Segmentation and Machine Learning for the Analysis of Bacterial Biofilm Images

---

A

Dissertation

Presented to
the faculty of the School of Engineering and Applied Science
University of Virginia

---

in partial fulfillment
of the requirements for the degree

Doctor of Philosophy

by

Jie Wang

December 2021

APPROVAL SHEET

This

Dissertation

is submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Author: Jie Wang

This Dissertation has been read and approved by the examing committee:

Advisor: Scott T. Acton

Advisor:

Committee Member: Zongli Lin

Committee Member: Tom Fletcher

Committee Member: Miaomiao Zhang

Committee Member: Andreas Gahlmann

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

December 2021

Segmentation and Machine Learning for the Analysis of Bacterial Biofilm Images

Jie Wang

a dissertation
presented to the faculty
of University of Virginia
in candidacy for the degree
of Doctor of Philosophy

recommended for acceptance
by the Department of
Charles L. Brown Electrical and Computer Engineering

Advisor: Scott T. Acton

December 2021

Abstract

As the first forms of life on earth billions of years ago, bacteria are essential participants in nature and exist almost everywhere. However, the living world of these cells remains mysterious due to limitations of conventional microscopy.

The recently introduced lattice light sheet microscope is capable of breaking the diffraction limit and performing long term live cell imaging with low photo-toxicity. This new modality necessitates advanced image analysis algorithms for understanding or ultimately controlling the activities of individual bacteria in a crowded, three-dimensional environment such as a biofilm. Bacterial behavior in biofilms is closely related to important problems in energy, disease, and the basic biology. Although there exist bacterial image analysis algorithms, they fail to delineate cells in dense biofilms, especially in 3D imaging scenarios in which the cells are growing and subdividing in a complex manner. This research develops an automated and effective analysis toolkit that overcomes the challenges of live cell imaging, such as the lack of apparent structure, limited contrast between conterminous cells, and high density of cells.

In the initial thrust of this research, a level set segmentation workflow, named Bact-3D, was explored with local constraints to stop the merging of level sets in different cell regions. The algorithm yielded promising results for multi-layered biofilm 3D data, but the performance degrades when applied to more complex biofilm images, where the gaps between neighboring cells are extremely hard to distinguish. Therefore, linear graph cuts (LCuts) is proposed as an automated cell segmentation algorithm to find individual bacteria by extracting and detecting the embedded linearity features in the biofilm. LCuts and its later improvements are generally extendable as a linear data clustering method; the method does not require prior knowledge of the number of cells as do other clustering methods. In the second major thrust of this dissertation, we investigate the incorporation of LCuts with deep neural networks to maximize the cell detection accuracy, and propose a generalized and unified algorithm, m-LCuts, for post-processing under- and over-segmented results.

Current training data are limited due to the limited number of real datasets and lack of corresponding annotated ground truth. The current gold-standard, manual annotation, is error-prone and time consuming. Therefore, as efforts to expand the annotated 3D biofilm datasets, both a model-based image simulation pipeline using optical and biological knowledge and an image generation workflow via 3D cyclic generative adversarial networks are introduced in the third thrust of this research. In order to evaluate those 3D synthetic datasets, a stochastic synthetic dataset quality assessment measure, named SSQA, is proposed that can fill the existing gap in the art to evaluate 3D synthetic dataset quality.

The automated algorithms, presented in this research, are able to promote the single-cell and population-level studies by combining super-resolution imaging with computational image advance. Furthermore, they also enable reconstruction and analysis of 3D biofilms. As conducted in the last thrust in this dissertation, biofilm reconstruction via geometrical model fitting and shape analysis of segmentation results can provide the cell biologists with statistics to further explore bacterial cell morphology and intrabiofilm mechanisms in future work.

Key words: 3D segmentation, data clustering, biofilm reconstruction, machine learning, graph, shape

Contents

Listing of figures

List of Tables

To my beloved parents,
my inspiration of happiness,
my strongest supporters,
for their unconditional love, caring, guidance and understanding.

Acknowledgments

A decade has passed, since I first entered the gate of Electrical and Computer Engineering as a fresh high school graduate. I still remember that I had no idea about how far I would go with the choice, but now I am presenting you my dissertation for the degree of Doctor of Philosophy in Electrical Engineering. This moment would never be possible if I did not have the support from the best advisor, wonderful mentors, sweetest friends, and beloved family along the path. A million thanks to all of you.

The first person, I would like to thank, is my advisor, Dr. Scott Acton. He led me to see the beauty of image processing, and provided me generous support, continuous encouragement and precious guidance. I learnt knowledge not only in academics from him, but also in the theory of life. I am very lucky for being in the VIVA lab under the protection of him.

I would like to thank Dr. Andreas Gahlmann and his lab in the Department of Chemistry. I really could not have done this research without all of their help. I am very grateful to have Dr. Gahlmann's advice to explore the image analysis potentials in bacterial biofilms. I am also thankful to work with the nicest collaborators, Dr. Mingxing Zhang, Ji Zhang, and Yibo Wang, for their generous sharing of data and knowledge.

Thank you to all my committee members, Dr. Zongli Lin, Dr. Tom Fletcher, and Dr. Miaomiao Zhang. I really appreciate every help and suggestion they offered me. They are always so kind, encouraging, and helpful. I am very honored to have them on my committee that I can share my work with.

The best VIVA, I owe you all so many thanks. I gained a lot of knowledge and priceless friendship from everyone in the VIVA lab. I will forever treasure the good old times with Dr. Andrea Vaccari, Dr. Rituparna Sarkar, Dr. Tamal Batabyal, Dr. Nazia Tabassum, Dr. Narsin Sadeghzadehyazdi, Dr. Tiffany Ly, and Haoyi Ma. I would also like to thank and welcome Tanjin Toma for her help and join in the biofilm research. Particular appreciation to Nazia, Tiffany and Nasrin, who really made my VIVA years bright.

I would send my gratitude to the VITAL lab, Dr. Zhang's lab, my roommates, and everyone I met at UVA and Charlottesville. It is my great pleasure to have the friendship with all of them. I will never forget the help and kindness I received. My UVA story is invaluable because of them.

My sincere gratitude also goes to all my bestests from back in China, for their continuous accompanying and support. Long live our friendship.

Finally, my deep appreciation goes to my dearest family, especially to my parents, for their unconditional love, caring, and understanding. They are my energy to keep positive and strong-minded. This dissertation is for you.

The real voyage of discovery consists not in seeking new landscapes, but in having new eyes.

Marcel Proust

1

Introduction

As the first forms of life on earth since billions of years ago, microorganisms are essential participants in nature and exist almost everywhere[30]. Based on the "two-empire system" since a century ago, cellular life is classified into two groups: Prokaryota and Eukaryota[16]. Prokaryotes account for more than half the biomass on earth with an estimated count of 5 nonillion[128], which is about 6.5×10^{20} the population of humans on the earth. Bacteria and archaea are all prokaryotic cells in the microscopic world. The survival of almost all the animals on the earth is dependent on bacteria and some archaea, because they provide fundamental genes and enzymes for vitamin B_{12} synthesis[23].

Analysis of individual microscopic cells, especially those bacterial cells which aggregate in multicellular communities called biofilms[82], promotes significant insights regarding microorganisms. Studies indicated that single-cell level inter- and intra-cellular variations and activities are critical for biofilm development[85,82], cellular functionality[87,3], and population heterogeneity[111,138]. Understanding these single-cell level activities is an essential step for preventing harmful impacts and harnessing beneficial outputs of microorganisms for human benefit. For example, the formation of bacterial biofilms on microbial surfaces can cause persistent infections with higher antibiotic tolerance in contrast to single wandering cells outside biofilms[22,131]. Thus, learning the inner workings of biofilms can assist in finding effective treatments for bacterial infectious diseases, such as cystic fibrosis pneumonia, cholera, and chronic implant-related infections[47]. A promising industrial application of bacteria involves wastewater that could produce electrical power or high value chemicals through electro-biosynthesis[71].

Studying single cell behavior in biofilms requires the identification of individual cells from the complex 3D environment, which remains a challenging task due to limitations in both imaging and image analysis techniques. In terms of imaging biofilms, phototoxicity, the optical diffraction limit (around 200nm), and anisotropic imaging resolutions (inferior vertical resolution), make accurate segmentation of individual cells difficult. In terms of image analysis, the main issues are caused by the dense aggregation of micron-sized cells, nonuniform intra-cellular fluorophore distributions and spatially variant illumination conditions. In this dissertation, an automated biofilm image analysis approach is presented to perform single cell analysis for images acquired by the advanced lattice light sheet imaging technique.

1.1 LATTICE LIGHT SHEET IMAGING OF BACTERIAL BIOFILMS

The imaging of bacterial biofilms has been a challenging task due to the lack of imaging techniques to resolve cellular behaviors over time among many micrometer-sized bacteria in 3D biofilms[82]. Conventional optical microscopy techniques, such as confocal and wide-field microscopy, suffered from the diffraction limit (200 nm) and high phototoxicity caused by high illumination intensity. For example, the resolution of images produced by these methods can only reach about 230 nm in the lateral plane and get worse in the axial direction with only 570 nm[137]. In these cases, bacterial cells with a typical radius ranging from 200 nm to 600 nm can be easily fused to the neighboring cells with diffraction limited microscopy. Meanwhile, live cell functions can be destroyed with high illumination intensity. For confocal and wide-field microscopy, the illumination intensities are usually at 10^5 and 10^3 w/cm^2 respectively, which are potentially harmful for light sensitive specimens and long-term live cell imaging[18]. Rapid photobleaching of fluorophores is another factor that affects the quality of images. In confocal microscopy modalities, the high contrast images are produced by laser scanning across the image and removing out-of-focus signals by pin holes. The out-of-focus signals are usually repeatedly illuminated when scanning across the whole image, resulting in fast photobleaching[137].

To overcome these limitations, 3D super-resolution imaging technology[18,40] is applied by using a lattice light sheet to excite fluorescence of a specimen. Lattice light sheet microscopy (LLSM) is capable of performing long term live cell imaging with non-phototoxic illumination intensities ($\sim$1 w/cm^2), which is comparable to the illumination of sunshine ($\sim$0.1 w/cm^2). Meanwhile, the fluorophore photobleaching can be 20–50 times lower than confocal microscopy to produce comparable resolution and contrast[137,18,118]. With the structured illumination mode, LLSM can even produce three-dimensional images at resolutions ten times smaller than that provided by traditional techniques[18].

The LLSM for acquiring fluorescent three-dimensional images was built in the Cell Imaging at Nanoscale lab at the University of Virginia (UVA). A 2D optical latticed thin sheet of light is used to illuminate and excite the flourophores in the excitation planes in the specimen[137]. An illustration of the imaging procedure is shown in **Figure 1.1**. Because the laser and detector are set up at an inclined angle, the final images for image analysis are de-skewed. Additional raw data

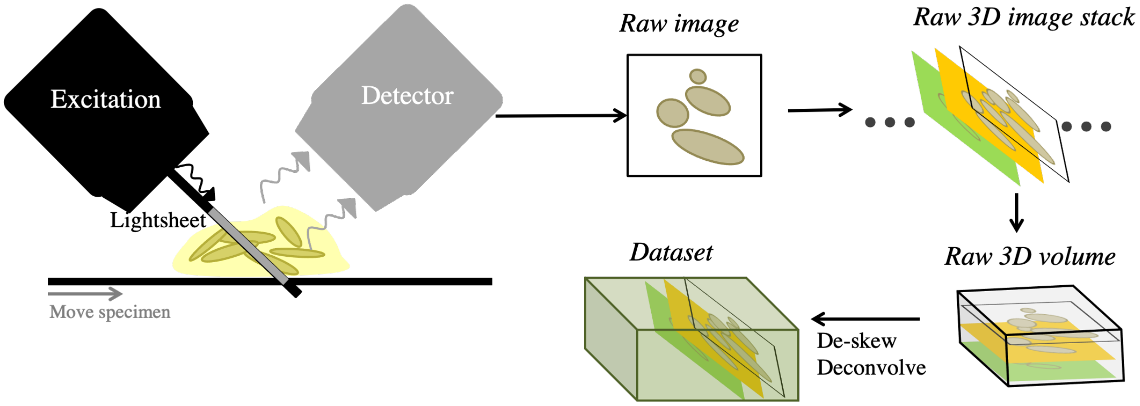


Figure 1.1: Illustration of the imaging procedure of bacterial biofilm using LLSM. The final output for image analysis are 3D images de-skewed and deconvolved from the raw volume produced by LLSM.

processing was also conducted in the imaging lab with background subtraction and deconvolution using the Richardson–Lucy algorithm. The optical resolution of LLSM outputs is 230 nm × 230 nm × 370 nm and the image voxel size for the final dataset is 100 nm × 100 nm × 100 nm.

1.2 DATASETS AND CHALLENGES IN THE IMAGE

There are three species of bacteria utilized in this dissertation: *Shewanella oneidensis* MR-1, *E. coli* K12, and *Myxococcus xanthus*. *Shewanella oneidensis* bacteria are commonly found in aquatic ecosystems. They are notable for their function as microbial fuel cells due to respiratory abilities[59]. Shewanella also has one of the densest biofilms; Shewanella is the model bacterium to study cell detection in dense biofilms (**Figure 1.2 a**). *E. coli* is a typical bacterium that normally lives in the intestines of people and animals, and serves as a sample model in the study of bacterial biofilms. Two different labeling methods, (1) genetic labeling through the expression of cell-internal fluorescent proteins with cytosolic fluorophores, and (2) staining of the cell membranes using fluorescent dyes, can be utlizied at the same time (**Figure 1.2 bc**). More challenges occur when cells are bending, growing, and subdividing. *Myxococcus xanthus* is a predator in the bac-

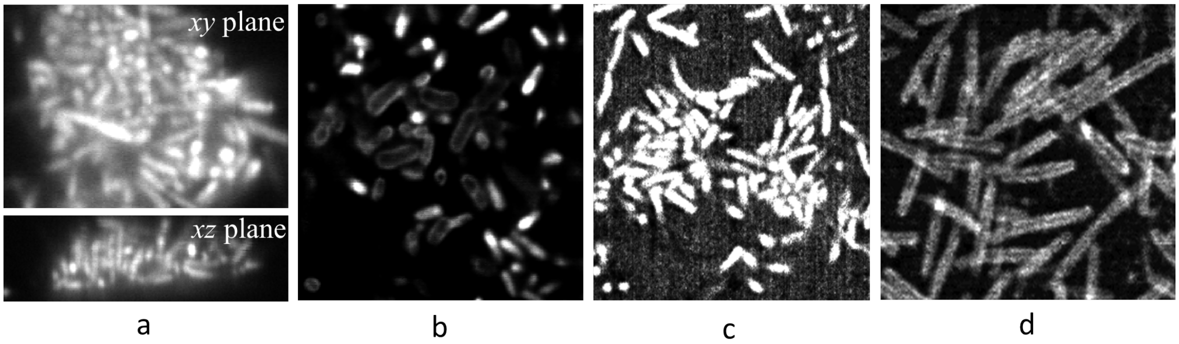


Figure 1.2: Sample experimental datasets covered in the dissertation. (a) Two viewpoints of Shewanella dataset. The others are *xy* views of (b) mix-labeling of E. coli dataset, (c) interior labeling of E. coli dataset, and (c) membrane staining of Myxococcus dataset.

teria family. One notable morphological behavior occurs when there are not enough nutrients in the biofilm; in such a case, some of the cells will turn quickly into spheres, the so-called *fruity body*.

As it can be observed in the datasets (**Figure 1.2**), the major challenges in the images for single cell analysis come from: (a) the presence of uneven fluorescence intensity within a single cell or across multiple cells; (b) the mixture labeling of cytosolic expression of cell interior fluorescent proteins and membrane staining with fluorescent dyes; (c) the diversity in cell shapes; (d) the high density of cells that are closely touching each other; (e) the complex noise conditions in the dataset, especially the ones imaged at later time points in biofilm formation.

1.3 Overview of image analysis techniques

The ultimate goal of image analysis is to extract information that is meaningful for the target dataset[104]. For single cell analysis of dense bacterial biofilms in this dissertation, informatics, such as cell size, cell shapes, cell-to-cell spatial and temporal relationships, are of great interest for biofilm researchers, where identification of individual cell regions is the essential first step.

There are generally two ways to find these individual cell regions: object detection and image segmentation. Object detection utilizes features in the image to localize each individual cell, usually by a coarse bounding box. Such features may include edges, intensity, texture, and curvature. For the cases when the maximum intensity is always inside of the cells, a local maxima intensity detection approach can satisfy the detection of cells [102]. With spherical cells, the curvature of cell shape is used for neuron detection with Laplacian of Gaussian (LoG) [125]. For deep learning related applications, the region-based convolutional network (fast R-CNN [43]) and its variations are the current top picks with successful application in cell detection (Cell R-CNN V_3 [69], CPN [5]). For R-CNN, the final input includes a bounding box regressor to locate the object and a classification score to determine the confidence of the detected region to be the targeted cell. However, object detection can only provide an approximate size and location of cells, while the other important informatics are not accessible.

Image segmentation for single cell analysis is an instance segmentation task. For instance segmentation, in addition to separation of the foreground of cells from the background, the identification of edges between neighboring cells is a vital step [46]. The classic image processing techniques usually take a combination of thresholding and watershed methods [57,132,8,49]. The general thresholding methods attempt to find the cell regions by way of low-pass or high-pass filter, but they are usually unable to distinguish noisy background from inhomogeneous foreground. Watershed [117] uses the gradient flow to identify the morphological changes along segment contours, or variations in the image intensity. Such method is sensitive to the noisy conditions in the image. Edge-based image segmentation methods, such as edge detection [20], define the contours by utilizing the gradient and Hessian of the image. They are also error-prone to discontinuities in color, texture, etc. Region-based solutions are usually less sensitive to noisy signals in the image and can provide segmentation results with smooth boundaries, such as using geometric active contour models. Geometric active contour models are known for their ability to automatically adjust to the image topology in the presence of an arbitrary number of regions [4]. However, for single-cell segmentation purposes, the basic geometric active contour models can easily ignore the small gaps between two adjacent cells. Another model, parametric active contours, do not merge to another contour belonging to the other region, but the initialization of landmarks on the contours for each individual bacterium in the biofilm is required.

Within deep neural networks, convolutional neural network (CNN) based models are especially popular approaches in biomedical image segmentation tasks[77,33,91,129,69,36,137,116,21,110]. There are two typical routines for achieving single-cell instance segmentation: (1) preprocessing with CNN to distinguish foreground from background, then followed by post-processing to identify each individual cell[116,137,36,110]; (2) localization of each cell region by object detection, then finding the precise segmentation inside of each object region[135,69,5]. However, the most significant obstacle for implementing such an experiment is the sizable demand for training datasets. For experimental biofilm images, the ground truth cell positions and boundaries are unknown. While often used as a gold-standard, manual labeling of training datasets is not only time-consuming, but often inconsistent across different annotators and different datasets[137]. Data augmentation that can extend the feature dimension of limited datasets is thus highly adopted in state-of-the-art pipelines for biological image segmentation[91,33,137].

1.4 Dissertation overview

This dissertation aims to present automated algorithms to break through the inability in resolving and analyzing individual cells in densely-packed biofilms incorporating both classic and machine learning image analysis techniques. The goals are captured by the four thrusts described below. They will be introduced and analyzed in detail from Chapter 2 to Chapter 5 correspondingly[*]. In the last chapter, we will conclude, discuss the contributions, and point to potential future work.

---

[*] The details and figures of each chapter are related to the author's publications as cited in the thrusts.

1.4.1 Thrust 1: segmentation of bacteria with LCuts

The complicated structure of three-dimensional biofilms poses distinct challenges for quantitative cell analysis and modeling, especially with the presence of variant density, noise, illumination, cell type, and fluorescent staining conditions[137]. In this thrust, we are going to introduce two methodologies to overcome the hurdles presented by this biofilm application: (1) active contour-based segmentation method with ***Bact-3D***[122] for multi-layered biofilm data, and (2) a more advanced segmentation solution with linear feature-based detection algorithm ***LCuts***[120], which can find individual cells in a more complicated structure in 3D. *LCuts* will be explored using a recursive graph cut approach to enhance the single cell identification performance.

1.4.2 Thrust 2: generalization of the segmentation method with m-LCuts

To handle the image analysis challenges, many toolkits were proposed over the last decade to detect individual cells. However, a trade-off that exists among all the methods is between the goal of identifying distinct cells and the goal of preserving morphology. Favoring one such criterion may lead to frequent cases of under-segmentation and over-segmentation of cells. These phenomena are even more common for bacterial cells with anisotropic cell shapes. Post-processing methods are thus required by many existing solutions, involving either the combining of the over-segmented clusters or the splitting of the merged cells. Therefore, ***m-LCuts*** (masked LCuts)[124] will be presented as a generalized and unified solution to fix both over-segmentation and under-segmentation problems with one theory in Chapter 3. Experiments are conducted that combine computational and machine learning-based techniques for a generalization of segmentation for bacterial biofilms in ***BCM3D***[137] (with an intermediate version of m-LCuts, referred to as ***refined LCuts***).

1.4.3 Thrust 3: image synthesis and evaluation for bacterial biofilm images

The advent of deep learning brings opportunities for solving challenging problems in a variety of fields, including single-cell analysis. Data-driven approaches typically require a large quantity of labeled training data to achieve reliable solutions. To expand the number of 3D images to quantify and evaluate data-driven analysis, we present solutions for synthetic data generation by way of model-based image simulation (in Chapter 3) and generative adversarial networks (GANs) based image generation algorithms in Chapter 4. For model-based image simulation, datasets are generated combining computational model of biofilm growth and optical knowledge of the microscope. For GAN-based image generation, ***3D Cyclic GAN (ℓ_{21})*** with unbalanced cycle consistency loss is proposed in our research[123], with the purpose of mimicking the realistic conditions in experimental images. GAN-based approach can take into account the challenging image conditions, such as uncalibrated image aberrations and illumination/emission heterogeneity; while, model-based image simulation workflow cannot accommodate these situations. Additionally, a stochastic synthetic dataset quality assessment (***SSQA***) measure[123] will be presented that can evaluate the qualities of synthetic datasets without the need for corresponding ground truth images in 3D.

1.4.4 Thrust 4: bacterial biofilm reconstruction and analysis

The aforementioned goals that segment, learn and evaluate biofilm images can further enable advanced bioinformatics analysis in Chapter 5. Informatics embedded in biofilms enable understanding mechanisms inside of biofilms and the control of the development of harmful biofilms. Reconstruction alternatives will be introduced, including using ***geometrical model fitting*** and exploit active surface for ***model shape refinement***. Furthermore, we would like to demonstrate the potential of shape-based biofilm analysis to analyze, visualize, and compare bacterial shapes in the biofilm, based on our work ***shape filter***[121].

For the things we have to learn before we can do them, we learn by doing them.

