combination of all segmented images can provide an accurate segmentation output.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

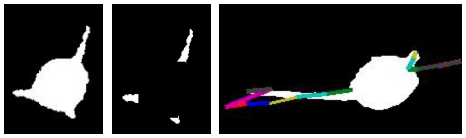
Fig. 5: Segmentation based on a video. (a-d): A series of cell images on z-stack; (e-h): Correspondingly, the segmentation result for each image in (a-d); (i) The combination of the segmentation result for a series of images.

We designed the surface normal features dedicated for tip detection. The distance between the features detected from each point on the boundary to the pre-learned Gaussian curve feature is presented in Fig. 6, which also exhibits the final tip detection result.



(a) (b) (c) (d)

Fig. 6: Tip detection based on surface normal features from sliding windows. (a) (b): The distance histogram of all the window features to the pre-learned Gaussian curve feature. The point in a window is considered as a tip if its surface normal feature distance to pre-learned Gaussian curve feature is smaller than a threshold; (c) (d): Tip detection result. Red points mark the detected tips corresponding to the feature distance histogram in (a) and (b).



(a) (b) (c)

Fig. 7: Stromule branch detection result. (a): Segmented cell; (b): Detected cell branches; (c): ICP MSE tracking result. The lines with different colors represent the tracking paths on different frames. This example contains the tracking result of 8 frames.

From Fig. 6, the stromule tip points are accurately detected by our surface normal feature. The tip points maintain a significant distinction from other boundary points with respect to feature distance. On top of tip detection, Fig. 7(a) and Fig. 7(b) show an example result of branches' segmentation, which demonstrates that our method can accurately detect and segment the stromule branches. With the detected tip, we can track the stromule tips and analyze the stromule behavior with the time moving. Fig. 7(c) shows the tracking result of a video sample.

V. CONCLUSION

In this paper, we present a system that can accurately segment, detect and track the stromules without human interference. In segmentation, the output of spatial fuzzy c-means clustering algorithm is used as the external constraint for active contour segmentation instead of user manual input, followed by smoothing through a regression line based method. We also create a surface normal based feature, which accurately detects the stromule branch tips. We also use image stack to complete the segmentation result. With morphological operations, we segment the branches out. For tracking the stromule, We apply ICP whose measurement metric is composed by both the point's position coordinate and the normal associated with the point. The experiments demonstrate that our system can accurately conduct the stromule segmentation, detection, and tracking tasks.

VI. REFERENCES

[1] J. Mathur, K. A. Barton, and M. H. Schattat, "Fluorescent protein flow within stromules," *The Plant Cell*, vol. 25, no. 8, pp. 2771–2772, 2013.

[2] M. R. Hanson and A. Sattarzadeh, "Trafficking of proteins through plastid stromules," *The Plant Cell*, vol. 25, no. 8, pp. 2774–2782, 2013.

[3] J. L. Caplan, A. S. Kumar, E. Park, M. S. Padmanabhan, K. Hoban, S. Modla, K. Czymmek, and S. P. Dinesh-Kumar, "Chloroplast stromules function during innate immunity," *Developmental cell*, vol. 34, no. 1, pp. 45–57, 2015.

[4] G. Mori, X. Ren, A. A. Efros, and J. Malik, "Recovering human body configurations: combining segmentation and recognition." in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2004, pp. 326–333.

[5] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3241–3248.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[7] G. Lu, L. Ren, A. Kolagunda, and C. Kambhamettu, "Neural network shape: Organ shape representation with radial basis function neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 932–936.

[8] G. Lu, L. Ren, A. Kolagunda, X. Wang, I. B. Turkbey, P. L. Choyke, and C. Kambhamettu, "Representing 3d shapes based on implicit surface functions learned from rbf neural networks," *Journal of Visual Communication and Image Representation (JVCIP)*, vol. 40, no. PB, pp. 852–860, 2016.

[9] A. Kolagunda, G. Lu, and C. Kambhamettu, "Hybrid hierarchical shape representation for medical shapes," in *British Machine Vision Conference (BMVC)*.

[10] S. Lankton and A. Tannenbaum, "Localizing region-based active contours," *IEEE Transactions on Image Processing (TIP)*, vol. 17, no. 11, pp. 2029–2039, 2008.

[11] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey Vision Conference*, vol. 15, 1988, p. 50.

[12] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994, pp. 593–600.

[13] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision (ECCV)*, 2006, pp. 430–443.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[15] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.

[16] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision." in *Proceedings of Imaging Understanding Workshop*, 1981.

[17] C. Ducroz, J.-C. Olivo-Marin, and A. Dufour, "Automatic detection of 3d cell protrusions using spherical wavelets," in *IEEE International Conference on Image Processing (ICIP)*, 2013, pp. 3499–3502.

[18] D. Tsygankov, C. G. Bilancia, E. A. Vitriol, K. M. Hahn, M. Peifer, and T. C. Elston, "Cellgeo: a computational platform for the analysis of shape changes in cells with complex geometries," *Journal of Cell Biology*, vol. 204, no. 3, pp. 443–460, 2014.

[19] F. Pomerleau, F. Colas, and R. Siegwart, "A review of point cloud registration algorithms for mobile robotics," *Foundations and Trends in Robotics*, vol. 4, no. 1, pp. 1–104, 2015.