Aristotle, The Nicomachean Ethics

2

Segmentation of Bacteria

The objective of this chapter is to introduce a single-cell segmentation solution, named LCuts (Linear graph cuts)[120], that overcomes the difficulty of segmenting dense aggregations in a large biofilm with non-homogeneous inter- and intra-cell intensities. In our initial attempt for single-cell segmentation, a level set segmentation method, Bact-3D[122], was introduced. Bact-3D prevents the level sets in different single-cell regions from merging into each other by establishing local evolution constraints. Although promising results for multi-layered biofilm 3D data was achieved, the performance degrades on more complex biofilm images as the gaps between neighboring cells are extremely hard to identify. In contrast, LCuts is capable of finding the linear structures in point cloud data where cell boundaries may be ambiguous. Instead of identifying the boundaries in intensity between densely packed cells, LCuts compute the local intensity maxima to delineate the central axis of each cell; then it partitions the point cloud data based on the approximate collinearity of bacterial cells embedded in the image to represent different groups of cells. LCuts is a graph-based multi-class linear data clustering algorithm that works in a recursive manner. It does not require prior knowledge of the number of cells as do other clustering methods.

2.1 Background

For single-cell segmentation tasks, the two major objectives are: (1) distinguishing the inhomogeneous foreground (bacteria) from a noisy background; (2) identifying the gaps between neighboring cells.

Thresholding-based methods ([98,86,90]) usually take a multilevel approach to deal with the inhomogeneity. In[98], the authors combined global and local adaptive thresholding methods to optimize segmented cells in iterations. Authors in[86] hierarchically found multiple thresholding levels followed by morphological operations to find nuclei regions. However, these methods can easily cause incomplete cell segment with rough boundaries or false detection of cells from the noisy background. Methods derived from geometric active contour models, *i.e.* level sets[17,4,80,63], are able to provide smooth segment for objects without the need for clear edges. Modeling with Legendre polynomials in[80] enabled the level sets segmentation with the presence of uneven intensities.[63] corrected the inhomogenity and noise in 3D directions of the image and performed region-based active contours segmentation. Both aforementioned thresholding and active contours methods face difficulty in identifying small gaps between closely touching cells. To analyze the behavior of an individual cell in a crowded collection of cells, the identification of edges between neighboring cells is another vital step. A popular choice is to use watershed algorithms[117] and its variations with seeds[11,132,8,49]. The latter, i.e. marker-controlled/seeded watershed algorithms, performed decent work in many cell segmentation problems[132,8,49]. A preprocessing step is usually taken to denoise and enhance the image quality. Then, seeds/cell-markers can be found as the regional maxima intensity within cell regions. With the watershed algorithm, these seeds will grow in the region of cells and stop at edge boundaries or merging points with other markers. Nevertheless, such methods are sensitive to images with heterogeneous fluorescence intensity across the image. The high density of cells also adds an additional challenge to these methods, in which case the gaps between neighboring cell are even harder to identify. Consequently, missing cells, over-segmentation, and under-segmentation problems are observed[49,41,132]. To decrease the cases of unsolved segmentation error, some alternative active contours methods were proposed[122,6,79]. [6] and[79] incorporated shape constraint for active contours model to segment overlapped cells. However, when it comes to 3D cell segmentation in much complex 3D structure in biofilms, additional challenges arise.

The structural heterogeneity of 3D biofilms, namely variant cell density, illumination intensity, cell type, and fluorescent staining, poses severe challenges for quantitative cell analysis[137]. Thus, modalities in the art tried to integrate different techniques together to identify each single-cell region. Seg-3D, proposed in[88], combines local thresholding, shape analysis, and concavity-based cluster splitting to identify single cells. The utilization of the concavity of the contours often causes incorrect splitting of the cells. The authors in[106] distinguish the cell regions from the background by way of a saliency cut that considers the color and size information of regions in the image. He et al.[51] detect single cell regions by analyzing features on cellular convex curvatures. Yet, these methods are not designed for the needs in the 3D segmentation tasks and can easily cause over- and under-segmentation.

To find an advanced solution that can resolve single bacterium in the dense 3D biofims, we investigated alternative active contour-based segmentation methods and feature-based data clustering methods, by introducing Bact-3D and LCuts.

2.1.1 Level set segmentation methods

Active contour-based segmentation methods can be categorized into parametric and geometric active contours[4]. Between these two alternatives, geometric active contours, *i.e.* level set methods, are more flexible in topological changes, which means it can segment different objects simultaneously. The general theory of level set segmentation was proposed by Osher and Sethain[84], in which a level set function $\varphi(x, y; t)$ evolves iteratively along time to approach the object boundary. For example, the level set function can be a signed Euclidean distance function. The zero level set, $C = \{(x, y; t) : \varphi(x, y; t) = 0\}$, represents the boundary contour, where the inside ($\varphi > 0$) denotes foreground and outside ($\varphi < 0$) is the background. The motion of the level set function can be generally formulated as:

$$\frac{\partial C}{\partial t} = V\mathbf{N} \tag{2.1}$$

where V is the velocity that evolves the curve and $\mathbf{N} = \nabla\varphi / |\nabla\varphi|$ is the outward normal force. Using the chain rule, the above general expression is equivalent to:

$$\frac{\partial \varphi}{\partial t} = -V|\nabla\varphi| \tag{2.2}$$

which can be further written into discrete time evolution as: $\varphi^{n+1} = \varphi^{n} - V|\nabla\varphi^{n}|\Delta t$.

In the level set literature, different velocity V functions have been proposed for different application needs. For example, Osher and Sethian defined edge-based curvature-dependent speed to propagate the zero level set to the object boundary. To find regions without clear edges, Chan and Vese[17] calculate regional mean values to lead the movement of level set function. To detect tubular structures of vessels, the authors in[81] analyzed eigenvalues of the Hessian of the images and incorporated curvature in the velocity functions to find vesselness structures in the image. The authors in[73] added a coupled tubularity and blob flow field to segment microgila cells. Legendre polynomials are utilized in the velocity function in[80] to capture inhomogeneity across the objects. Shape priors are added in[34] in the curvature-based velocity function to regularize the deformable model. More recently, the authors in[114] presented LyMPhi, which automatically refines the speed of the level set function towards the object foreground based on matted hierarchies in image intensity and location. Deep learning techniques were embedded in the region-based level sets formulation in[53], where deep convolutional neural networks were employed to learn the saliency of regions in the image to update the speed function.

2.1.2 FEATURE-BASED DATA CLUSTERING METHODS

Clustering is a method to group data with high degrees of similarity into the same set[83]. Major measures for similarity of data include, but is not limited to: distance, density, predefined models, and hierarchical relationships. K-means clustering[74] partitions the data into k groups which are centered at k arithmetic means. Gaussian mixture models[13] estimate the variables for multiple probability distributions in the point cloud by expectation-maximization. Both approaches can easily cause mis-classification when non-spherical morphology is encountered. To detect arbitrarily shaped clusters in the presence of noise, density-based measures have emerged. DBSCAN[37] and DensityClust[89] are widely used in many settings. Density-based measures find clusters with different size and shape based on local density of each point, while DensityClust also maximizes the relative distance between cluster centers. A common drawback of all of these approaches is that they require specification of the number of groups, which substantially affects the final clustering result. To automatically identify single cells in the presence of over- and under-segmentation errors, we propose to build a graph on the initially segmented results and then cluster the nodes in the graph based on their collinearity feature.

2.1.3 GRAPH CUTS

Graph cuts was originally a terminology in graph theory. A graph usually consists of vertices and edges, while a cut attempts to disconnect some edges to partition the graph into two disjoint sub-graphs. Based on the purpose of a cut, there are a variety of graph cut methods, such as minimum cut[130], maximum cut[28], and sparsest cut[76]. The first application of graph cuts theory in computer vision problem was in 1989 by Greig et al.[45] to evaluate how the maximum *a posteriori* (MAP) optimization problem in the images can be obtained by maximum flow. Since two decades ago, graph cuts for image segmentation emerged, such as normalized graph cuts[99] and interactive graph cuts[15]. As image segmentation is an effort to separate the regions that shared the similar features in the image, graph cuts-based segmentation methods also utilize these features to disconnect the edges among pixels or regions-based graphs. The graph construction and similarity definition are usually the key to applying graph cuts methods in image segmentation[14,99]. In normalized graph cuts, pixels are used as vertices in the graph while image intensity and location of regions in the image are used to define the cut energy[99]. In interactive graph cuts, the vertices for graph cuts are denoted as seeds, which may start from several bounding boxes of the regions of interest to construct the initial graph then iteratively add new seeds if needed[134,15]. Recently, graph cuts image segmentation pipelines are widely applied in biological and biomedical settings. The authors in[72] specified shape constraints in multi-dimensional feature-based min-cut to extract liver regions from CT images. Centroid-based and k-means based seed selection methods were proposed in[31] to construct graph on brain Magnetic Resonance Imaging (MRI) images. In[119], image intensity information and second-order tensor features, such as Hessian of the image, were used in max-flow min-cut workflow to segment vesselness structures in blood vessel. For the bacteria biofilm images in our experiment, graph cut-based methods provide us a direction by proposing biofilm graph construction and collinearity feature extraction to delineate cells even though they are closely touching to each other in 3D.

2.2 INITIAL WORK ON BIOFILM SEGMENTATION WITH BACT-3D

Bact-3D was firstly developed to resolve individual cells in dense 3D bacterial biofilms [122]. In order to identify and segment the individual bacteria in 3D, we propose an approach in which we first analyze each layer and finally reconstruct the segmented 3D volume. The intuition behind Bact-3D is that although bacteria are densely packed, their ellipsoidal shape, due to the interaction between the distribution of fluorophores and the simulated point spread function, helps in identifying a z-plane where all the individual cells are completely separated.

2.2.1 METHODS

Fig. 2.1 gives an overview of the methodology involved in performing slice-by-slice segmentation in a layered biofilm. In Bact-3D, we evolve the level set function from the seed points inside of each cell using Hessian-based curvature detection (**Fig. 2.1** b) and allow the contours, defined by the level set function, to evolve in a local neighborhood (**Fig. 2.1** c). The level set is evolved until it reaches a stopping criterion (**Fig. 2.1** e), obtained exploiting the prior slice segmentation result.

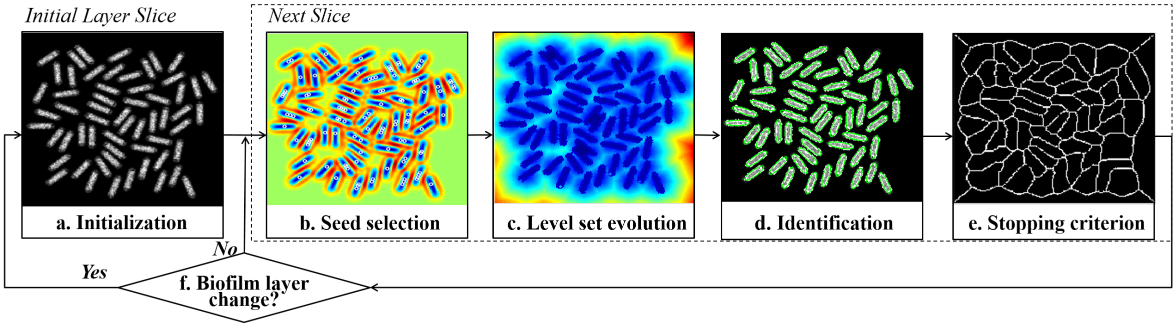


Figure 2.1: Flowchart of the Bact-3D algorithm.

The local level set method developed in Bact-3D is motivated by the idea of over-segmentation from [64]. To prevent the merging of evolving contours in slices where the boundaries of adjacent bacteria blur into each other due to the imaging resolution, it is critical to define a stopping criterion (where the curve evolution velocity is set to 0) that clearly separates adjacent cells. This is achieved by defining a piece-wise velocity function as follows:

$$V = \begin{cases} 0, & \text{if SC} = 1 \\ g[1 - \varepsilon\kappa] - \beta \nabla g \mathbf{N}, & \text{otherwise} \end{cases} \tag{2.3}$$

The $V = 0$ condition effectively prevents the contours, driven by the level set evolution, from crossing the boundaries defined by the stopping criterion (SC). The velocity model $V = g[1 - \varepsilon\kappa] - \beta \nabla g \mathbf{N}$[105] represents the propagating rate of the evolving curve toward the cell boundary. In this formula, κ is generally the curvature forcing a slow down at the boundary areas, ε and β are the contour smoothing and edge attracting regularization, and g is the local affinity function as defined in[64]:

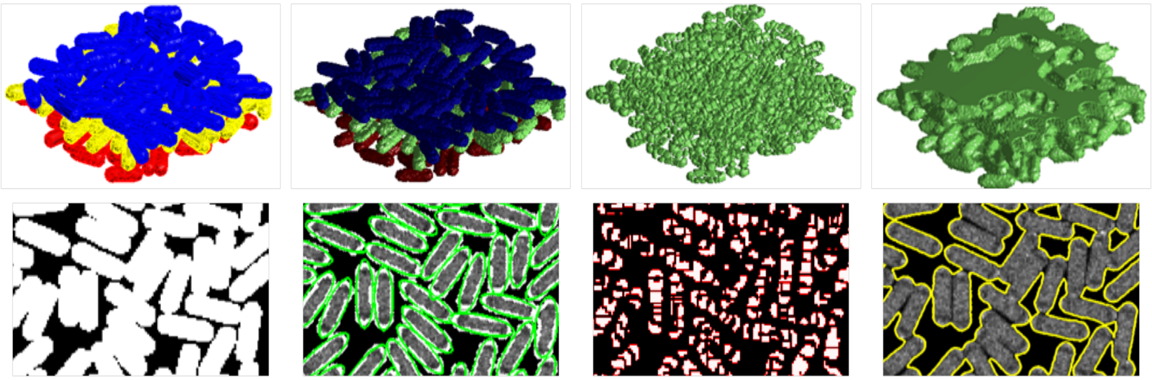
$$g(x,y) = e^{-E(x,y)/\nu},\ E(x,y) = \frac{|\nabla I|}{G_\sigma * |\nabla I| + \gamma} \tag{2.4}$$

where ∇I is the gradient of the image, G_σ is a smoothing kernel, and γ is a constant to guarantee that E will remain limited in presence of small gradients.

The stopping criterion (SC) in **Fig. 2.1 e** is the skeleton of background that extracts smoothed cell regions from the initialization layer of the biofilm, where cells are not touching to each other. The smoothed cell regions are cell segmentation results fitted with ellipse fitting, so that we can obtain a smooth skeleton of the background region. The initialization layers of the biofilm are automatically detected by analyzing the number of detected bacteria as function of slice location along the z-direction. At initialization layers, the numbers of bacteria are the local maxima.

2.2.2 DISCUSSION AND MOTIVATION FOR LCUTS

We qualitatively and quantitatively compared our results on simulated datasets with two published algorithms: *single cell tracking* by Yan, et al.[132] and *L2S*[80] in our Bact-3D paper[122]. **Fig. 2.2** is a visual comparison of all three algorithms. Yan's single cell tracking method is implemented using the available open source code where the value of 10 used in the watershed algorithm was empirically chosen to optimize the output. For L2S, we used the published code in a slice-by-slice implementation and then merged all the segmented slices along the z-direction to achieve the final 3D volume.



(a) Ground truth (b) Bact-3D (c) Yan et al.[132] (d) L2S[80]

Figure 2.2: Comparison of segmentation algorithms with Bact-3D. The first row shows final 3D volumes reconstructed by each algorithm (different color for each layer). The second row shows the segmentation on an individual slice.

The *Dice coefficient*, *mean squared error* and *cell detection accuracy* are used to evaluate the performance of the above methods (Table. 2.1). The Dice coefficient compares the similarity between two sets: the ground truth V_g and the detected V_t volume, and it is evaluated by $\frac{2|V_g \cap V_t|}{|V_g|+|V_t|}$, where $|\cdot|$ denotes the cardinality of the corresponding set, and varies between 0 to 1 with unity indicating optimal segmentation. Mean squared error ($\mathrm{MSE} = \|V_g - V_t\|_2^2$) measures the average squared errors between ground truth V_g and detected V_t volumes. The MSE is normalized by the total number of pixels in the volume. The cell detection accuracy (CD) determines how many cells are segmented and identified N_t compared with the actual number in the ground truth volume N_g. We define this measure similarly to the Dice coefficient: $\mathrm{CD} = \frac{2\min(N_g, N_t)}{N_g+N_t}$. The ground truth was defined, identically for all the resolutions, as the solid volumes occupying the simulated bacterial cells before the blurring by the Gaussian kernel. The value was set to unity for the inside of the cells and zero for the outside. The same index-type function was used for the detections

	Bact-3D			Yan, et al.[132]			L2S[80]		
Data	Dice	MSE	CD%	Dice	MSE	CD%	Dice	MSE	CD%
1	0.88	0.08	**100.0**	0.54	0.24	58.7	**0.89**	**0.07**	1.1
2	0.86	0.08	**100.0**	0.56	0.19	61.9	**0.88**	**0.07**	9.1
3	0.87	0.09	**100.0**	0.56	0.26	59.1	**0.89**	**0.08**	3.1
4	0.87	0.09	**99.5**	0.56	0.26	47.4	**0.89**	**0.07**	4.1
5	0.87	0.09	**99.5**	0.57	0.25	55.6	**0.89**	**0.08**	8.9

Table 2.1: Comparison of segmentation efficiency of Bact-3D. Each dataset was generated using the cylinders models to simulate the biofilm images with pixel resolution to be 77×77 nm[122].

obtained by the three methods allowing for a direct comparison.

Due to the high over-segmentation results, the single cell tracking method[132] has no evident advantages when compared with the two other methods. Although L2S[80] can define the homogeneous foreground from the background, when considering the accuracy in single cell identification (CD) in[122], the proposed Bact-3D shows a significant advantage with an average accuracy of 99.81%. Bact-3D output provides us not only with a clear 3D reconstruction of the layered biofilm but also with the ability to identify each individual bacterium (**Fig. 2.2**, second column) allowing the extraction of statistical information about location, size and orientation.

When Bact-3D was applied on more complex biofilm images as shown in **Fig. 1.2**, where the bacterial cells are not layerly-positioned and are more densely packed, the performance of Bact-3D degraded. Therefore, we explored a solution that can identify individual bacterial regions by linear data clustering using a recursive graph cuts method, named LCuts[120]. LCuts is built on the following insight: even though the raw image data does not show distinct boundaries in intensity between densely packed cells, we are still able to reliably compute local intensity maxima that delineate the central axis of each cell.

2.3 LINEAR CLUSTERING ALGORITHM

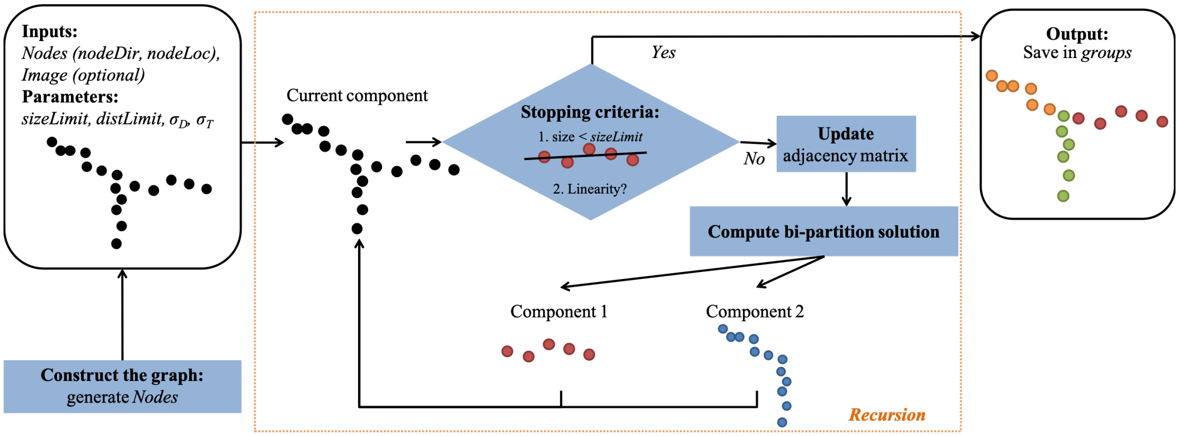


Figure 2.3: Intuitive workflow for the recursive program of LCuts. Here, $sizeLimit$, $distLimit$, and $eccLimit$ are parameters based on prior biological information. Blue panels are major steps proposed in Sec 2.3.1 to Sec 2.3.3.

Unlike most data clustering methods (such as k-means[74], DBSCAN[37], and DensityClust[89], LCuts does not require manual intervention in order to locate the appropriate number of clusters with a proposed recursive work flow (**Fig. 2.3**). Incorporation of structural constraints, such as the distance limit and the eccentricity of the bacteria into LCuts, obviates the need for *a priori* information regarding the number of clusters (in our case, the number of cells), making LCuts a fully automatic approach. The whole algorithm can be primarily divided into three parts: construct the graph (**Sec 2.3.1**), compute the bi-partition solution (**Sec 2.3.2**), and recursively re-partition until the stopping criteria are reached (**Sec 2.3.3**).

2.3.1 GRAPH CONSTRUCTION

Nodes: The nodes (local maxima along the ridgeline of a cell) in the constructed graph have two features: location (*nodeLoc*) and direction (*nodeDir*). Location is simply the Cartesian position of the node. Direction of each node is the principal axis direction of the ridgeline computed via majority voting (*see* **Fig. 2.4**).

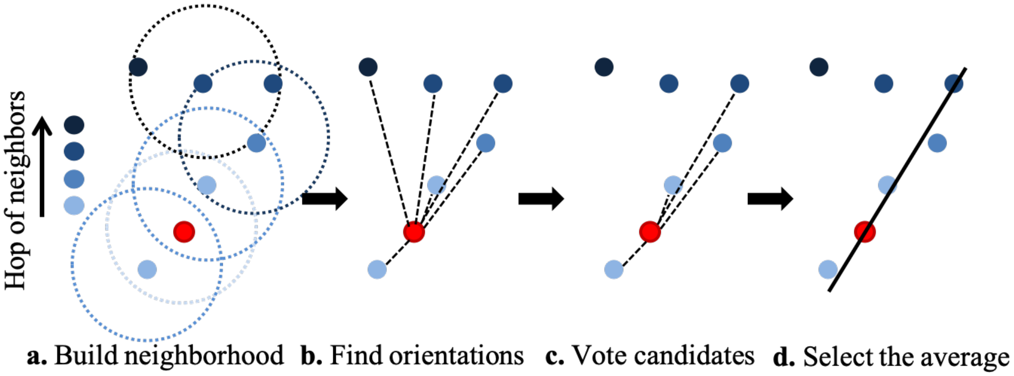


Figure 2.4: An illustration of majority voting. (a) A 4-hop "neighborhood" example. In graph theory, a "hop" between two nodes is defined as the number of edges that one has to traverse in order to reach from one node to the other node. Each hop-neighbor is found within a specified distance (dashed circles) to node. (b) The dashed lines connecting target node with all the other nodes in the neighborhood are possible orientations. (c) Those orientations that have larger relative angles with respect to the orientations are excluded from the candidates. (d) The direction to represent the target node is determined as the average orientation from the candidates.

A $N_r \times N_p$ accumulator is set up for the majority voting. One dimension of this accumulator represents the N_p possible orientations ($\mathbf{p}$) in N_p bins (see **Fig. 2.4b**). Another dimension corresponds to the quantized relative angles (φ) with N_r bins, where φ is computed from each possible orientation to all the others. Here, N_r is chosen based on the "hop" number. The accumulator will count the number of parameter pairs ($\mathbf{p}$, φ) that lie in each bin. Within the first bin of φ, the orientations with the largest value are selected which give the candidate directions. These candidates are averaged to yield the major direction for the target node.

Adjacency matrix: The adjacency matrix reflects the likelihood if two nodes are in the same group. Suppose there are N nodes in the graph, then the dimension of the adjacency matrix is $N \times N$. Each attribute in the matrix represents the connectivity and edge weight between two nodes (i, j): $w_{ij} = w_{distance} \cdot w_{direction} \cdot w_{intensity}$.

Three similarity measures are involved: the Euclidean distance of locations of nodes (**eq 3.3**), the relative angle between major directions (**eq 3.6**) and the dissimilarity of intensity along the segment connecting two nodes (**eq 2.7**). Here, the terms are defined as follows:

$$w_{distance} = \begin{cases} e^{-||\text{nodeLoc}_i - \text{nodeLoc}_j||_2^2/\sigma_D^2}, & \text{if } ||\text{nodeLoc}_i - \text{nodeLoc}_j||_2 \leq r \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

$$w_{direction} = e^{-(\cos(\theta)-1)^2/\sigma_T^2} \tag{2.6}$$

$$w_{intensity} = \min I_{i \to j}, \text{ if } \min I_{i \to j} \leq thresh \tag{2.7}$$

Here, $w_{distance}$ weighs the distance between two nodes and $w_{direction}$ weighs the difference between node directions. $||\text{nodeLoc}_i - \text{nodeLoc}_j||_2$ is the Euclidean distance between node i and node j, and r is set to eliminate edges between two far away nodes. θ_{ij} is the relative angle between the directions of nodes i and j. σ_D and σ_T are adjustable parameters that control the rate of exponential decay.

The third term, $w_{intensity}$, detects the intensity dissimilarity along the segment joining two nodes in the image. *thresh* equals the difference between the mid-range (*Mid*) of all the nodes and the variance (*Var*) of the constituent node intensities. In the case that the nodes have no intensity information, this term can be set as 1. Otherwise, we extract the intensities along the connecting segment from node i to node j from the image as shown in **Fig. 2.5** and find the lowest intensity along the segment and compare to *thresh*.

2.3.2 COMPUTATION OF BI-PARTITION SOLUTION

The computation of the bi-partition solution in the proposed recursive workflow was introduced by Shi and Malik[99]. The image is represented as a weighted, undirected graph $G = (V, E)$, where $V = A \cup B$ and E represents the connecting edges. Then, the image is partitioned into two groups, A and B, by disconnecting the edges between these two groups. A closed-form solution of this partitioning problem is found by rewriting *Ncut* into generalized eigenvalue system.

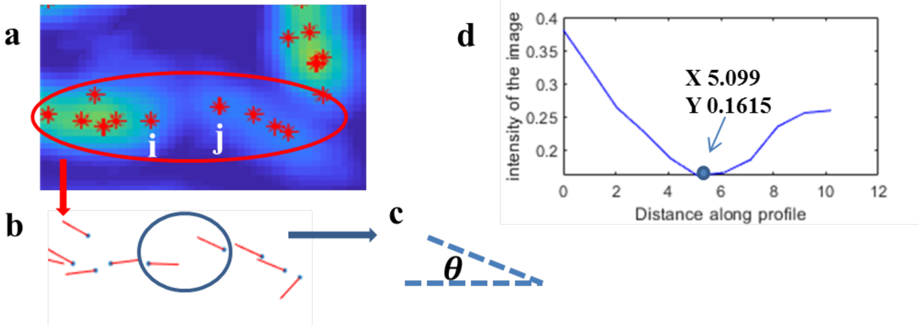


Figure 2.5: Illustration and motivation of defining intensity on edge weight. **a**: Nodes are denoted as red asterisk. **b**: After extracting the node directions from the red region in **a**, it is still hard to separate two groups as the relative angle (in **c**) and relative distance (shown in d, the distance is 10) are close. In this case, we evaluate the intensity along the connection of the two nodes. The intensity changes are shown in **d**. The intensity weighting is then assigned as the lowest intensity lower than *thresh*.

Then, the optimal bi-partition solution is given by computing the eigenvector with the second smallest eigenvalue of the generalized *Ncut* function. Please refer to **Section 3.4.7** for more details regarding the mathematical calculations.

2.3.3 STOPPING CRITERIA

Two stopping conditions are checked after each bi-partition step to decide the completeness of the recursion.

Criterion 1 - size: The preliminary components that are less than *sizeLimit* have the potential to be an individual group. Here, *sizeLimit* is determined by the prior bio-information of the maximum length of the bacterium.

Criterion 2 - linearity: Three aspects are checked to ensure the linearity: (1) *Standard deviation* (**Std**) from nodes in the group to the least square fitted line; (2) *Intensity changes* between the nodes within the group (as explained in **Sec 2.3.1**); (3) *Eccentricity* of the group. This is an optional condition based on the data type. For linear components, the eccentricity (*eccLimit*) is closer to 1; while, for circular components, it is closer to 0.

2.4 EXPERIMENTS ON 2D BACTERIAL IMAGES

2.4.1 DATASET

For qualitative and quantitative assessments, LCuts is first tested on 10 two-dimensional point cloud datasets which are generated from bacterial images using Airyscan microscopy. From these images, we obtain prior information regarding the longest bacterium in the dataset (approximately maximum 60 pixels in length and 15 pixels in width, where each pixel is 46nm × 46nm). The typical data have 250 to 600 nodes with approximately 20 to 60 cells observed. To build the graph, we generate the point cloud data following the pipeline in **Fig. 2.6**.

To build the graph, we generate the point cloud data following the pipeline in **Fig. 2.6**. An experimental result and corresponding node features is shown in **Fig. 2.7**.

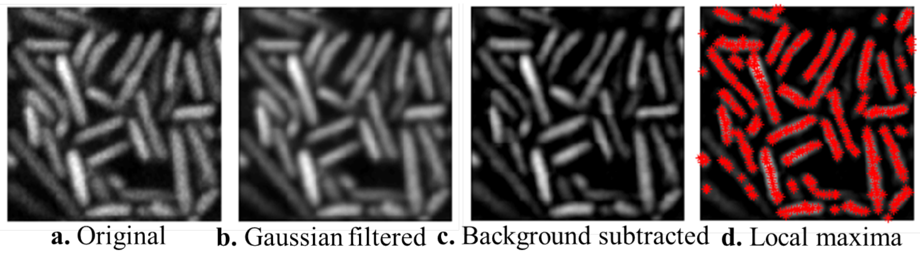


Figure 2.6: Pipeline for finding nodes from bacterial images. Step 1: Filter the original image with a Gaussian kernel (a → b). Step 2: Enhance the signals in the image via background subtraction (b → c). Step 3: Find the local maxima (c → d). Step 4: Clear the points if they have no neighbors or overlap with other points - the rest are the found nodes (red asterisks in d).

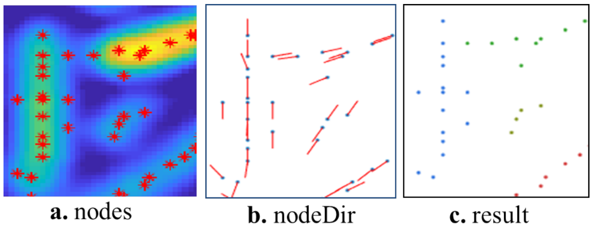


Figure 2.7: An example performance of LCuts with constructed graph features. (a) Nodes are marked in red asterisks. (b) Red lines show the major direction features for each nodes (blue dots). (c) LCut clustering results for the constructed graph.

2.4.2 EVALUATION

The performance of LCuts is analyzed qualitatively and quantitatively by comparing with two current methods used in the bioimaging community, *DensityClust*[89] and *Single Cell tracking*[132]. *DensityClust* is a highly-rated clustering method that has favorable performance in classifying elements based on the relative density around each points. *Single Cell tracking* is one of the representative toolkits for reconstructing bacterial biofilms. It uses a band-pass filter for image denoising and performs marker controlled watershed segmentation to separate single cells. Qualitative comparison is shown in **Fig. 2.8**.

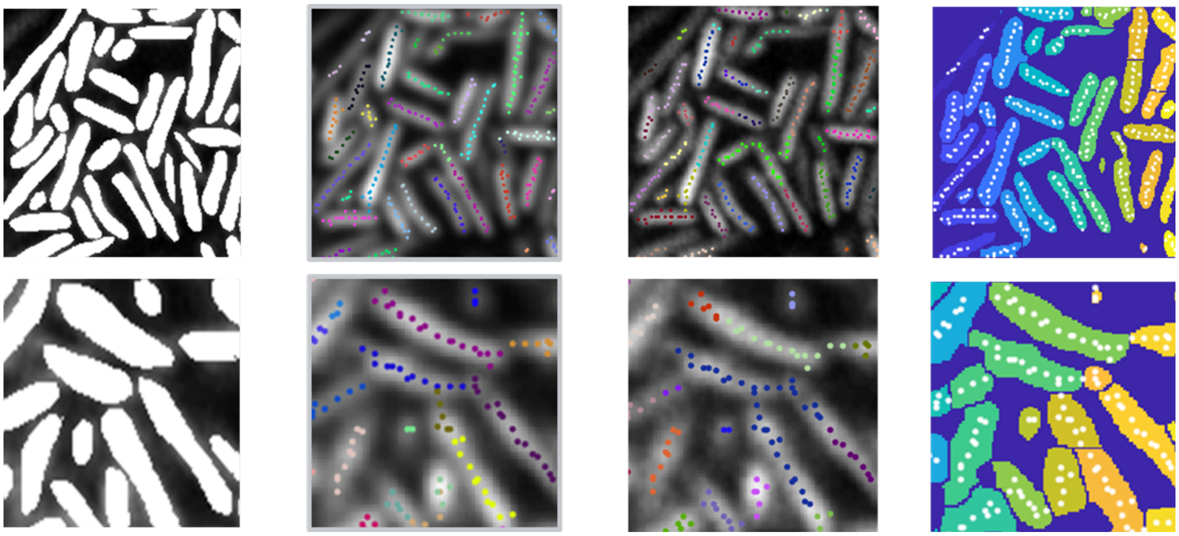


Figure 2.8: Qualitative comparison for the proposed method (second column) with DensityClust[89] (third column) and Single Cell Tracking[132] (fourth column). The first column is manually labeled ground truth. Different colors represent different single cell groups in all the cases.

Two measures, grouping accuracy (GAcc) and counting accuracy (CAcc), are computed for quantitative comparison using *Dice* = 2TP/(2TP + FP +FN). For both measures, the true positives (TP) are the number of nodes per cell that are correctly classified within each cell region. Here, cell regions are manually labeled as binarized ground truth as consistent standard in the comparison. For GAcc, false positives (FP) count the nodes mismatched in experimental data; while, false negatives (FN) represent those nodes missing in corresponding ground truth. Similarly, for CAcc, FP is the count for cells that are over-segmented and FN count for cells that have no corresponding cluster in the experimental result.

	LCuts		DensityClust		SCT	
	GAcc	CAcc	GAcc	CAcc	GAcc	CAcc
Best	**95.9**	**95.1**	94.6	92.3	94.4	94.1
Worst	**87.8**	**85.2**	78.3	83.7	77.8	53.5
Avg.	**91.6**	**91.2**	85.9	87.2	87.7	86.5

Table 2.2: Quantitative comparison of LCuts with DensityClust[89] and Single Cell Tracking (SCT)[132] using *Dice* scores.

Overall, LCuts outperforms DensityClust and SCT in GAcc and CAcc by a margin of at least 4% on average. There are circumstances that some cells are misclassified in LCuts. One cause is the non-linearity of auto-produced point cloud data, especially when cells are randomly floating in the three-dimensional space. Another reason is the trade-off between the tolerance in distance/intensity changes and the continuity of the linear structure.

2.5 EXPERIMENTS ON 3D BACTERIAL IMAGES

2.5.1 DATASET

LCuts can be directly applied on the three-dimensional data. We tested a refined version of LCuts (modifications stated in Chapter 3) on two different types of bacterial experimental 3D datasets: *Shewanella oneidensis* MR-1 and *M. xanthus*. To test the performance of cell identification algorithms, manually segmented ground truth were provided by the cell biologists from the Department of Chemistry, at UVA (**Fig. 2.9**).

Point cloud data: The performance of two different point cloud data inputs is tested. One of the data inputs is the centroids from the manually labeled contours by cell biologists (**Fig. 2.9**). The other one is the central points automatically extracted by detecting local maxima from the intensity image slice by slice along *x*-, *y*-, and *z*- directions (**Fig. 2.6**).

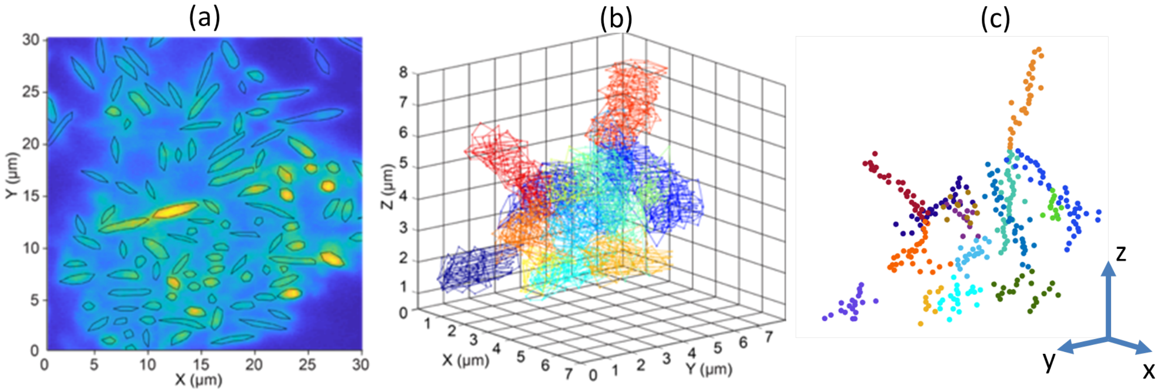


Figure 2.9: Pipeline for manual labeled ground truth for 3D experimental datasets provided by Imaging at the Nanoscale Lab, University of Virginia. (a) Cell contours are manually labeled slice by slice along *x*-, *y*-, and *z*-. (b) Contours from 2D slices are manually grouped together in 3D (one color represents one group). (c) Then the ground truth of the central points is provided as grouped centroids of the contours.

2.5.2 RESULTS AND EVALUATION

Qualitative preliminary results on *Shewanella* and *Myxococcus* are shown in **Fig. 2.10**, where we compared the qualitative results on manually labeled point cloud data and auto-detected centroids using local maxima detection.

Evaluation: Due to the fact that ground truth was manually grouped to reflect the single bacterium layout in 3D space (**Fig. 2.9 (c)**), the ground truth cannot perfectly align with the segmented results in some cases. A new evaluation matrix was used to test the accuracy by matching the segmented result with ground truth using the Hungarian[61] method.

The Hungarian algorithm[61] is a linear programming technique that considers all possible one-to-one assignments between segmented cells at two time points and then determines the set of cell-to-cell assignments that minimizes the sum $\sum cost$. In this case, each segmented point cloud can only be matched with one group in the ground truth where the overall cost of distance and relative angles are minimized. In detail, the cost function for the Hungarian matching algorithm

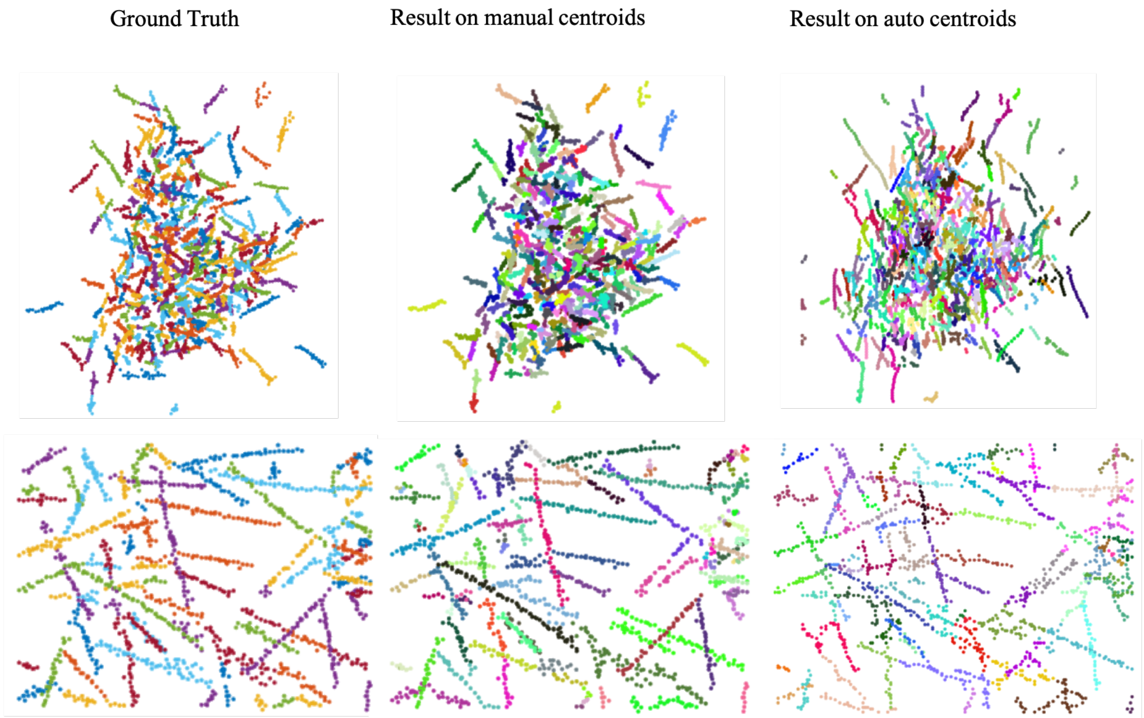


Figure 2.10: Comparison of qualitative results on manual labeled centers and auto-detected centroids. Top: *Shewanella*; Bottom: *Myxococcus*. The cell counting accuracy results using the Hungarian algorithm on manual and auto centroids are 88.7% and 59.5% respectively on the top, and 86.8% and 65.9% on the bottom row.

takes into account the relative distance and angle between two segments, which is defined as:

$$cost = ||c_{gt} - c_{result}||_2^2 \times \text{acos}(dir_{gt}, dir_{result}) \tag{2.8}$$

where c_{gt} and c_{result} denote the centroid coordinates of a segment in ground truth and a segment in the LCuts result. dir_{gt} and dir_{result} are the segment (point cloud) directions.

After the one-to-one assignment via Hungarian algorithm, we are able to visualize and quantify the histogram of distance and angles between matched cells as shown in **Fig. 2.11**. When the number of cells are not the same in the ground truth and the result volumes, which means there are missing cells or extra cells in the results, then those cells are counted as unmatched or false negatives. For the other matched cells, the diagrams in **Fig. 2.11** present the histograms on distance and relative angle cost separately. To quantify the preliminary cell counting accuracy using

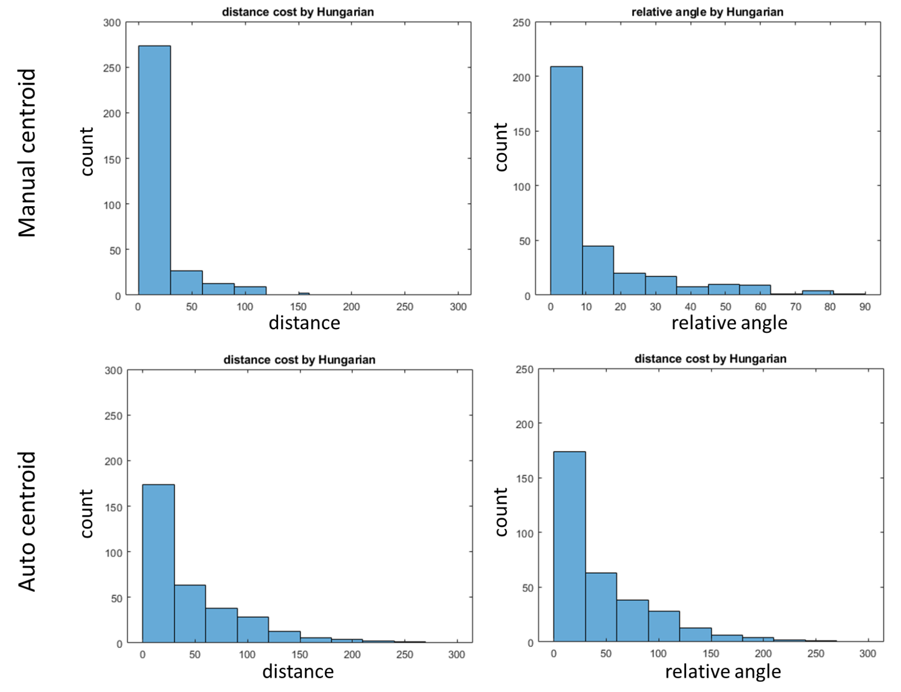


Figure 2.11: Histogram of distance and relative angles between matched cells using Hungarian algorithm. This is an example of *Shewanella* analysis figure. The unmatched number of cells in manual labeled centroids and auto-detected centroids are 33 and 24 respectively. After using the thresholding on distance $\leq$ 60 pixels as a sample, the counting accuracy for the two centroids are 88.7% and 59.5%.

Dice coefficient, we chose a thresholding value on distance of 60 pixels (6μm) to count the cells within the range as true positives while the others are false positives.

The preliminary 3D results reveal that LCuts is promising with manually label datasets; however, there is a challenge for the current workflow that comes from the noisy input of point cloud data (**Fig. 2.10**). Therefore, we proposed an improvement on the automated central axis extraction as more details will be introduced in the next chapter.

2.6 SUMMARY

This chapter firstly presented biofilm segmentation with an edge-based level set segmentation method, named Bact-3D. In Bact-3D, a local velocity constraint was proposed to locate individual cell segment. However, when cell edges are ambiguous, the situation for segmentation becomes more challenging. Thus, an advanced segmentation solution is presented, named LCuts. This solution is a data-clustering approach and has been shown to be efficacious in finding individual bacteria in dense biofilms. It outperforms the existing methods in the majority of cases (2D). Furthermore, LCuts can benefit the biofilm community by its automation and easy adaption to multi-dimensions. With the expected output, cell biologists are able to analyze the biofilm with the quantifiable information provided by LCuts in the form of cellular positions, orientations, and the physical contact points between them. Beyond bacterial biofilms, LCuts can be extended to other linear clustering or segmentation problems where identification of linear structures of the objects is desired.

When we do the best we can, we never know what miracle
is wrought in our life or the life of another.

Helen Keller

3

Generalization of the segmentation method

To generalize the bacterial cell analysis for any segmentation pipelines and to maximize the single cell identification performance, the incorporation of the refined LCuts within a deep neural networks framework was firstly investigated. By utilizing pre-segmented cell regions to generate smooth point cloud data, the performance of LCuts also improves as it can effectively post-process the under-segmented clusters in the u-net processed segmentation results. With more experiments conducted using different initial segmentation methods, the need for additional post-processing emerged where both under- and over-segmentation errors may occur. Post-processing attempts to balance the trade-off between the global goal of cell counting for instance segmentation, and local fidelity to the morphology of identified cells. We present a unified solution, named masked LCuts (m-LCuts), which is an advanced algorithm to automatically detect collinearly structured clusters and to post-process the unsolved cell segmentation errors in an analyzed image of a 3D bacterial biofilm. Generally, the mask can be any preprocessed under- or over-segmented result; then, m-LCuts can automatically further split or connect the incomplete segmentation result. Construction of outlier-removed graphs to extract the collinearity feature in the data adds novelty to *m-LCuts*. The superiority of *m-LCuts* is observed by the evaluation in cell counting with over 90% of cells correctly identified, while a lower bound of 0.8 in terms of average single-cell segmentation accuracy is maintained. This proposed method does not need manual specification of the number of cells to be segmented. Furthermore, the broad adaptation for working on various applications, with the presence of data collinearity, also makes *m-LCuts* stand out from the other approaches.

3.1 Background: the need for post-processing

The need for post-processing commonly exists in most workflows (*e.g.,* [49,132]) due to the difficulties in cell segmentation. As also mentioned earlier, main categories for bacterial biofilm segmentation include thresholding-based[98,86,90,94], watershed-based[11,132,8,49], active models[63,122,6,79], feature-based detection[99,106,66,120,137], and deep learning workflows[116,91,21,77,33,127,108]. A tradeoff that exists among all the methods is between the global goal of identifying the separations among cells and the local goal of preserving maximal appearance of cell morphology. Over-tuning on each side may lead to frequent cases of under-segmentation and over-segmentation of cells. These phenomena are even more common for bacterial cell where an anisotropic cell morphology is presented. Post-processing methods are thus required by many workflows, as either combining the over-segmented clusters or further splitting the diffused cells. Common techniques include but are not limited to region growing, watershed, model fitting, and cutting along curvature. However, none of these methods can target the two post-processing demands naively.

For cases where cells are still connected after segmentation, *i.e.*, under-segmentation and watershed-based methods, are most frequently applied as an additional step[77,33,60]. However, these methods are sensitive to uneven intensities in the images. For large connected cell regions, Gaussian mixture models with the estimated number of cells are used to divide the cluster[49]. Here, the estimated number of cells can heavily affect the final output. Concavity along the cell contour is another index for splitting cells[58,51]. Such methods have not yet been tested for 3D images and they can be computationally expensive when the number of foreground pixels is large. There are also methods for splitting unsolved cases (*i.e.,* under- and over-segmentation errors) based on intensity, segment distance, and curvature[51,106]. However, the current state-of-the-art methods perform sub-optimally when tasked with resolving multiple cells that are similar to each other and closely packed.

Common practices for solving over-segmentation problems are region merging, grouping, and morphological closing. Hartmann et al.[49] connects the neighboring regions by maximum contacting area. This is difficult in the case when multiple over-segmented clusters gather together in parallel layouts. In[41], area and eccentricity information are used to determine if the morphological closing operation is needed to connect the pixels between neighboring segments. This method is however unable to solve complicated cases in 3D where morphological operations might erroneously connect other nearby cells. The authors of Ilastik[10] modeled the edges of over-segmented data as superpixels and trained a random forest classifier to decide if the edges of the two segments need to be merged. The interface for merging 3D cell segmentation is not yet available.

3.2 Cell segmentation using u-net

One barrier in cell segmentation using deep neural networks comes from the limited experimental data and the lack of ground truth for training. This problem arises especially for 3D neural networks where labeling of 3D ground truth is not consistent from volume to volume, and from annotator to annotator. For example, for a 3D image with approximately 300 cells, it takes more than one week for multiple domain experts to get a manual ground truth (**Fig. 2.9**). In this case, the initial choice of the neural network structure is critical.

U-net[91] can provide promising segmentation results with limited training data based on the experiments with a variety of biomedical images. It can extend the training dataset by data augmentation such as scaling, rotation, and elastic deformation. Many extensions of u-net, such as DeepCell[9], BiofilmQ[48], and StarDist[96], also exhibit advantages for learning limited datasets for cell segmentation problems. However, inconsistent 3D ground truth still affects the reliability of qualitative and quantitative assessment of proposed algorithms. Simulated datasets are used to test the feasibility of proposed methods, where the ground truth is automatically produced via simulation.

3.2.1 Data simulation

Simulation of bacterial biofilms is conducted in the Gahlmann lab as a first step in the proposed workflow, named *Bacterial Cell Morphometry 3D* (**BCM3D**)[*]. The procedure of simulating an *E.coli* dataset is shown in **Fig. 3.1**. An agent-based computational model of biofilm growth, CellModeller[93], is used to simulate the layout of bacterial biofilm. Random numbers of cells were placed in the biofilms with two variables: signal to background (SBR) ratio and cell density. The fluorescence emitters for each cell are randomly selected within the cell volume and cell surface to simulate expression of intracellular fluorescent proteins and membrane staining,

---

[*] *BCM3D* is a collaboration manuscript with M. Zhang, et al.

respectively. Each cell was simulated with 500 to 1000 fluorophores to express variations between cells according to biofilm experiments[137]. Convolution is performed with an experimental point spread function (PSF) to produce an intensity image, and noise (approximating the actual distribution) is added. Mix-type of cells are also simulated as myxococcus dataset where spherocylinder cells and fruity body (sphere) cells are present at the same time.

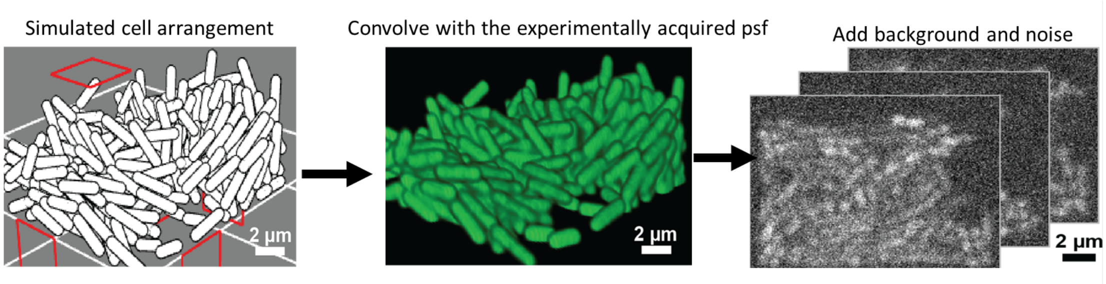
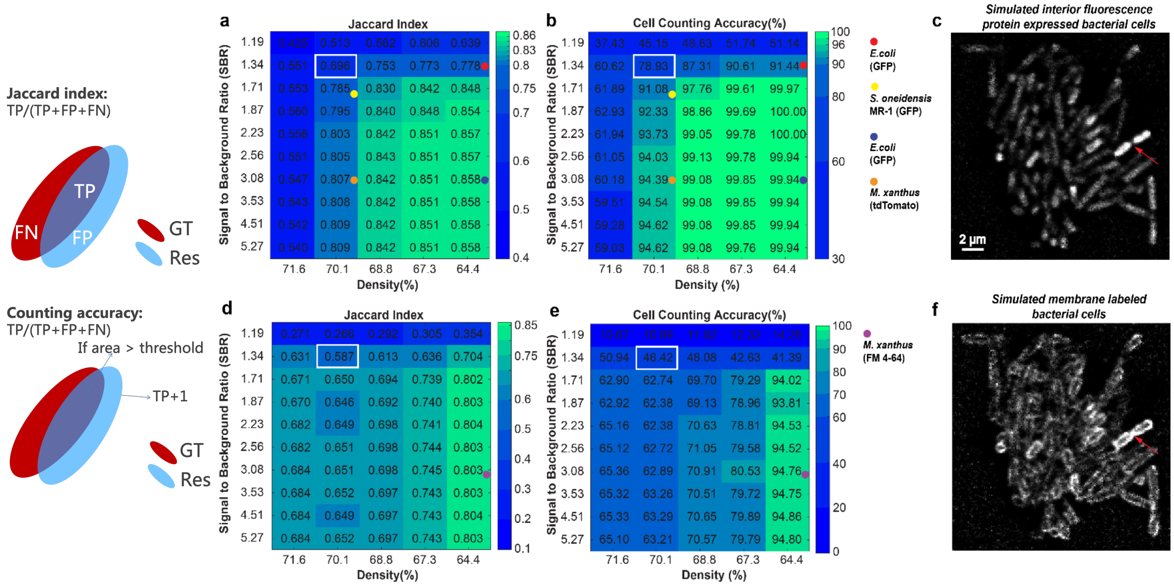


Figure 3.1: Model-based data simulation procedure[137]: Step1: Different layouts of bacterial biofilm are generated by CellModeller[93]; Step 2: Randomly simulate fluorescence emission points in the layouts and convolve these points with experimental acquired point spread function (psf); Step 3: Add background and other microscope noise.

3.2.2 Experiments on simulated datasets

A 3D u-net with three voxel-based classes (cell, cell boundary, and background) was trained and tested in BCM3D[137] using the NiftyNet platform[42]. The network architecture contains depth of 4 convolutional layers. Each convolutional layer has kernels of size $3 \times 3 \times 3$ and uses ReLU as the activation function. There are 32 initial feature maps, and followed by a dropout layer of 50% probability. Ten 3D images ($\sim$ 9 million voxels per image) with different SBRs and cell densities were used for training, and five hundred 3D images were used for testing. The final output of u-net provides a confidence map for each class. The segmentation results are then chosen by thresholding of the cell confidence map with a value that maximizes the cell counting accuracy.

Evaluation: Two metrics are used to evaluate the performance of segmentation results by u-net (**Fig. 3.2**): The *Jaccard Index* represents the segmentation accuracy using TP/(TP + FP



a Jaccard Index

SBR	71.6	70.1	68.8	67.3	64.4
1.19	0.425	0.513	0.562	0.606	0.639
1.34	0.551	0.696	0.753	0.773	0.778
1.71	0.553	0.785	0.830	0.842	0.848
1.87	0.560	0.795	0.840	0.848	0.854
2.23	0.556	0.803	0.842	0.851	0.857
2.56	0.551	0.805	0.843	0.851	0.857
3.08	0.547	0.807	0.842	0.851	0.858
3.53	0.543	0.808	0.842	0.851	0.858
4.51	0.542	0.809	0.842	0.851	0.858
5.27	0.540	0.809	0.842	0.851	0.858

b Cell Counting Accuracy(%)

SBR	71.6	70.1	68.8	67.3	64.4
1.19	37.43	45.15	48.63	51.74	51.14
1.34	60.62	78.93	87.31	90.61	91.44
1.71	61.89	91.08	97.76	99.61	99.97
1.87	62.93	92.33	98.86	99.69	100.00
2.23	61.94	93.73	99.05	99.78	100.00
2.56	61.05	94.03	99.13	99.78	99.94
3.08	60.18	94.39	99.08	99.85	99.94
3.53	59.51	94.54	99.08	99.85	99.94
4.51	59.28	94.62	99.08	99.85	99.94
5.27	59.03	94.62	99.08	99.76	99.94

d Jaccard Index

SBR	71.6	70.1	68.8	67.3	64.4
1.19	0.271	0.266	0.292	0.305	0.354
1.34	0.631	0.587	0.613	0.636	0.704
1.71	0.671	0.650	0.694	0.739	0.802
1.87	0.670	0.646	0.692	0.740	0.803
2.23	0.682	0.649	0.698	0.741	0.804
2.56	0.682	0.651	0.698	0.744	0.803
3.08	0.684	0.651	0.698	0.745	0.803
3.53	0.684	0.652	0.697	0.743	0.803
4.51	0.684	0.649	0.697	0.743	0.804
5.27	0.684	0.652	0.697	0.743	0.803

e Cell Counting Accuracy(%)

SBR	71.6	70.1	68.8	67.3	64.4
1.19	10.67	10.89	11.82	12.32	14.26
1.34	50.94	46.42	48.08	42.63	41.39
1.71	62.90	62.74	69.70	79.29	94.02
1.87	62.92	62.38	69.13	78.96	93.81
2.23	65.16	62.38	70.63	78.81	94.53
2.56	65.12	62.72	71.05	79.58	94.52
3.08	65.36	62.89	70.91	80.53	94.76
3.53	65.32	63.26	70.51	79.72	94.75
4.51	65.33	63.29	70.65	79.89	94.86
5.27	65.10	63.21	70.57	79.79	94.80

Figure 3.2: Evaluation of u-net segmentation performance. Illustration of used evaluation metrics is shown on the left panel. Red cell is the ground truth (GT) and blue cell is the experimental result (Res). (a-c): Performance of images labeled with cytosolic fluorophores; (d-f): Performance of images labeled with membrane-localized fluorophores. Each value shows the averaged performance on 10 datasets for each setting. Dots represent different experimental data setup in terms of SBR and cell density. (c) and (f) show the one slice result with the SBR and cell density marked in the white window, where the u-net failed to split closely connected cells (pointed out by the red arrow).

+FN). True positive (TP) is the number of pixels in which the ground truth overlaps with the experimental result. False positive (FP) is the portion of the segmented areas without any overlap with the ground truth, and false negative (FN) is the non-matching part in the ground truth.

The *Counting accuracy* uses the calculation of Jaccard index as well. TP here counts for the number of results that have an overlapped area (between segmentation result and ground truth) larger than a *thresholding value*, which is also referred to as Intersection-over-Union (IoU) value. FP is the number of cells that are counted in the experimental result but do not exist in the ground truth. FN's are the ones that are not-counted in the result, but are present in the ground truth.

Evaluation of u-net segmentation performance on two different types of labeled datasets is shown in **Fig. 3.2**. The overall performance of u-net is decent in most cases. However, when images are heavily affected by the noise (SBR: low) or cells are closely located with respect to one another (density: high), both metrics show a noticeable drop in accuracy. For example, with the lowest simulated SBR, the average counting accuracy is 46.82% for 50 images labeled with cytosolic fluorophores and only 11.97% for 50 images labeled with membrane-localized fluorophores. From another aspect, when cell density increases from 64% to 72%, the cell counting accuracy decreases from 99% to 60% for the images with higher SBR.

3.3 Post-processing U-net result with refined LCuts

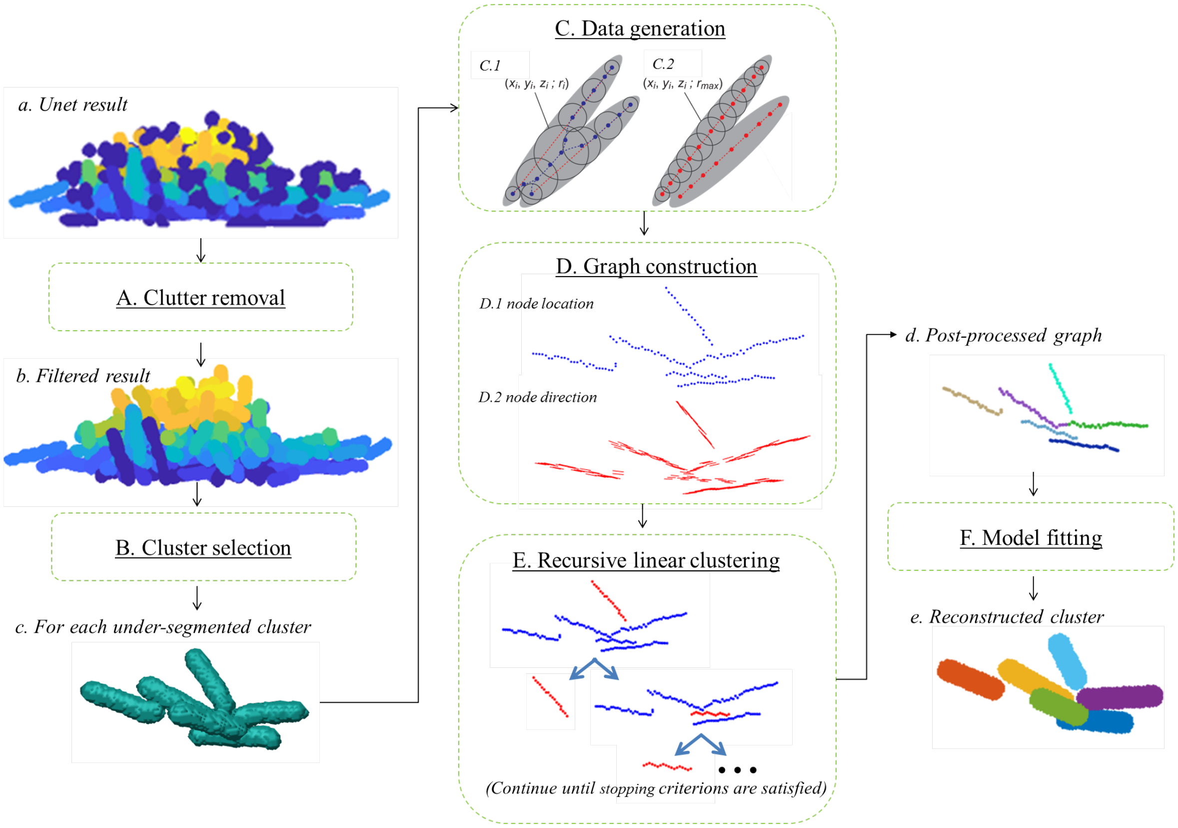


Figure 3.3: Post-processing using refined LCuts[120]. A: Clutter removal using CV-filtering. B: Under-segmented clusters are selected for further splitting based on cluster size. C: Illustration of modified medial axis (dashed red lines) extraction for fused clusters or rod-shaped cells using the method of inscribed spheres. Proposed medial axis extraction limits the radii of inscribed spheres (C.2) overcomes the drawbacks in traditional method (C.1). D: The constructed graph has two node features: location (D.1) and direction (D.2). E: The further splitting of the graph is done in a recursive manner until all the groups meet the stopping criteria. F: The final reconstructed cluster is obtained by model fitting on the post-processed cluster.

When cells are closely touching each other, segmented results after u-net remain connected which forms a cluster with multiple cells (under-segmentation). In the datasets with low SBR, connected voxel clusters may be detected that do not correspond to cells at all (noise). To improve the accuracy of single cell segmentation, post-processing on u-net results using refined LCuts (**Fig. 3.3**) is proposed in the BCM3D workflow.

CV-filtering to remove noisy clutter: Noisy clutter is identified by evaluating the coefficient of variation[136,113] for each connected voxel cluster i (**Fig. 3.3A**):

$$CV_i = \frac{\sigma_i}{\mu_i} \tag{3.1}$$

where σ_i and μ_i denote the standard deviation and the mean of the intensity taken over all voxels contained in connected voxel cluster i. If the coefficient of variation is larger than δ, then the current object will be classified as a noisy clutter and removed from the preprocessed result. The value of δ is selected based on the coefficient of variation of the background. For the data presented here, we used δ=1.1. After CV-filtering, objects smaller than 25% of the expected bacterial cell size are also removed by setting their voxels to zero.

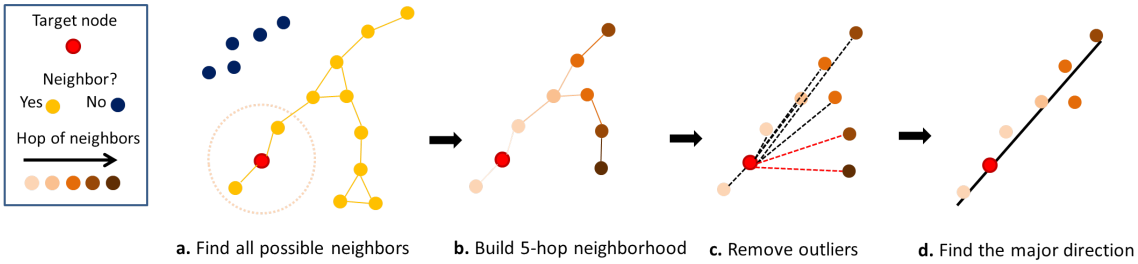
3.3.1 Refined LCuts for post-processing

Refined LCuts preserves the original framework of linear clustering algorithm as described in Chapter 2 with some modifications addressed as follows:

Medial axis extraction: To prepare under-segmented clusters for refined LCuts to further identify and delineate individual cells (**Fig. 3.3 B**), the idea of medial axis extraction using the method of inscribed spheres is exploited[19]. A novel modification is a restriction on the radii of the inscribed sphere that does not exceed the maximum diameter of a single bacterial cell ($d = 0.8\mu$m) (**Fig. 3.3 C**). The set of N inscribed spheres are constrained to be tangent to the object's surface and parameterized by $(x_i, y_i, z_i; r_i \leq d)$ for $i = 1, , N$. Determination of the $(x_i, y_i, z_i; r_i)$ coordinates is achieved using the Euclidean distance transform of the objects' boundary, as described in [133], so that the points with coordinates (x_i, y_i, z_i) reliably trace out the central cell axes of individual bacterial cells.

The fundamental elements of a weighted mathematical graph are nodes, edges, and edge weights. Here, the points with coordinates (x_i, y_i, z_i) represent the graph's nodes. Edges are the connections among nodes. Edges are assigned weights, for example, to reflect the confidence that two nodes belong to the same group. LCuts works by assigning weights to edges in the fully-connected graph to reflect the similarity between two nodes.

Node features: The features of each node include its location and orientation (**Fig. 3.3 D**). The location is each node's Cartesian coordinates. The orientation of each node is found as illustrated in **Fig. 3.4**.



a. Find all possible neighbors **b.** Build 5-hop neighborhood **c.** Remove outliers **d.** Find the major direction

Figure 3.4: Illustration of finding each node's orientation in an outlier-removed neighborhood. a: A neighborhood of the target node is a sub-graph, where all adjacent nodes are connected via edges to the target node. Here, if the distance of two nodes is less than a chosen value (indicated by the dashed circle), these nodes are adjacent to each other. b: A 5-hop neighborhood of the target node is built. c: The directional vectors are found from the target node to the other nodes in the neighborhood (dashed lines). The nodes are evaluated as outliers if they have larger net relative angles compared to all the other directional vectors (red dashed lines). d: Finally, the node's orientation is evaluated as the major direction of the outlier removed neighborhood using principle component analysis.

Stopping criteria: LCuts continues to separate groups of nodes until each group satisfies the stopping criteria. The stopping criteria are based on the expected length of a single cell and a group's linearity after each recursion. The group linearity accounts for node variations compared to the major direction of the nodes. In addition, if two cells are linearly aligned, a gap detection will distinguish long cells and cells aligned in a row. Here, a gap is detected if the neighboring nodes have a distance larger than the average point-to-point distance.

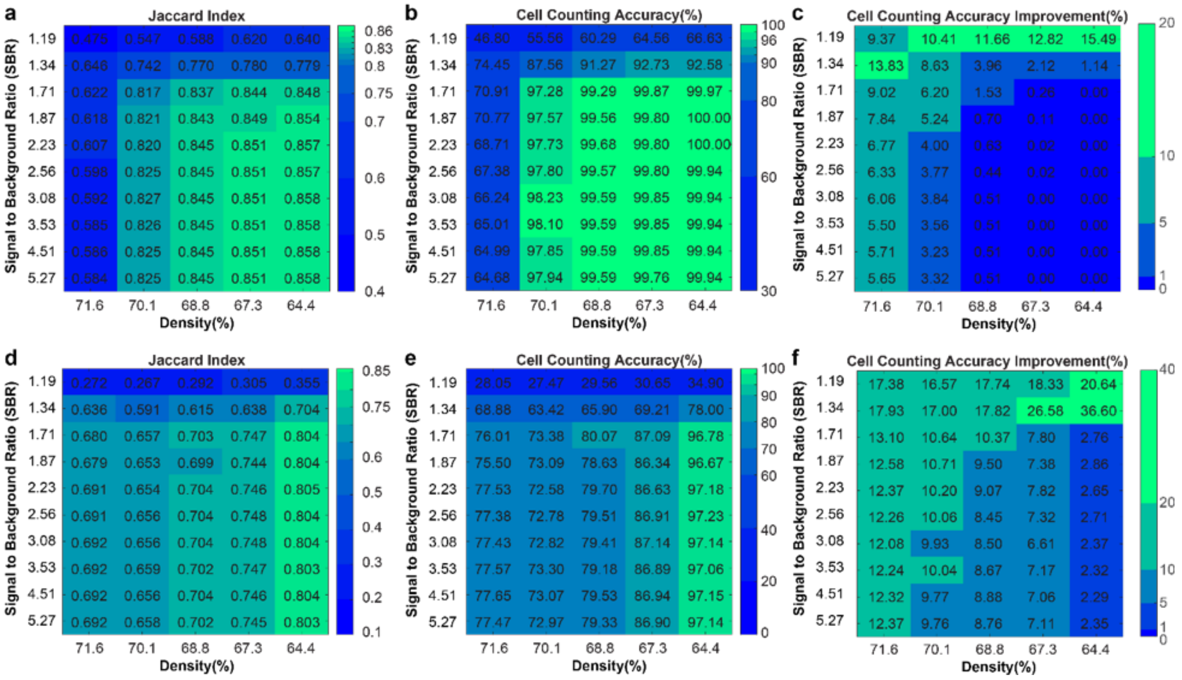


Figure 3.5: Evaluation of segmentation performance after post-processing using the workflow of refined LCuts[137]. (a-c): Performance of images labeled with cytosolic fluorophores; (d-f): Performance of images labeled with membrane-localized fluorophores. (c) and (f) show the improvement of cell counting accuracy after post-processing.

Model fitting: The final output of linear clustering can provide length, location, and orientation of each cell. Based on these linear clusters, the cellular architecture of the biofilms can be reconstructed by placing geometrical models of cells in the space as shown in (**Fig. 3.8F**). More details about biofilm reconstruction will be discussed in **Section 5.3**. As before, when the length of the linear clusters is shorter than 25% of the expected cell length, the segment will be deleted. Here, we chose a spherocylinder for the geometrical model with the radius determined based on known sizes of bacterial cells.

3.3.2 EVALUATION OF POST-PROCESSING PERFORMANCE

The same evaluation metrics are displayed for the performance of post-processed results (**Fig. 3.5**). The Jaccard index, which shows the segmentation accuracy, cannot reveal further splitting performance. This is because further splitting of the cells may erode true positive pixels com-

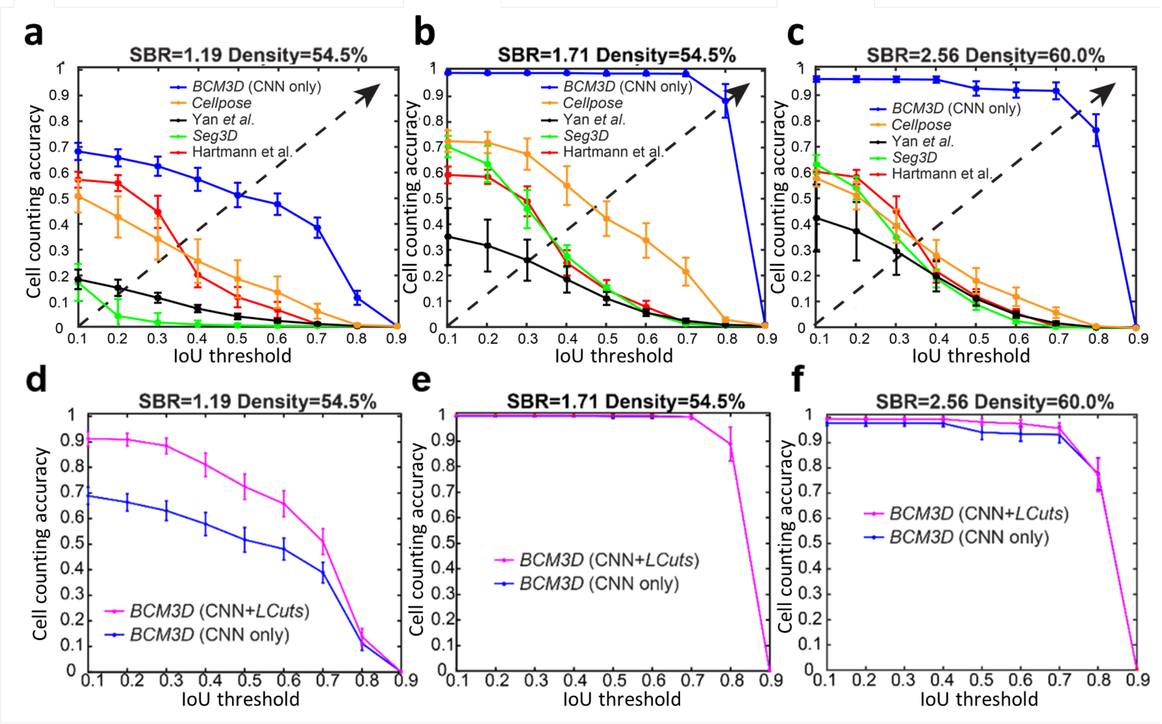


Figure 3.6: Comparison of refined LCuts post-processed results with other segmentation methods in the biofilm community[137], in terms of cell counting accuracy with variant IoU values. (a-c): Performance of cytosolic-labeled data without post-processing with refined LCuts; (d-f): Post-processing performance with refined LCuts.

pared to the ground truth. At the same time, fitting of a spherocylinder model adds false positive pixels. Thus, the improvements on Jaccard index are not representative ($< 5\%$). Instead, it is worth mentioning that post-processing using the proposed workflow can boost cell counting accuracy by over 10% for both datasets (**Fig. 3.5 cf**). This is especially true for membrane-stained datasets, which were a challenge for u-net. A maximum improvement of 36.6% is observed. Although the score for averaged cell counting accuracy remains low for low SBR (typically for membrane-stained dataset), it comes from the weak performance of u-net.

More experiments on the other datasets have been performed. 100 images with mixed labeling using both cytosolic and membrane-localized fluorophores are tested with variant percentage of mixture of labeling, e.g., in an image, one third of the cells are cytosolic-labeled while the others are membrane-labeled. overall, the proposed post-processing pipeline improves u-net results by

$5 - 15\%$. Another dataset with 100 images that includes mixed cell types (spherocylinder cell and fruity body cell) is also improved by a range from 20% to 35% after post-processing. More details can be found in our BCM3D paper[137].

Comparison to other biofilm segmentation methods: We also compared our method, post-processing with refined LCuts, with other cell segmentation methods in the biofilm community[137]. Three selected SBR and Density conditions are selected for quantitative comparison as shown in **Fig. 3.6**. The comparison methods are Hartmann et al.[49], Seg3D[88], Yan et al.[132] and Cellpose[110]. Hartmann et al. and Yan et al. are watershed-based cell segmentation methods. Hartmann et al. also includes post-processing of under- and over-segmented errors using Gaussian Mixture Models and Region merging. The experiments using the these methods were conducted by the Drescher lab. Seg3D uses a slice-by-slice thresholding method followed by post-processing using ellipsoid fitting. Cellpose is a CNN-based cell segmentation model, which was pre-trained with a variety of images, including eukaryotic cells.

According to **Fig. 3.6**, BCM3D (CNN only), which means only trained u-net with simulated dataset to segment the images, already achieves the best performance overall with variant IoU thresholding values. However, the performance degrades at SBR$=1.19$ as the combination of low contrast and high density increases the difficulty in single-cell segmentation. With refined LCuts as a post-processing step, we are able to bring back the cell counting accuracy to the level around 80% for IoU values less than 0.5.

3.3.3 EXPERIMENTS ON REAL DATASETS AND MOTIVATION FOR M-LCUTS

The quantitative and qualitative comparison of experiments on real datasets are shown in **Fig. 3.7**, which is also reported in our BCM3D paper[137]. Due to the difficulty in manual annotation of 3D ground truth of individual cells, only several 2D slices in the 3D raw image are available. Therefore, the results shown in a to c in **Fig. 3.7** are performance averaged over these selected 2D slices. A visual 2D qualitative comparison is also presented in **Fig. 3.7 d**. It can be observed that, with refined LCuts, the regions that are still connected after u-net trained with a simulated dataset (BCM3D (CNN only)) can be further split and filled with estimated cell models, and thus improves the cell counting accuracy.

Motivation for m-LCuts: The refined version of LCuts improves the u-net outputs by filtering out the noisy cell regions and splitting under-segmented clusters into different cells (marked with white circle in **Fig. 3.7 e**). However, there are still imperfect regions that are not yet fixed by refined LCuts (marked with red circles in **Fig. 3.7 e**). These imperfections may be caused by the bending cell shape in the real images or rough segmentation boundaries in pre-segmented outputs, which may further cause over-segmentation errors as a single cell is separated into parts. Therefore, we propose masked LCuts (m-LCuts) in the purpose of generalizing the segmentation method.

3.4 GENERALIZED SOLUTION FOR POST-PROCESSING WITH M-LCUTS

Frequent cases of under- and over-segmentation are often observed when segmenting dense biofilm images using currently available cell segmentation tools[49,132,106,91,21,77,33,122]. Additional post-processing procedures are thus required to refine the segmentation results, with either under- or over-segmentation errors. The proposed masked collinear cuts (*m-LCuts*) can post-process the unresolved clusters by automatically dividing them into different components or by connecting oversegmented objects into larger groups. The user does not need to specify the number of cells in advance. The overall workflow and pseudocode of *m-LCuts* are shown in **Fig. 3.8** and **Algorithm 1**. Details of each step are explained in the following subsections, including introducing **Algorithm 2** and **Algorithm 3**. Currently, the *mode* in the algorithm is a user selected input, but this input can also be determined automatically for different applications, e.g., by checking cluster size and convexity for the bacterial biofilm imaging. The final results can provide cell location, orientation and cell length, and thus assist in biofilm reconstruction.

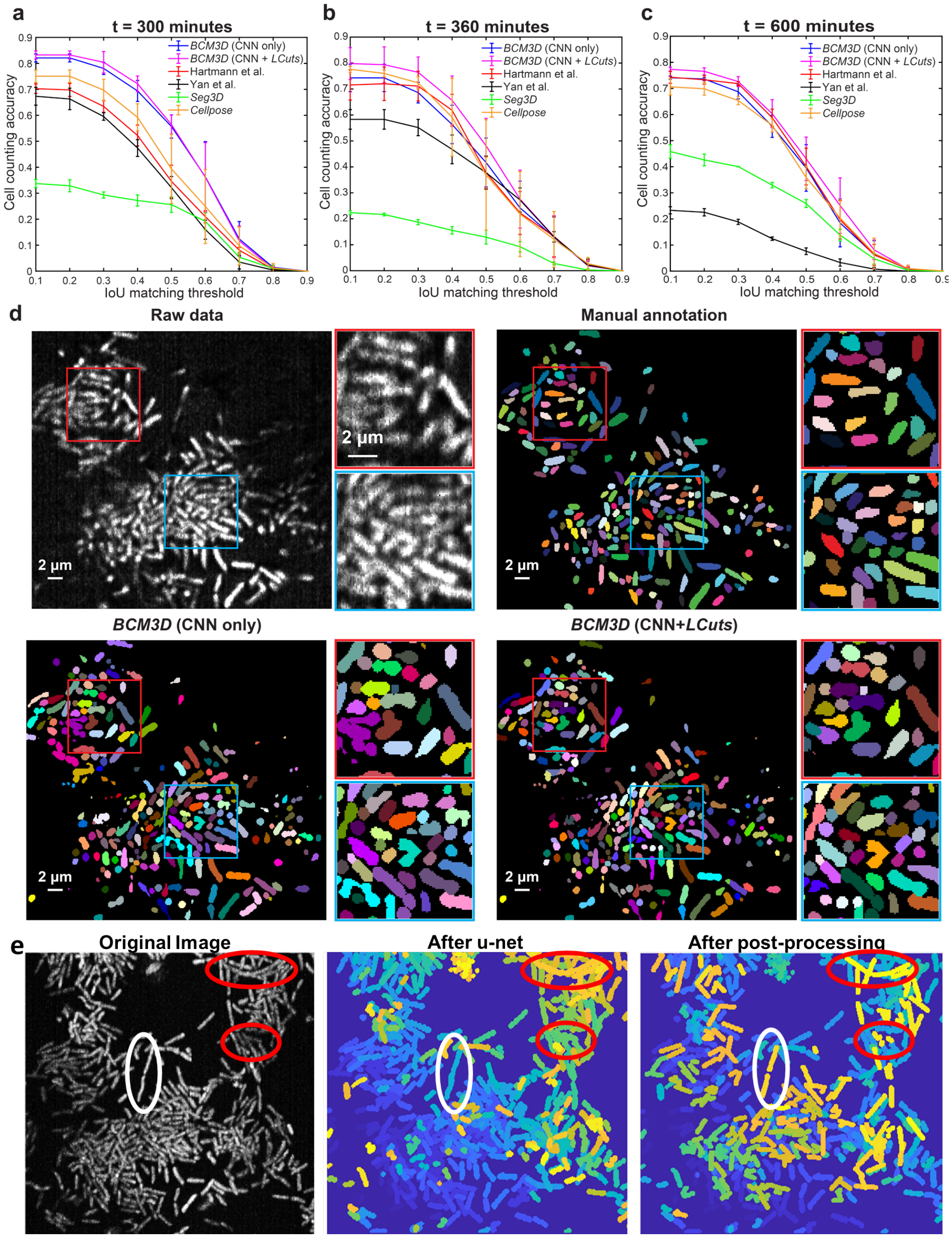


Figure 3.7: Experiments on real experimental image and motivation for m-LCuts. (a-c) Experiments on real experimental image and comparison to the other cell segmentation methods at three biofilm growing time points, 300 mins, 360 mins, and 600 mins[137]. The SBR and Density conditions for these time points are 2.2 and 54.8%, 1.8 and 59.0%, and 1.3 and 64.6%, respectively. (d) The performance of BCM3D without/with refined LCuts on a 2D slice in the raw biofilm 3D image at time = 600 mins. Red and blue boxed areas are enlarged regions for visual performance comparison. (e) Post-processing on real dataset shows that the current pipeline is capable of fixing undersegmented clusters after u-net in most of the cases (success example circled in white). However, when cells are bending or the output from u-net has a rough boundary, imperfect post-processing, such as over-segmentation errors, may occur.

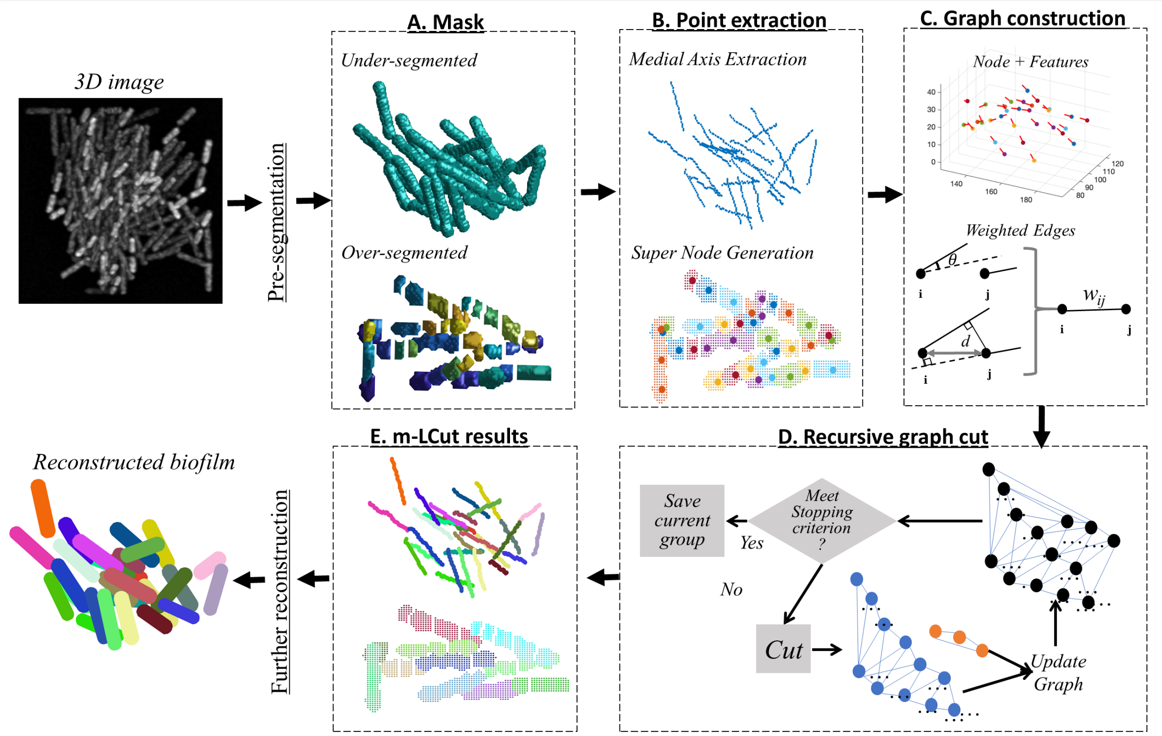


Figure 3.8: Workflow of m-LCuts with the application to the post-processing of unsolved cases in biofilm segmentation.

3.4.1 DIFFERENCES OF TWO MODES

The theory for post-processing an under- or over-segmented mask remains the same, but the algorithm runs in two modes with slightly different specifications to address these different errors. The major differences in the running modes are how key point cloud data is extracted (**Algorithm 1** lines 2-9), how the major direction feature of a graph node is defined (**Algorithm 2** lines 3-10), and how the stopping criterion is reached (**Fig. 3.12**).

3.4.2 DIFFERENCES COMPARED TO PREVIOUS WORK

In contrast to our previously published work (*LCuts*[120] and refined version of *LCuts* used in[137]), this paper reports a generalized and unified data clustering solution with input data either being point or segment to address different post-processing concerns. The main clustering algorithm majorly differs from the other two in the extension to the mode with over-segmented mask. Furthermore, there are also changes in details of the algorithms for generalization: (i) graph construction now contains medial axis extraction and supernode generation; (ii) feature extraction is modified with the new **Algorithm 2**; (iii) the similarity measure defines new measures in distance (Eq. (3.4)) and direction (Eq. (6-8)); and (iv) the stopping criterion adds automatic gap detection and concavity check (**Fig. 3.12**). Compared to the initial work *LCuts*, the node-based graph is constructed differently in *m-LCuts* with binarized masks as input instead of intensity images. The collinear feature extraction in *m-LCuts* replaces the majority voting process in *LCuts* by automatic outlier removal and principal direction computation.

Algorithm 1 Procedure of *m-LCuts* on post-processing

```
Input The Mask of region of interest; mode to be processed.
Output The final clustered groups;
        Optional reconstructed volume postV.
Parameters sizeLimit, distLimit, $\sigma_D$, $\sigma_T$
procedure mLCuts(Mask, mode)
    Find nodes with features:
    if mode: under-segmented mask then
        nodeLoc ← extract medial axis point cloud data;
        nodeDir ← use Algorithm 2 for all the nodes.
    else if mode: over-segmented mask then
        nodeLoc ← generate supernode with segments;
        nodeDir ← use Algorithm 2 for all supernodes.
    end if
    Construct the graph and compute Eq. (3.2) to find W.
    Compute recursive graph cuts solutions using Algorithm 3 to detect collinear
    groups in the graph.
    Optional: reconstruct the volume, postV, by fitting geometrical models to groups.
return
end procedure
```

3.4.3 FROM MASK TO GRAPH

A mask, in general, is a binarized image provided by the preprocessing methods. The target object of interest can be extracted by masking an image where the unselected regions are set to zero. Based on the size of a bacterial cell in the application, a mask can be classified into two categories: an under-segmented mask contains connected cells, while an over-segmented mask encloses cells that are erroneously divided into different parts (**Fig. 3.8A**).

The segmentation problem is transformed into a data clustering problem by constructing a graph from the binary mask. Based on graph theory, the proposed graph has two major components: nodes and weighted edges. The nodes represent the key points in space extracted from the mask. There are two features on the nodes, node Cartesian coordinates (*nodeLoc*) and node major direction (*nodeDir*). The connections between each pair of the nodes are called edges. The weights on these edges reflect the similarity of the connected nodes.

3.4.4 NODES

For under-segmented masks, the nodes lie along the medial axes of the binary image (**Fig. 3.8B** left). A radius-constrained medial axis extraction method is presented, which exploits the idea of inscribed spheres[137]. Inscribed spheres are the largest spheres with at least two tangent points to the object's inner surface. The centers of the spheres are extracted as the object's medial axis, known as the medial axis transform (MAT)[19]. However, accurate implementation of MAT is computationally expensive and hard to process especially in 3D[25]. We adopt the idea of approximate MAT skeletonization using the Euclidean distance transform[100], adding the constraints on the distance ranges to obtain the modified binary foreground for centerline extraction. The motivation of using the MAT with radius constraints (r) is explained in **Fig. 3.9**. The proposed algorithm can avoid the merging of medial axes of touching cells, as well as can maintain the continuity of the extracted medial axes. Although there are still connections between parallel cells

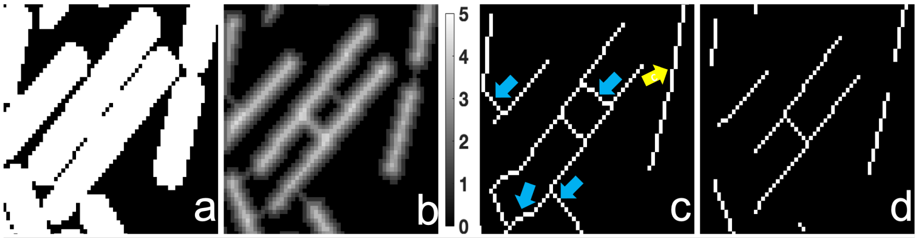


Figure 3.9: Procedure of medial axis extraction with example on a 2D image. Step 1: Compute the Euclidean distance transform (**b**) on the perimeter of foreground (**a**). Step 2: Binarize the distance transform map (**b**) with/without a constraint on distance ranges, which is the so-called radius constraint, to further extract the centerline/skeleton (**c**/**d**). **c**: If there is no radius constraint, the skeleton of neighboring cells are still connected to each other, as indicated by the blue arrows. The ones that are connecting in a line (marked with yellow arrow) are harder to distinguish. **d**: If a radius constraint is added, the medial axes more closely match the central cell axes.

(**Fig. 3.9d**) with the current choice of constraint ($r \in [1, 7]$ pixels), *m-LCuts* is still capable of splitting them, because they present different collinearity. The lengths of cells are shortened due to choice of lower limit of radius constraint, but the loss of length is added back during final cell reconstruction.

For over-segmented masks, the nodes are supernodes representing each connected component, referred to as a segment, in the mask (**Fig. 3.8B** right). The location feature (*nodeLoc*) of a supernode contains the coordinates of all the voxels in the segment, which can be roughly represented by the centroid of the current segment for indexing and visual display (**Fig. 3.8C**).

3.4.5 MAJOR DIRECTION ON NODES

For both cases, the major direction of a node is calculated from its outlier-removed multi-hop neighborhood using principle component analysis (PCA). An example of constructing a 3-hop neighborhood is shown in (**Fig. 3.10**). Suppose G is an undirected fully-connected graph. Then, we remove the edges where the Euclidean distance between the two nodes is larger than a thresholding value, d_T, resulting in subgraph Gn. The two nodes, connected by an edge in Gn, are neighbors to each other. For a target node i, its multi-hop neighborhood is defined as a subgraph, Gn_i. Here, the number of hops is the number of edges that one node needs to traverse

Algorithm 2 Finding major direction feature on node *i*

```
Input
    The inputs are the index $i$ for target node;
    The multi-hop neighborhood $Gn_i$ for node $i$;
    The node locations $nodeLoc$;
    The current algorithm $mode$.
Output
    The major direction feature $nodeDir_i$ on node $i$.
procedure FindMajorDir($i$, $Gn_i$, $nodeLoc$, $mode$)
    for each neighbor $j$ in $Gn_i$ do
        if $mode$: $undersegment$ then
            Find the candidate direction from $i$ to $j$:
            $candidate_{ij} \leftarrow nodeLoc_i - nodeLoc_j$.
        else if $mode$: $oversegment$ then
            Find sub-point cloud that includes all the points in $i^{th}$ and $j^{th}$ nodes;
            Find principal direction of the sub-point cloud:
            $candidate_{ij} \leftarrow$ PCA of sub-point cloud.
        end if
    end for
    $S \leftarrow$ compute pairwise direction similarity between each two directions in
        $candidate$ using eq. (4);
    $S_{max} \leftarrow \max S_j, j \in [1, ..., N_i]$;
    $outlier \leftarrow$ find the indices $id$ of $candidate$ that $S_{max}^{id} \leq$ the median value of $S_{max}$;
    $Gn_i^- \leftarrow$ remove $outlier$ nodes from $Gn_i$;
    $nodeDir_i \leftarrow$ compute the principal direction of $Gn_i^-$.
    return $nodeDir_i$
end procedure
```

to another node. For example, for the target node *i* in Gn_i, three edges are traversed to reach the nodes numbered as 3rd hop.

The major direction to represent the collinearity feature of a target node *i* is the principal direction of the points/nodes in an outlier-removed neighborhood, Gn_i^- (**Fig. 3.10**). Noisy data points (outliers) can dramatically change the trending direction. Thus, **Algorithm 2** is presented to find the major direction of node *i* by removing the outliers from its neighborhood. In this algorithm, *S* is the similarity matrix where each attribute is the pairwise similarity of each two direction vectors using eq. (4). The size of *S* is N_i by N_i, where N_i is the number of neighbors in Gn_i. S_{max} is a column vector that finds the column-wise maximum values in *S*, which corre-

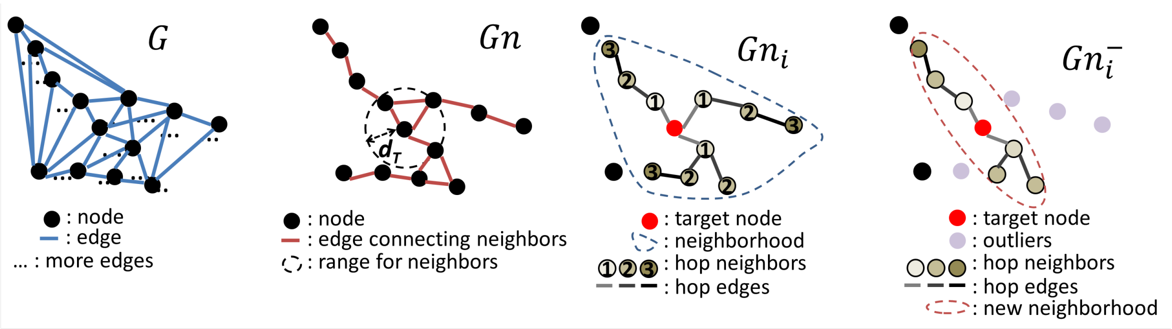


Figure 3.10: Construction of a 3-hop outlier-removed neighborhood for node i. G: a fully connected graph. Gn: a subgraph where all the neighboring nodes are connected to each other. Gn_i: an example of 3-hop neighborhood for node i, where numbered nodes are hop neighbors enclosed in this neighborhood. Gn_i^-: outlier nodes are removed in the outlier-removed neighborhood in order to find the node major direction.

sponds to the maximum similarities for each *candidate* direction on neighbors.

3.4.6 SIMILARITY MEASURE BASED ON COLLINEARITY FEATURES

Given a graph with collinearity features on the nodes (**Fig. 3.8C**), we propose a similarity measure that considers both distance d_{ij} and relative angle θ_{ij} of major directions between two nodes, e.g., i and j. If the adjacency matrix is $\mathbf{W}$, then each attribute $w_{ij} \in \mathbf{W}$ is defined as:

$$w_{ij} = w_D(d, \sigma_D) \cdot w_T(\theta, \sigma_T) \tag{3.2}$$

Both w_D and w_T vary from 0 to 1. The σ_D and σ_T are hyper-parameters that control the decay of similarity measure (eqs. (3.3) and (3.6)).

The similarity measure on distance simply considered the Euclidean distance d_{ij} in Cartesian coordinates (**Fig. 3.11a**). Then,

$$w_D(d, \sigma_D) = e^{-d_{ij}^2/\sigma_D^2} \tag{3.3}$$

where

$$d_{ij} = \begin{cases} ||nodeLoc_i - nodeLoc_j||_2, & \text{for UM} \\ \min(||nodeLoc_i - nodeLoc_j||_2), & \text{for OM} \end{cases} \tag{3.4}$$

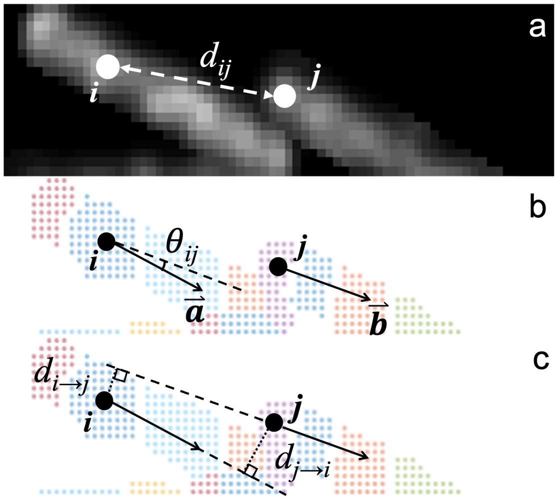


Figure 3.11: Illustration of measuring similarity in distance and relative angle. (**a**): Euclidean distance, d_{ij}, is calculated based on the node Cartesian locations. (**b**): Relative angle, θ_{ij}, weighs the difference in node major direction difference. (**c**): Penalty on direction displacements, $d_{i\to j}$ and $d_{j\to i}$.

Here, $||\cdot||_2$ is L2-norm; OM and UM stand for over- and under-segmented masks. In the case of supernodes (OM), d_{ij} is the shortest distance between two connected components. Pair-wise distances are computed between all the points in the two components, referred as $nodeLoc_i$ and $nodeLoc_j$ in eq. (3.4). In actual experiments, the computation can be simplified by only considering the boundary points in the supernodes.

For similarity measure on node major direction features, if $nodeDir_i = \vec{\mathbf{a}}$ and $nodeDir_j = \vec{\mathbf{b}}$, the measure based on relative angle θ_{ij} (**Fig. 3.11b**) is evaluated as:

$$\cos(\theta_{ij}) = \frac{\text{abs}(<\vec{\mathbf{a}}, \vec{\mathbf{b}}>)}{||\vec{\mathbf{a}}|| \cdot ||\vec{\mathbf{b}}||} \tag{3.5}$$

Then the corresponding similarity measure yields:

$$w_T(\theta_{ij}, \sigma_T) = e^{-(\cos(\theta_{ij})-1)^2/\sigma_T^2} \tag{3.6}$$

One challenging case with the above measure arises when two groups of data are neighbors to each other in parallel ($w_T \rightarrow 1$) and their Euclidean distance d_{ij} is within a *distLimit* such as the case in **Fig. 3.11**. This is particularly important for the over-segmented case, as *distLimit* can be larger than a cell radius. To penalize on these cases, a weighting for direction displacement is added, where $d_{i\rightarrow j}$ is the displacement from node i to the major direction of node j and vice versa for $d_{j\rightarrow i}$:

$$\begin{cases} d_{i\rightarrow j} = \dfrac{||(nodeLoc_i - nodeLoc_j) \times nodeDir_j||_2}{||nodeDir_j||_2} \\ d_{j\rightarrow i} = \dfrac{||(nodeLoc_j - nodeLoc_i) \times nodeDir_i||_2}{||nodeDir_i||_2} \end{cases} \tag{3.7}$$

Then the penalty can be formulated as:

$$\delta_{ij} = \begin{cases} e^{-(\max(d_{i\rightarrow j}, d_{j\rightarrow i}))^2/r^2} & \text{, if } d_{ij} \leq distLimit \\ 0 & \text{, otherwise} \end{cases} \tag{3.8}$$

Here, r is a value retrieved from the estimated maximum radius of the bacterial cells under investigation. In this case, parallel displacement more than a cell radius is more penalized on dissimilarity. The updated similarity measure on node direction feature is then

$$w_T(\theta_{ij}, \sigma_T) = \delta_{ij} \cdot e^{-(\cos(\theta_{ij})-1)^2/\sigma_T^2} \tag{3.9}$$

3.4.7 RECURSIVE GRAPH CUTS

The clustering of different collinearly structured groups follows the theory of recursive graph cuts in LCuts, while more details and pseudocode **Algorithm 3** are added in this chapter. For bacterial biofilm segmentation, the variables (*sizeLimit*, *distLimit*, σ_D and σ_T) are chosen based on prior biological knowledge (see analysis in Sec. 3.5). *distLimit* is the distance limit between two nodes in the same group, and *sizeLimit* is the limit of number of nodes in each group. σ_D and σ_T are semi-automated parameters that can be chosen based on the former two parameter selections. The whole algorithm can be primarily divided into three parts: update the graph, compute the bi-partition solution, and recursively re-partition until reaching the stopping criterion.

The bi-partition solution is given by exploiting the idea of normalized graph cut in[99], which utilizes the total global similarity as well as dissimilarity between the different groups in the image. Given a weighted, undirected graph, $\mathbf{G} = (\mathbf{N}, \mathbf{E}, \mathbf{W})$, where $\mathbf{N}$ contains nodes, $\mathbf{E}$ denotes edges and $\mathbf{W}$ is the adjacency matrix that represents the weightings on the edges. Suppose $\mathbf{L}$ and $\mathbf{R}$ are two disjoint groups and $\mathbf{N} = \mathbf{L} \cup \mathbf{R}$. The goal of a graph cut, at each graph cut level in the recursion, is to disconnect the edges between these two groups where the net weighting, *i.e.*, similarities of node features, is minimized. The disconnection is found by minimizing *Ncut* in[99], which removes the edges between $i \in \mathbf{L}$ and $j \in \mathbf{R}$ in our case (**Fig. 3.8D**):

$$\begin{aligned} &\min Ncut(\mathbf{L}, \mathbf{R}) \\ &= \frac{\sum_{i \in \mathbf{L}, j \in \mathbf{R}} \mathbf{W}(i,j)}{\sum_{i \in \mathbf{L}, n \in \mathbf{N}} \mathbf{W}(i,n)} + \frac{\sum_{j \in \mathbf{R}, i \in \mathbf{L}} \mathbf{W}(j,i)}{\sum_{j \in \mathbf{R}, n \in \mathbf{N}} \mathbf{W}(j,n)} \end{aligned} \tag{3.10}$$

Here, the numerators take into account the cross-group weightings. The denominators calculate the connecting weighted edges between nodes in either $\mathbf{L}$ or $\mathbf{R}$ to all the other nodes in $\mathbf{N}$, which establish the unbiased normalized association between each group and the whole graph.

By adding an indicator vector $\mathbf{x}$, where $dim(\mathbf{x}) = dim(\mathbf{N})$, define $x_i = 1$ for $i \in \mathbf{L}$ and $x_j = -1$ for $j \in \mathbf{R}$. Also, let $\mathbf{d}$, ($dim(\mathbf{d}) = dim(\mathbf{N})$) be the total connection matrix between one node, i or j, to all the other nodes $n \in \mathbf{N}$. Then, eq. (3.10) can be rewritten as:

$$\min_{\mathbf{x}} Ncut(\mathbf{x}) = \frac{\sum_{(\mathbf{x}>0)} w_{ij} x_i}{\sum_{(\mathbf{x}>0)} d_i} + \frac{\sum_{(\mathbf{x}<0)} -w_{ji} x_j}{\sum_{(\mathbf{x}<0)} d_j} \tag{3.11}$$

Let $\mathbf{D}$ be the diagonal matrix where each diagonal is the total connection $\mathbf{d}$ from one node to all the nodes in the graph, and $\mathbf{W}$ be the adjacent matrix that stores w_{ij} in each cell. Meanwhile,

Algorithm 3 Pseudocode for recursive collinear graph cuts using advanced LCuts on a single component.

```
Input The current nodes $sinComp$ and node features $nodeLoc$ and $nodeDir$;
      The current adjacency matrix $adj$ of $sinComp$;
      The $status$ of recursion; The current $groups$ and the number of groups $n$ in the data.
Output The updated $groups$ in the data; The updated number $n$ of groups.
Parameters $sizeLimit$, $distLimit$, $\sigma_D$, $\sigma_T$, $mode$
Initialize $n \leftarrow 1$, $status \leftarrow 0$, $groups\{n\} \leftarrow 0$, $adj \leftarrow \mathbf{W}$, $sinComp \leftarrow \mathbf{N}$
procedure LCuts($inputs, parameters$)
    $sinComp \leftarrow$ current single component for LCuts;          ▷ Update the graph
    $linearity \leftarrow 0$;
    if $mode$: $undersegment$ then
        $curSize \leftarrow$ number of nodes in $sinComp$;
    else if $mode$: $oversegment$ then
        $curSize \leftarrow$ number of all points in supernodes;
    end if                                                          ▷ Check recursion stopping criterion
    if $sizeLimit(1) < curSize \leq sizeLimit(2)$ then
        $linearity \leftarrow$ check collinearity of $sinComp$;
    else if $curSize \leq sizeLimit(1)$ then
        $status \leftarrow 1$;
    end if
    if $status = 1$ or $linearity = 1$ then
        $groups\{n\} \leftarrow$ save $sinComp$ in the result; $n \leftarrow n + 1$;
    else                                                            ▷ Bi-partition $sinComp$
        $adj \leftarrow$ Update current adjacency matrix;
        $Comps \leftarrow$ Compute bi-partition solution;
        if $Comps$ just has one group then
            $status \leftarrow 1$; $sinComp \leftarrow Comps$;          ▷ complete partition
        else
            $status \leftarrow 0$;                                    ▷ not complete
            $sinComp \leftarrow Comps_{\mathbf{L}}$;
            return LCuts($inputs, parameters$);
            $sinComp \leftarrow Comps_{\mathbf{R}}$;
            return LCuts($inputs, parameters$);
        end if
    end if
    return
end procedure
```

set $\mathbf{y}$ as a deformed indicator vector relaxed to take real values:

$$\mathbf{y} = (1 + \mathbf{x}) - \frac{\sum_{(\mathbf{x}>0)} d_i}{\sum_{(\mathbf{x}<0)} d_j}(1 - \mathbf{x}) \tag{3.12}$$

Then the normalized bi-partition problem in eq. (3.11) yields a generalized Rayleigh quotient:

$$\min_{\mathbf{y}} \frac{\mathbf{y}^{\mathrm{T}}(\mathbf{D}-\mathbf{W})\mathbf{y}}{\mathbf{y}^{\mathrm{T}}\mathbf{D}\mathbf{y}} \tag{3.13}$$

where a optimal closed-form solution of the above problem is given by solving $(\mathbf{D}-\mathbf{W})\mathbf{y} = \lambda\mathbf{D}\mathbf{y}$. The eigenvector with the second smallest eigenvalue then indicates the optimal bi-partition of the current graph[99].

To automatically determine the continuity or termination of each recursion, the stopping criterion is checked, which considers the size and collinearity (**Algorithm 3** lines 9-14). A cut component that is smaller than *sizeLimit* represents a final group of nodes if that group also satisfies the collinearity criterion. The check for collinearity is mainly composed of two parts (**Fig. 3.12a** and **b**) with an additional part (**Fig. 3.12c**) for supernode mode.

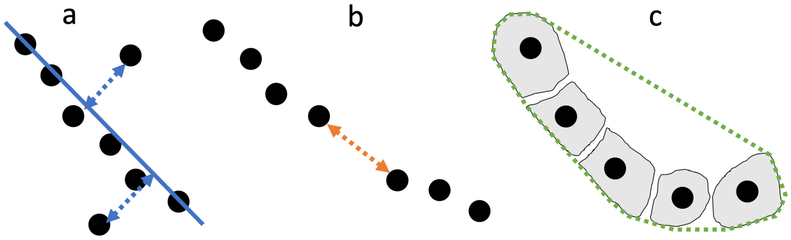


Figure 3.12: Stopping criterion for collinearity check of a cut component. (**a**) Deviation detection: check if there are points apart from principal fitted line of the component that is larger than a cell radius. (**b**) Gap detection: check if the distance between a pair of neighboring points is larger than node distance limit. (**c**) Concavity check: compare ratio of the volume of the convex hull and the total number of points in all the supernodes (yielding the approximate volume of current cell.)

3.5 EVALUATION ON THE PERFORMANCE OF M-LCUTS

3.5.1 DATASET

The validation of the performance for analyzing densely packed biofilms in 3D is challenging, because of the limited amount of experimental data and the lack of ground truth. Manual anno-

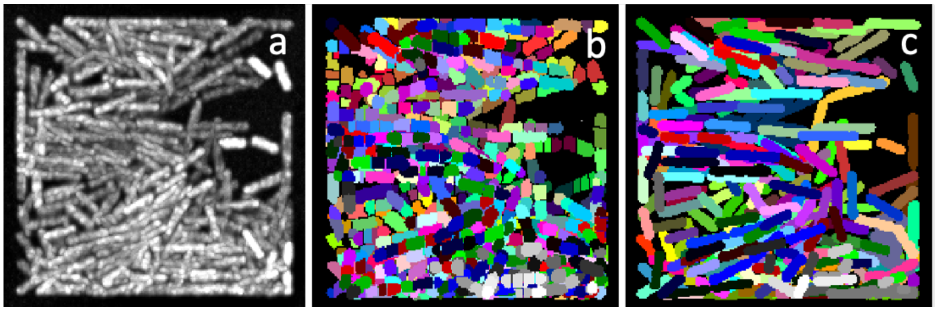


Figure 3.13: Max intensity projection along z-axis of (**a**) an example input image, (**b**) over-segmented segmentation results, and (**c**) under-segmented results. Different colors denote different connected components.

tation of experimental data is time consuming and unreliable[137]. Thus, we used simulated 3D biofilms with ground truth to generate the regions of interest (ROI). The process of data simulation is described in[137], where the data mimic the actual microscopic images collected by lattice light sheet microscopy (LLSM). The mean signal-to-background ratio and cell density of the simulated image dataset in this paper are 2.2 and 60%. The background intensity and read noise of the microscope are simulated by Poisson noise ($\lambda = 199.73$) and Gaussian noise ($\sigma = 3.04$). The voxel size of each image is $100\text{nm} \times 100\text{nm} \times 100\text{nm}$.

The input images (**Fig. 3.13**) for image analysis are preprocessed with basic deconvolution and background subtraction[137]. Then, masked ROI with under-segmentation and over-segmentation errors are used for quantitative and qualitative performance evaluation for the proposed method. Although the quality of the masks after initial segmentation is not the emphasis for assessing *m-LCuts*, the *Baseline* performances are still reported in **Tables 3.1, 3.2** and **Figs. 3.15, 3.17** to demonstrate the effectiveness of post-processing. *Baseline* compares the initial segmentation results in a mask to ground truth for the ROI.

3.5.2 EVALUATION METRICS

The data clustering performance of *m-LCuts* is evaluated using three measures: cell counting accuracy (**CA**), single-cell grouping accuracy (**SGA**) and single-cell boundary F1 score (**SBF1**).

The calculation of each score is described below.

The cell counting accuracy accounts for how many cells are correctly identified as true positives (TPs) in the post-processed result ($postV$) when compared to the ground truth (GT). False positives (FPs) are the detected cells that do not exist in GT but are identified in $postV$. False negative (FN) counts the number of cells that are missing in $postV$.

$$\mathrm{CA} = 2 \cdot N_{TP} / (N_{postV} + N_{GT})$$

Here, N_{postV} and N_{GT} are the number of groups or segments in $postV$ and GT respectively. Successful counting (N_{TP}) of cells occurs when the number of overlapped voxels/nodes, $||GT \cap postV||_0$, is larger than the IoU value. IoU value stands for Intersection-over-Union, which indicates the fraction of intersecting area of two components over their union.

The single-cell grouping accuracy evaluates how many nodes/voxels are correctly classified/grouped in a cell region. It takes the mean value over matched segments i that have corresponding pairs of $postV_i$ and GT_i.

$$\mathrm{SGA} = \frac{1}{N_{match}} \sum_{i}^{N_{match}} \frac{2 \cdot ||GT_i \cap postV_i||_0}{||GT_i||_0 + ||postV_i||_0}$$

The number of matching segments is N_{match}. The percentage of overlapped nodes/voxels over the total number of nodes/voxels is used to evaluate SGA for each segment.

The single-cell boundary F1 score averages the scores on how many points on the boundary of the segment can match with the contour of corresponding GT at single-cell level. Boundary F1 score, or contour matching score, for semantic image segmentation was proposed in [24]. According to its definition, $precision_i$ is the ratio of matching boundary points in $postV_i$ and GT_i, that are within a distance threshold, to the total length of $postV_i$ boundary. Similarly, $recall_i$ is the ratio of the matching boundary points to the total length of GT_i boundary.

$$\text{SBF1} = \frac{1}{N_{match}} \sum_{i}^{N_{match}} \frac{2 \cdot precision_i \cdot recall_i}{precision_i + recall_i}$$

While SBF1 measures boundary fidelity, SGA measures pixel-wise volumetric overlap.

3.5.3 UNDER-SEGMENTED MASK MODE

The under-segmented data-set is produced by using the output confidence map of cell interior trained with a 3D u-net as described in [137]. The confidence map for a class indicates the probabilities that a given voxel belongs to a given class. To generate under-segmented masks, the confidence map is thresholded with a value of 0.5. Clusters that are larger than a single cell volume are then automatically selected. The number of cells in each under-segmentation mask is between 1 to 29, which sums up to 211 different under-segmented cell clusters. The dataset contains a few clusters with only one cell, because their size is larger than a normal estimated bacterial size. However, the algorithm can decide itself whether to post-process further.

The experiments and qualitative performance of *m-LCuts*, compared to other works with under-segmented masks, are shown in **Fig. 3.14**. In this mode, *m-LCuts* aims to automatically find collinearly-featured structures in the point cloud data after medial axis extraction, with the purpose of enhancing cell counting accuracy. The output of the algorithm is comprised by the different clusters of central cell axes that represent individual cells, so only CA and SGA are available for evaluation as shown in **Table 3.1**. Clustered segments that are less than the minimum cell length are eliminated, because such segments, especially those extracted from the connection of touching cells, do not represent a cell (**Fig. 3.9**). A reconstruction of the biofilm is also provided by *m-LCuts*, which substitutes each node cluster with sphero-cylinder models.

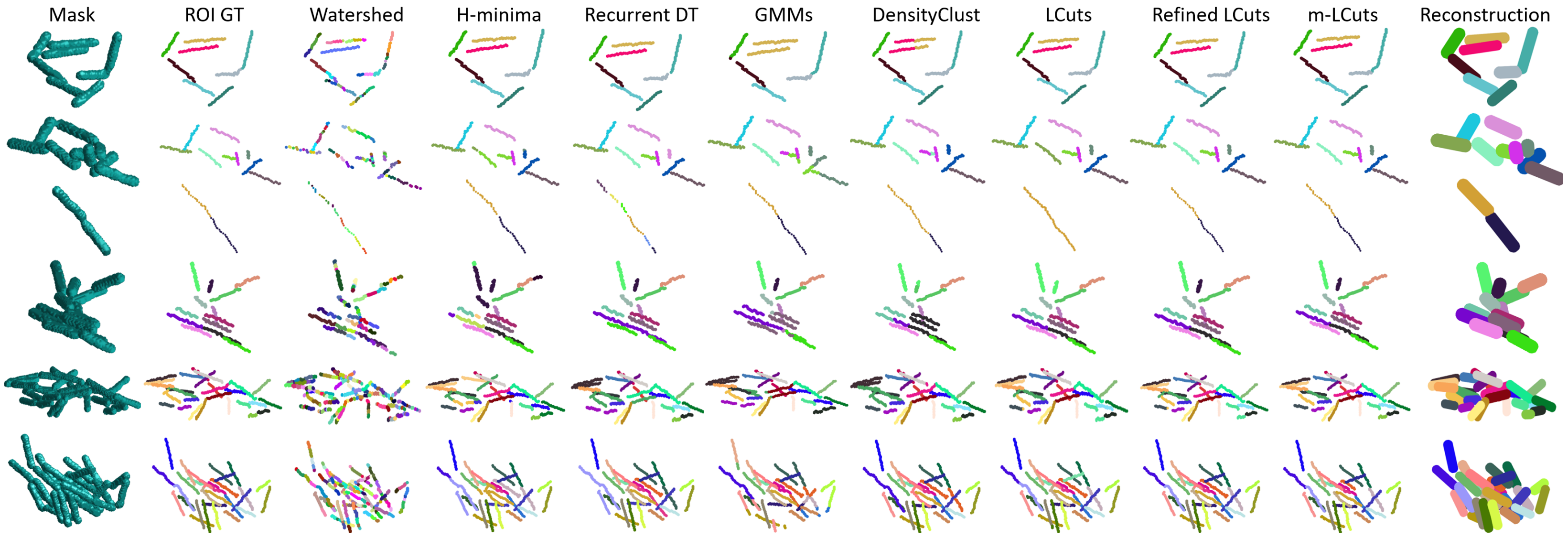


Figure 3.14: Performance on post-processing under-segmented masks. The rows are different example results. The columns are: Under-segmented mask, ground truth of masked ROI, the post-processing methods in [77], [132], [92], and [49], DensityClust [89], the original LCuts [120], refined LCuts applied in [137], proposed *m-LCuts* method and its reconstructed results. Segments that can match with ROI GT are coded with the corresponding colors, while the others are randomly color-coded.

3.5.4 Comparison for under-segmented mode

For the post-processing methods compared in this section, the final clusters are displayed by pseudocoloring the medial axes of the clusters (**Fig. 3.14**). For a given data clustering method (e.g., DensityClust), the results are obtained by directly testing on the medial axes point cloud data. Details of the compared methods and observed limitations are listed below. Parameters were chosen to achieve relative optimal performance.

Watershed: The basic watershed is utilized in CellProfiler[77] to further split under-segmented cellular regions. The software over-segments the data due to the rough surface of the non-spherical binary mask. We used *watershed* function in MATLAB to test this method. It appears that it over-segments the data due to the rough surface of the non-spherical binary mask.

Hminima: Yan et al.[132] applies H-minima transform[103] to mark the high intensity regions of a cell, followed by watershed. The tuning parameter, h-minima height, was chosen to be 0.3 times the maximum intensity of the image in the code[132]. Although this method shows apparent improvements in single cell identification compared to basic watershed, a cell can still be mistakenly labeled with multiple markers caused by inhomogeneous image intensity.

Recurrent DT: Roszkowiak et al.[92]added recurrent distance transform to consider the morphology of the segment for marker-controlled watershed. The optimal threshold to find the foreground marker is recurrently searched until reaching a *no split* criterion. We define this criterion as segments are successfully separated into different groups to identify individual bacteria. Two limitations are observed: one is that it is sensitive to the pre-segmented morphology, where a rough and thin object may lead to over-segmentation; the other one is that it fails to split connecting components when the diameter of their connection is similar to the cell body.

GMMs: Gaussian mixture models are fitted to clumped clusters to identify single cells as used in[49]. It needs manual input on the number of clusters, *k*, in the data. We automatically estimated the value based on the approximate regular size of a cell (4 µm in length and 0.4 µm in radius).

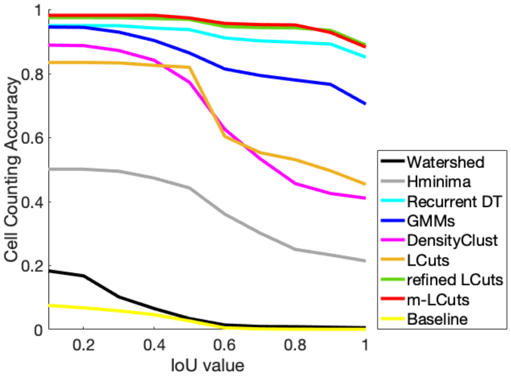


Figure 3.15: Comparison on cell counting accuracy of under-segmented masks averaged over 211 experiments with variant IoU values.

The value of k was found to be normally equal to or less than the actual group numbers. It is challenging to determine a proper class number when different single data sizes are present in the group. Additionally, the fitted Gaussian models vary from time to time; even sometimes they are unable to converge. Its performance degrades when cells are densely packed.

DensityClust: DensityClust[89] finds the cluster centers based on local density and global neighboring distance. This method was tested with the code provided in[32]. We changed the manual selection of cluster centers to automatically select the top *numClust* high densities. Here, *numClust* is also estimated from the single cell size. This method thus also requires the number of clusters to be specified as input and is prone to clustering errors.

LCuts & refined LCuts: Our prior work reported in[122] and[137]. *m-LCuts* is an advanced version of the earlier approaches with better performance in a under-segmentation mode, and with a major advance in being able to accommodate an over-segmented mask. The details of the differences are described in **Sec. 3.4.2**.

Advantages and limitations: Overall, *m-LCuts* outperforms the aforementioned methods both qualitatively (**Fig. 3.14**) and quantitatively (**Fig. 3.15** and **Table 3.1**). Although the performance of refined LCuts is comparable to the current m-LCuts version according to **Fig. 3.15**, the refined LCuts cannot fulfill the need for post-processing over-segmentation conditions. *Re-*

	CA	**SGA**
Watershed[77]	0.065 ± 0.122	0.483 ± 0.153
Hminima[132]	0.473 ± 0.210	0.835 ± 0.168
Recurrent DT[92]	0.942 ± 0.147	0.971 ± 0.076
GMMs[49]	0.904 ± 0.186	0.919 ± 0.134
DensityClust[89]	0.842 ± 0.191	0.832 ± 0.143
LCuts[120]	0.826 ± 0.116	0.855 ± 0.147
refined LCuts[137]	0.972 ± 0.092	0.984 ± 0.051
m-LCuts	**0.982** ± 0.059	**0.986** ± 0.049
Baseline	0.046 ± 0.038	0.560 ± 0.166

Table 3.1: Quantitative comparison of cell counting and point grouping/classification performance on further splitting under-segmented cells. All the scores include the mean value over all the experiments plus the standard deviation. Here, IoU threshold for CA is 0.4. *Baseline* compares the initial segmentation results in a mask to ground truth for the ROI.

current DT also exhibits stable cell counting accuracy with variant IoU values. Visual inspection of the results shows that *Recurrent DT* fails in the cases of closely end-to-end connected clusters and non-smooth segments with variant changes in diameter (e.g., **Fig. 3.14**, 3rd row). The margin between the two, though, is not larger than the other methods, which is because the under-segmented mask dataset does not have many of those challenging cases. *m-LCuts* presents the best performance in both averaged CA and SGA scores with the smallest standard deviation (**Table 3.1**).

The major advantages of *m-LCuts* include the lack of requirement to specify the number of clusters, the stability in varying inter- and intra-cellular intensity environments, and the insensitivity to the smoothness of the mask. However, mask morphology is still important for *m-LCuts* in terms of mask diameter and initial segmentation error. If the associated inscribed sphere diameters (Section III-B) along the medial axes of a mask are much less than an expected minimum cell radius, the current medial axis extraction method might create a super sparse point cloud data with no collinearity feature presented. In this case, the lower limit of the radius in medial axes extraction can be set to zero to consider these weak connections. As a consequence, connections between collinearly connected cells will also be included in the point cloud data (like the condition in **Fig. 3.14**, 4th row), but *m-LCuts* is able to determine the continuity of recursive graph cuts based on the cell length limit. Another factor is the falsely detected or missing cell regions in the initial mask, which affects the collinearity feature in the data. This limitation is going to be discussed later.

3.5.5 OVER-SEGMENTED MASK MODE

The over-segmented results are generated by H-minima transform based marker-controlled watershed method as used in [56,132]. All minimum inverse intensities less than the height H are used as markers. The value of H controls the over- and under-segmentation rate in the result, and is chosen as around 4% of the maximum image intensity in this paper to create more regions of interest with over-segmentation. Then, the over-segmented regions of interest (ROIs) are randomly chosen from the over-segmentation results, which further generate the masked dataset (Mask). Challenging conditions, such as parallel, T-, and L-shaped cell arrangements, were given particular scrutiny (**Fig. 3.16**).

3.5.6 COMPARISON FOR OVER-SEGMENTED MODE

Similarly, experiments on post-processing over-segmented masks start from graph construction, where each segment are treated as a supernode. The qualitative performance of *m-LCuts* masked with over-segmentation results is shown in **Fig. 3.16**. Different segments that belong to the same cell structure are merged into the same cluster (represented with the same color).

The inputs of all the methods in this section include all the regions/points in the masks. The parameters are selected empirically or based on prior information about the bacterial biofilm.

Region Merging: Region merging by connecting maximum contact areas is used to fix over-segmentation problems in [49]. Here, regions are recursively merged by their maximum contact area until all the segments were traversed. This method can correctly merge the components if

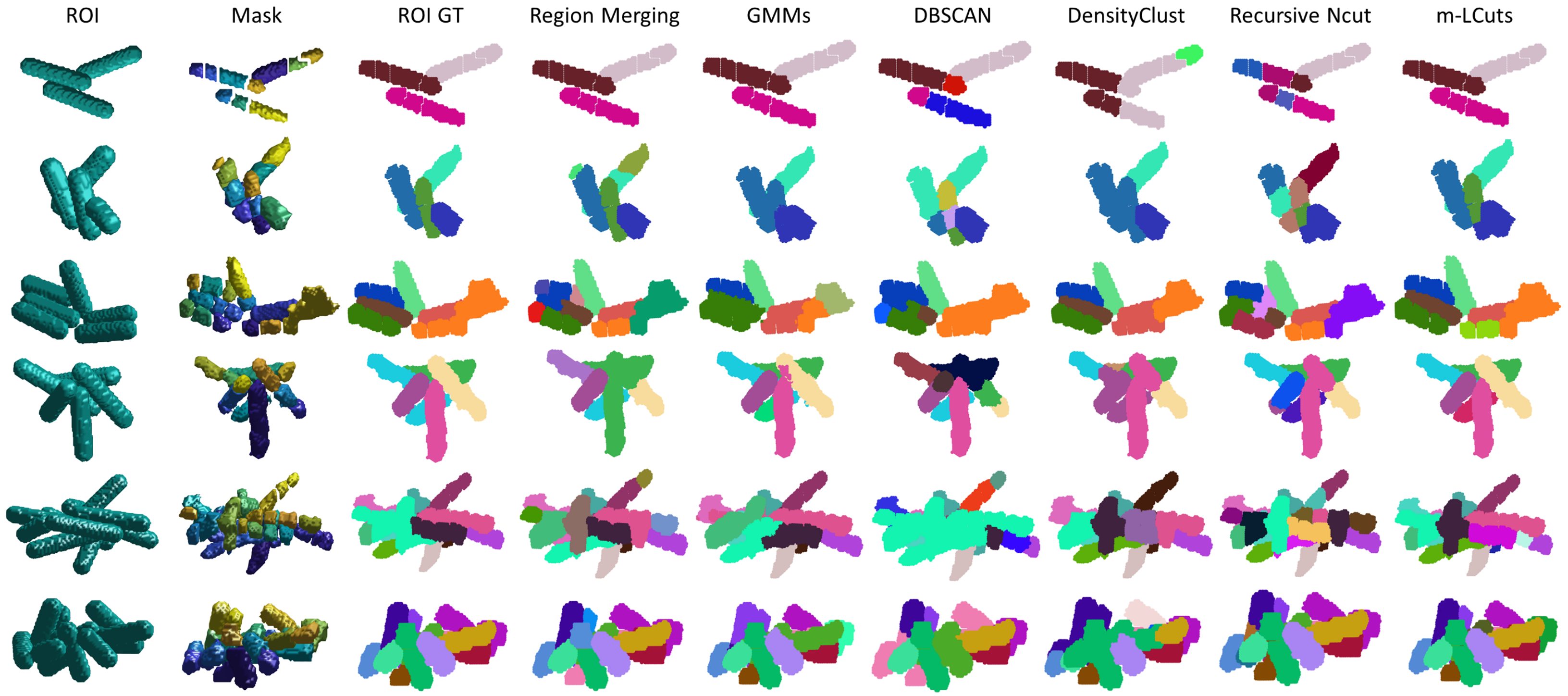


Figure 3.16: Performance on post-processing over-segmented masks. The rows are different example results. The columns are: ROI, over-segmented mask, ground truth of masked ROI, post-processing method in[49], clustering approaches in[13],[97], and[89], recursive Ncut, and proposed *m-LCuts*. Segments that can match with ROI GT are coded with the corresponding colors, while the others are randomly color-coded.

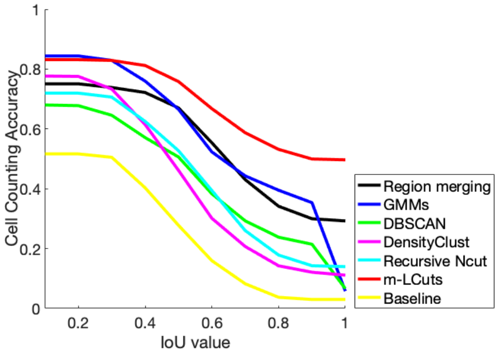


Figure 3.17: Quantitative comparison on cell counting accuracy of over-segmented masks. The scores are averaged over 125 over-segmented data.

one cell is just over-segmented into two or three segments and their maximum contacting sides are different on each side of the segment. It is unable to resolve the cases when cells are packed in dense side-by-side configuration or are extremely over-segmented.

GMMs and *DensityClust:* The same algorithms as we introduced earlier are tested here. The cluster numbers are chosen with respect to cell size. Similar to our previous observation, both methods are not only sensitive to the choice of cluster number, but also unable to separate closely packed cells.

DBSCAN: DBSCAN[97] is a leading data clustering method based on data local density. Similar to *DensityClust* and *GMMs*, *DBSCAN* requires an initial estimate of cluster cardinality. Here, we estimated this number based on single cell size and tested with the function *dbscan* in MATLAB. The results show that it is challenging for *DBSCAN* to correctly identify different cells in densely packed biofilm images.

Recursive Ncut: Recursive Ncut is similar to the *m-LCuts* workflow, but here only the similarity on spatial location is considered in calculating the adjacency matrix, *i.e.*, only Eq. (3.3) is kept. All the parameter settings and stopping criterion remain unchanged. The test of recursive Ncut indicates the necessity of adding collinearity feature for classifying collinearly structured data.

Advantages and limitations: *m-LCuts* exhibits superior performance over all other methods

	CA	SGA	SBF1
Region merging[49]	0.721 ± 0.222	0.847 ± 0.073	0.866 ± 0.064
GMMs[13]	0.760 ± 0.183	0.815 ± 0.108	0.834 ± 0.097
DBSCAN[97]	0.571 ± 0.222	0.766 ± 0.113	0.786 ± 0.109
DensityClust[89]	0.615 ± 0.237	0.718 ± 0.095	0.739 ± 0.089
Recursive Ncut[99]	0.625 ± 0.177	0.762 ± 0.087	0.789 ± 0.077
m-LCuts	**0.812** ± 0.145	**0.888** ± 0.082	**0.900** ± 0.074
Baseline	0.403 ± 0.159	0.676 ± 0.075	0.714 ± 0.073

Table 3.2: Quantitative comparison of performance on combining components in 125 over-segmented masks, where the IoU threshold for CA is 0.4. *Baseline* compares the initial segmentation results in a mask to ground truth for the ROI.

at most IoU values (**Fig. 3.16**). The only exception is the average performance of *GMMs* when IoU $\leq$ 0.3, which is slightly better than *m-LCuts*. This is due to the fact that *GMMs* will not produce false positives as k is always less or equal to the number of cells. It is worth mentioning that only *m-LCuts* and *GMMs* meet a reasonable level of precision, say 0.7, for cell counting purposes. At IoU $\geq$ 0.4, *m-LCuts* beats *GMMs* and all other methods by 20% in sensitivity. In addition, *m-LCuts* can better preserve the desired cell shape with respect to SGA and SBF1 (**Tabel 3.2**). When considering both global performance (cell counting accuracy) and local fidelity (morphology of the classified cluster), *m-LCuts* is superior to all the other methods. During the experiment, we noticed that the initial segmentation result will affect the final performance. This is because false detected or missing cell regions in the initial masks can change the collinearity features embedded in the supernodes. Some of these cases are evident in **Fig. 3.16**. An analysis of these cases will be discussed in the following section.

3.5.7 EXPERIMENTS ON REAL DATASETS

In the last two experiments, we tested the performance of *m-LCuts* for collinear data clustering with different types of input data, including point-based and segment-based data, with the goal of addressing different post-processing needs. Here, we explore how *m-LCuts* functions when applied to a real dataset with uncategorized post-processing needs. However, a problem for conducting systematic evaluation of such experiments arises from the lack of 3D ground truth. Manual annotation is the current approach to generating ground truth, but such a process is very difficult to perform in 3D. In this sense, only 2D manual annotations were available in [137]. These manual annotations are also error-prone and inconsistent between different annotators as may be observed in the rough object boundaries as shown in **Fig. 3.18** second column. In order to demonstrate the ability of *m-LCuts* in dealing with real cases, we will validate the method using cropped views of the original real image and compare to manual annotation results by four different annotators for reference. Of course, the performance of *m-LCuts* is not necessarily determined by the original datasets but by the initial segmentation outputs (masks). An additional experiment is discussed in Sec. 3.6.1.

Automatic mode detection: To determine the mode for post-processing the current mask, volume checks and concavity checks are employed, which are based on the estimated size and shape of the cells. For the volume check, if the volume of a connected component in a mask is larger than the upper limit of a regular bacterial size, it is likely that the cell requires additional *cuts* in *under-segmented mode*; otherwise, if the size is lower than the lower limit for a typical cell size, it will be treated with the *over-segmented mode*. The concavity check is the same as described in **Fig.** 3.12 (c), with an additional guarantee to include the components that are within the regular cell size.

Mixture of two different cases in one mask: In many real image scenarios, the sources of problems in the initial segmentation results might be unknown. Two different situations, under- or over-segmentation, may exist in the same mask. Here, we demonstrate a possible solution. Each case, the under- or over-segmented cases, is still treated separately via *m-LCuts*, but in a sequential manner. Given an initial segmented result, the volume and concavity checks are first performed to find all under-segmented components. If such cases exist, *m-LCuts* will run in the under-segment mode. Next, the volume and concavity tests are computed again to find the over-

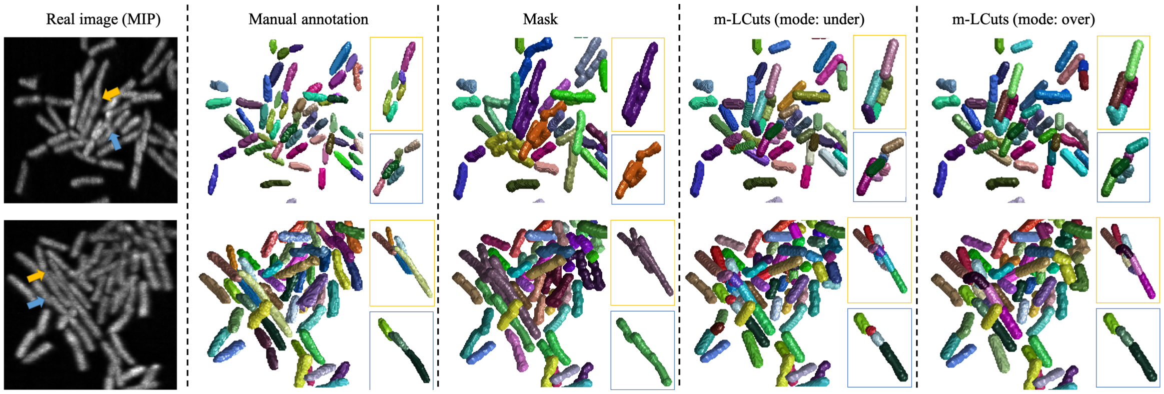


Figure 3.18: Evaluation using real data. Maximum intensity projection (MIP) along the *z-axis* of 3D real biofilm images are shown in the first column. The experimental signal-to-background ratio of the two images are 2.2 and 1.8 referring to [137]. The other images are 3D annotated/segmented volumes with a top view of *z-axis*. Yellow and blue arrows in the first column are examples that are highlighted for comparison in the other columns. The masks are provided by CNN only results from [137]. Then, we run the two modes of *m-LCuts* sequentially to post-process the initial segmentation results. Quantitatively, the counting accuracy (CA) scores of columns three to five, compared with manual annotations, are 0.34, 0.49, 0.50 for the first row, and 0.52, 0.61, 0.68 for the second row.

segmented ones for recombination using *m-LCuts* in the over-segment mode. In this mode, the neighboring segments of the over-segmented fragments will also be included in the graph for the consideration of recombination, even if they pass the mode checks.

Evaluation on real biofilm dataset: Experiments on real dataset are presented in **Fig. 3.18** with quantitative evaluation on counting accuracy. The experiments were conducted automatically with the mode detection mentioned earlier under the hypothesis that there is mixture of cases in the initial segmentation results. We ran under- and over- segmented modes sequentially as shown in the fourth and fifth columns in **Fig. 3.18**. Although there are not many mixed cases in the cropped views of CNN only (u-net based) initial segmentation results from [137], the ability of *m-LCuts* to reconnect the remaining over-segmented components after splitting under-segmented clusters can still be observed in **Fig. 3.18**. Final individual segments that are still less than the minimum cell size are eliminated during volume reconstruction in the second mode, because such segments are usually initial segmentation errors or from the touching cell connections. At least 16% improvement in counting accuracy (CA) was achieved after the two-step post-processing, at an IoU thresholding value of 0.4.

The previous methods for quantitative performance comparison are recurrent DT, GMMs, and region merging, as they are the sub-optimal under- or over-segmentation post-processing methods in sections 3.5.4 and 3.5.6. The clusters of interest are found by volume check. The post-processed CA values with recurrent DT to fix under-segmentation errors are 0.53 and 0.58 for the two real data as shown in **Fig. 3.18**. We executed the GMMs and region merging sequentially to solve under- and over-segmentation problems as this post-processing combination was applied in[49]. The updated CA values respectively after the two algorithms are 0.50, 0.47 for the first dataset, and 0.63, 0.65 for the second dataset. We also attempted to test over-segmented mode performance individually with all the methods, but there is no improvement as the under-segmented error is the dominant problem in the testing with real data. On the whole, the final *m-LCuts* outputs showed quantitative improvements over the other methods. Although sometimes the CA values after recurrent DT and GMMs are slightly higher, they usually detect less true positive cells than *m-LCuts* does. Moreover, manual annotation is difficult and error-prone for the 3D data, for example, it is difficult to identify single bacterium by human eyes in the regions highlighted with orange and blue arrows in **Fig. 3.18**.

Validation of sequential workflow on simulated dataset: The performance of *m-LCuts* is not related to the original image type, i.e., simulated or real, as the inputs of *m-LCuts* are binary images (masks). Owing to the limited amount of ground truth for real 3D data, the proposed sequential workflow is additionally verified on ten simulated images pre-processed with u-net and watershed methods. Half of these examples are dominated by under-segmentation errors, while the other half are dominated with over-segmentation errors. There are approximately 100 to 200 ground truth cells in each dataset. The quantitative cell counting performances of the current sequential *m-LCuts* and the combined post-processing method in[49] are listed in **TABLE 3.3**. Similar to the above observations for real data, *m-LCuts* presents the optimal performance in the majority of cases. After Step 1, *m-LCuts* sometimes cannot exceed the performance of

Data	Initial	1- GMMs	2- Region merging	1- m-LCuts	2- m-LCuts
UM1	0.67	0.76	0.72	0.73	**0.74**
UM2	0.43	0.63	0.62	0.69	**0.70**
UM3	0.53	0.65	0.64	0.73	**0.75**
UM4	0.57	0.70	0.68	0.77	**0.77**
UM5	0.56	0.74	0.72	0.79	**0.78**
OM1	0.67	0.71	0.71	0.71	**0.72**
OM2	0.63	0.64	0.66	0.65	**0.67**
OM3	0.65	0.68	**0.67**	0.69	0.65
OM4	0.68	0.68	0.68	0.70	**0.73**
OM5	0.83	0.84	0.82	0.83	**0.85**

Table 3.3: Comparison of the sequential workflow on simulated dataset. The simulated image dataset is described in section 3.5.1. UM denotes under-segmentation error dominated data and OM stands for over-segmentation error dominated data. The combination of GMMs and region merging is the two-step post-processing workflow used in[49]. The best final (after two steps) outputs are marked in bold and the best first step outputs are underlined.

GMMs as *m-LCuts* may detect extra line segments at the connections of cells, but these errors can be filtered out in Step 2.

To sum up, the above validations on both real dataset and simulated dataset demonstrate the effectiveness and generality of *m-LCuts*.

3.6 FACTOR ANALYSIS OF M-LCUTS

In this section, we conducted experiments to analyze the factors that can affect the performance of *m-LCuts*. In general, our analyses show that majorly the cell counting accuracy is related to the initial segmentation error. Initial segmentation errors are particularly problematic, when the masks do not possess clear collinearity, as is the case for heavily over-segmented or noisy masks. Therefore, extensive experiments were carried out to assess this input-output relationship. The results are discussed below together with a validation of the selected parameters.

3.6.1 INFLUENCE OF INITIAL SEGMENTATION RESULT

To produce under- and over-segmented masks, U-Net and seeded watershed were utilized to obtain the binarized initial segmentation result. The influence of initial segmentation result is analyzed in terms of initial segmentation error for each ROI. For over-segmented masks, this error is averaged over all the cells in the ROI by comparing original ground truth (GT^{v}) of layouts with corresponding masked ground truth (GT^{ROI}). The initial segmentation error per cell, Err_c, is calculated as 1 minus voxel-level precision, which is:

$$Err_c = 1 - \frac{||GT_c^{v} \cap GT_c^{ROI}||_0}{||GT_c^{ROI}||_0}$$

This is to analyze how many voxels in the segmented regions (mask) are not belonging to the original ground truth.

In **Fig. 3.19**, a linear regression can be estimated between *m-LCuts* performance and initial segmentation error. When the masked area includes extra large segments (noisy clutter) in multiple cell regions, the corresponding *m-LCuts* performance degraded because of the missing collinearity feature. Similar evaluation was also repeated on the ratio of missing GT^{v} voxels in the segmented result by changing voxel-level precision to recall. No substantial linear relationship was observed with this analysis, because there are not many missing component situations in the data. However, we still noticed that missing components can split the cell regions with big gaps, in which case, *m-LCuts* would not connect them.

For under-segmented masks, above evaluations per ROI were performed by comparing the whole ROI GT regions with under-segmentation result. Since rare cases of missing components or extra noisy clutter were found by thresholding the u-net result with a confidence of 0.5, the under-segmented masks can still maintain the collinear structure after medial axis extraction. Thus, averaged cell counting accuracy above 0.95 was achieved with $IoU < 0.8$ (**Fig. 3.15**). It is notable that the radius-constrained medial axis extraction method proposed along with *m-LCuts* can take into account radius of an initial segment as small as $r_{min} = 1$. This action prevents point cloud data from breaking in the middle of a cell region. Additionally, it is beneficial for identifying central axes of parallel cells by setting the upper limit of inscribed radius (r_{max}).

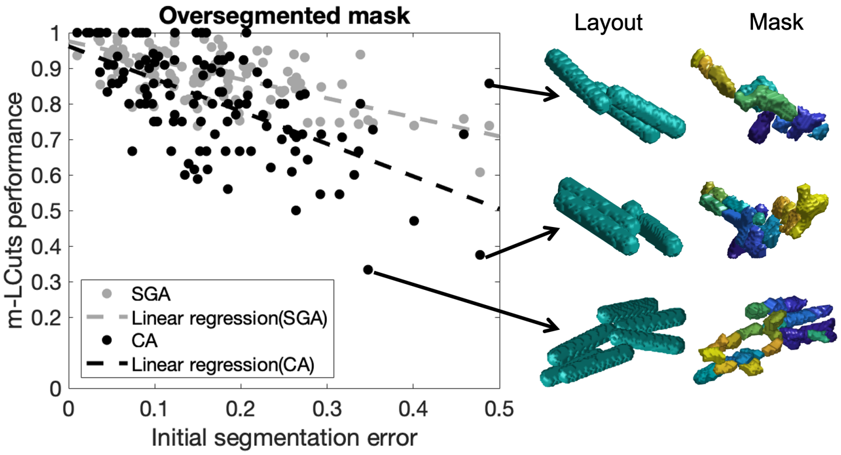


Figure 3.19: Analysis on the effect of initial segmentation results. Scatter plots of single-cell node grouping accuracy (SGA) and cell counting accuracy (CA) versus initial segmentation error are shown on the left panel. A linear relationship between *m-LCuts* performance and initial segmentation error is estimated. On the right panel, three masks are shown. The noisy clutter and missing components on these masks affect the collinearity of segments compared to the original ground truth of the layout. Hence, cell counting performance is affected, especially when multiple cell regions were affected by the errors.

3.6.2 PARAMETER VALIDATION

Twenty randomly selected under- or over-segmented clusters participate in validation for each case. Grid search is conducted on the major two adjustable hyper-parameters, *sizeLimit* and *distLimit*, while the other parameter σ_D can be estimated from these two values in the algorithm (see **Algorithm 1**). σ_T is related to the user definition for collinearity. In our case, σ_T equals to 0.2 while the relative angle (θ_{ij}) between two vectors is around 36 degrees. There are also some other parameters inside of the algorithm, such as the maximum hop-level number, minimum neighboring segment contacting area, and the cell concavity ratio limit in stopping criterion. The values of the former three parameters can be estimated by different applications, e.g., chosen as 5, 3π and 1.8 respectively based on bacterial cell morphology.

In the scenario of bacterial cells in this paper, *sizeLimit*, determined by r, is selected based on cell size information, where a typical cell in our experiment has a radius between $3 - 5$ pixels and length between $8 - 60$ pixels. The *distLimit* is chosen based on both cell size and mask

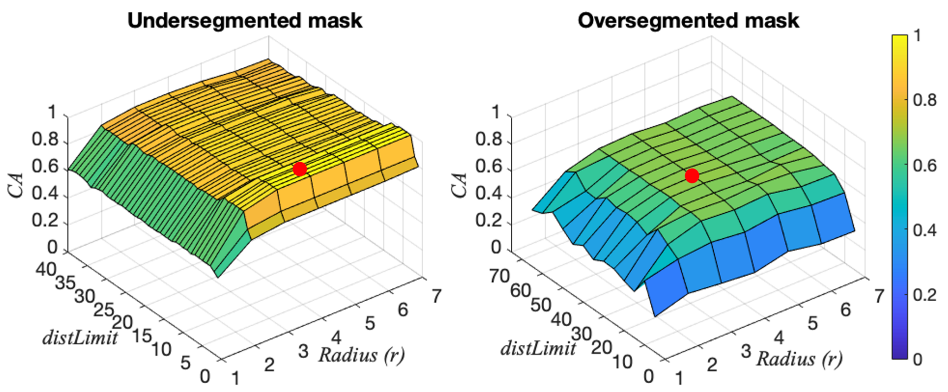


Figure 3.20: Parameter validation via grid search. Counting accuracies (*CA*) are averaged over 20 randomly selected data experiments. For both masks, radius $r \in [1,7]$ pixels with a step size of 1 pixel. $distLimit \in [1,40]$ with a step size of 1 pixel for under-segmented mask and $distLimit \in [10,70]$ with a step size of 5 pixels for the other mask. Red markers are the positions of selected parameters in our experiment.

conditions. It reflects an allowance of distance between neighboring nodes in the same group. For under-segmented masks, nodes that are within the radius of a cell ($distLimit = r$) are more likely to be in the same group; while for the other case, *distLimit* was selected by evaluating the supernode-to-supernode distance in the data. By investigation, the mean supernode-to-supernode distance lies in the range of 20 to 40 pixels for all the datasets. Small variations of CA with previously mentioned parameter ranges are observed in **Fig. 3.20** for both masks.

3.6.3 RUNTIME

The time complexity of the core section in *m-LCuts* can be estimated as $\mathcal{O}(n^4)$, where n is the number of nodes/supernodes. Here, the worst case of recursion depth is n for bi-partition problems. However, *m-LCuts* will actually never reach this worst case especially for the under-segmented dataset, because the stopping criterion reduces the recursion depth by checking the size, collinearity and concavity conditions at each branch. Additionally, the implementation of graph cut at each level involves solving the eigenvalue problem, which has a complexity of $\mathcal{O}(n^3)$. *m-LCuts* takes the advantage of constructing a sparse graph with key points, *i.e.*, medial axes for under-segmented mask and segment centers for over-segmented mask. Thus, it can effectively reduce the computational expense.

The experiments were tested on two laptops: (1) Windows 8.1 pro, Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz, RAM 16GB, and (2) macOS Big Sur, 3.1 GHz Dual-Core Intel Core i5, RAM 16GB. The reference running time on Windows 8.1 pro for correcting over-segmented masks in **Fig. 3.16** first and last row are 18.4 seconds and 95.8 seconds, respectively. For further splitting under-segmented masks, such as the mask in the last row of **Fig. 3.14**, the runtime is 7.7 seconds. There are still several ways to optimize the proposed runtime in the future, such as modifying the bi-partition workflow to a k-class classification problem, or increasing the efficiency of graph cut by substituting the computation of eigenvalues in the solution of optimized graph cut. It is worth mentioning that the k-class strategy may need an estimated number of groups, *k*, while the recursive bi-partition strategy in the current *m-LCuts* does not require this parameter beforehand.

3.6.4 Other factors

A complete list of the factors that were tested is shown in **Table 3.4**, typically for the over-segmented mask mode. Three major aspects are considered for factor analysis: parameters, image environment, and connectivity/layouts of the segmented connected components. In image environment, SBR stands for signal-to-background ratio, which accounts for the contrast of intensity in the image. SNR stands for signal-to-noise ratio, which can also be alternatively represented by the coefficient of variation. Initial segmentation error has been explained above. For connectivity/layout of segments, the number of cells is just the ground truth cell number in the masked cluster. Cell density is calculated by dividing the volume of cell regions by the whole cluster/mask volume. Average node-to-node distance evaluates the average distance from the centroids of over-segmented supernodes to each other; while the average cell-to-cell distance is the average distance between the centroids of each two ground truth cell regions. The last one, contact surface ratio, considers how many area on the surface of each cell is contacting with a neighboring cell.

Parameters	Image environment	Connectivity/layout of segments
Algorithm input parameters	SBR/contrast	Number of cells
	SNR/ coefficient of variation	Cell density
	Initial segmentation error	Average node-to-node distance
		Average cell-to-cell distance
		Contact surface ratio

Table 3.4: A complete list of factors that were analyzed to test the performance of *m-LCuts*, typically for the mode with over-segmented masks. Based on our analyses, the cell counting accuracy is mainly affected by the initial segmentation error, underlined in the table.

There were no typical relationship between the performance of *m-LCuts* and other factors at the current experimental set-up as described in section 3.5.1, except for the two discussed in the above section, *i.e.,* initial segmentation error and parameter validation. To analyze these factors, we plot the performance (CA and SGA score as y) and the factors in **Table 3.4** as x. Some example plots are shown in **Fig. 3.21**. There is a linear drop of accuracy when SBR ≤ 1.87, but that is beyond the current dataset (SBR=2.2) in the experiment section of m-LCuts.

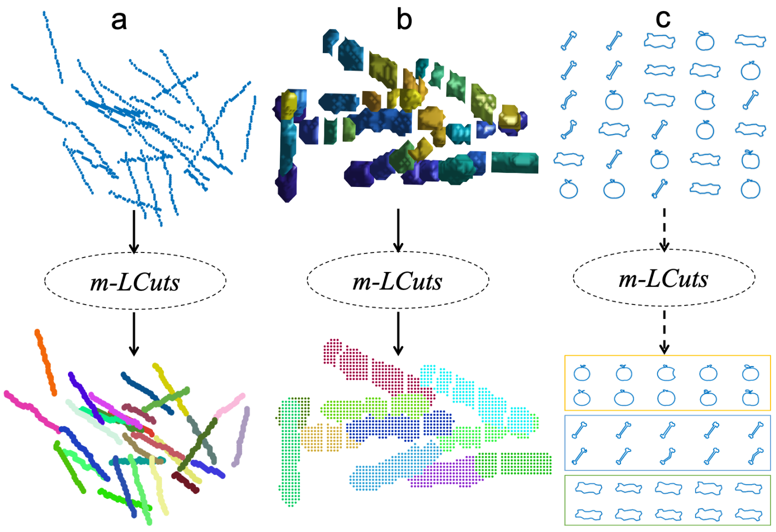


Figure 3.22: Applications of *m-LCuts* to find collinearly featured clusters on masked datasets. In this paper, the theory of *m-LCuts* and its application in clustering (**a**) points and (**b**) over-segmented components are presented. It also has the flexibility to be extended to other scenarios, such as shape classification in an elastic shape space (a preliminary result is shown in **c**.)

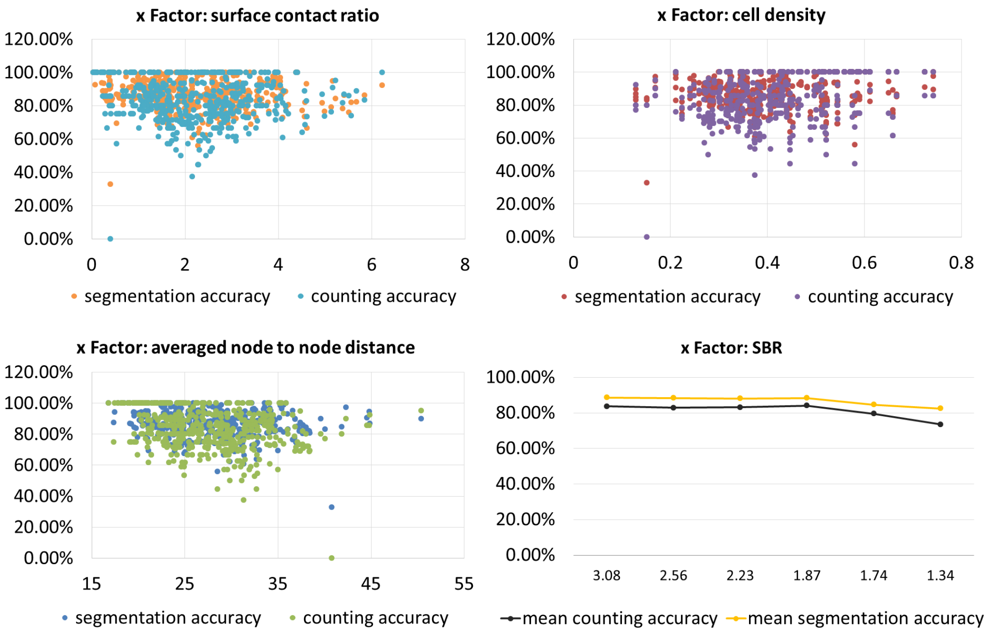


Figure 3.21: Example plots for m-LCuts factor analysis. The *x* axes for each plot are the factors denoted in the titles, while the *y* axes are accuracy in terms of single-cell segmentation/grouping accuracy and cell counting accuracy. The data for the scattor plots contain the results with all six image SBR conditions. The accuracy at each SBR level in the SBR plot is averaged over 125 regions of interest experiments as the ROI also used in the experimental part for the m-LCuts.

3.7 EXTENSION ON SHAPE CLASSIFICATION

The proposed method, *m-LCuts*, is capable of detecting and separating collinearly-featured objects in 3D segmentation masks, as demonstrated in the previous sections. Compared to the state of the art, *m-LCuts* achieves the best performance and does not require manual specification of the number of different objects in the data. *m-LCuts* is not only a post-processing solution for biofilm segmentation, but also a general collinear data clustering approach. It accepts different types of input, such as points and segments. *m-LCuts* can also be extended to other applications where collinearly-featured data are presented. A preliminary experiment on shape classification was performed (**Fig. 3.22c**).

A current workflow of *m-LCuts* for shape classification is shown in **Fig. 3.23**. The mask is the

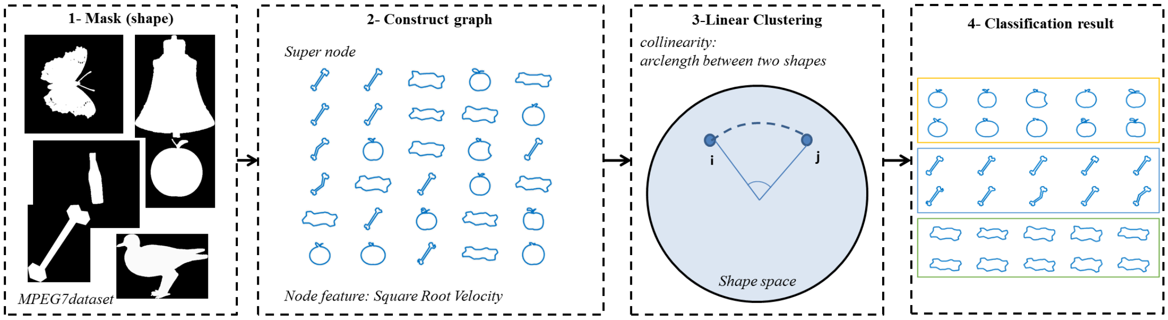


Figure 3.23: Preliminary Procedure of applying *m-LCuts* for shape classification.

contour of binarized objects in the image (we used the MPEG7 CE-Shape-1 Part B dataset[26]). Then, each contour is a supernode in *m-LCuts* where the node collinearity feature is the arc length between two shapes, defined as geodesic distance, in the square-root velocity transformed elastic shape space[109]. Details about how we apply the shape space in our bacterial problem will be discussed in detail in Chapter 5. We also investigated features such as the relative angle between the geodesic shooting velocity from one shape to the other shape. The classification result was not affected significantly under the current shape clustering requirement. Currently, the stopping criterion is the number of shapes in the group. This criterion can be further investigated by evaluating the linearity variance of nodes as presented in *m-LCuts*, but in terms of shape.

3.8 SUMMARY

In this chapter, we presented the extension of *LCuts* to *m-LCuts*. In general, *m-LCuts* yields a generalized and unified data clustering solution with input data either being points or segments to address different post-processing concerns. m-LCuts reveals strengths in post-processing unsolved cases in biofilm segmentation, as evidenced by both higher cell counting and higher single-cell segmentation accuracies compared to the current state of the art and the previous versions of LCuts. The advances of m-LCuts include the following aspects:

- **Generalization:** The whole algorithm is adaptable to different kinds of input data, such as points, segments, or even shapes;
- **Clean key point extraction:** m-LCuts constructs the graph using radius-constrained medial axis extraction or super node generation from masks, which provides cleaner key points that can preserve the collinearity of data in 3D. Here, a *cleaner key point* means that the collinearity features are less affected by the noisy data points outside the central axes of cells;
- **Stable collinearity extraction in 3D:** Outlier-removed graphs are constructed that replaced the original majority voting process to better describe collinearity features in 3D, which is observed in both visual and quantitative experiments;
- **Unified recursive clustering:** The new similarity measure is defined with additional linearity penalty and new stopping criterion with gap detection to separate different cells in a line, which can fulfill the needs for all the inputs (points or segments) as a unified solution.
- **Improved segmentation performance after post-processing:** In both experiments with initial segmentation methods that incorporating convolutional neural networks and applying general watershed-based segmentation methods, m-LCuts is observed to effectively improve the performance by at least 16% in cell counting accuracy after post-processing in the real image experiments.

Never stop dreaming, never stop believing, never give up,
never stop trying, and never stop learning.
Roy T. Bennett, The Light in the Heart

4

Learning and evaluating biofilm images

When incorporating our proposed post-processing methods with data-driven cell segmentation strategies (*e.g.,* machine learning-based approaches), the performance of experiments on real LLSM datasets is typically inferior to the performance on simulated datasets. Furthermore, the real data experiments are often difficult to evaluate. One possible reason comes from the lack of ground truth annotations of the 3D images as cells are densely-packed and randomly-oriented in the 3D space. More challenges arise because of the difficulty in reproducing the realistic experimental image conditions in the simulated dataset for training those deep learning-derived models. As a consequence, further bioimformatic analysis of the realistic biofilms can be affected. Therefore, we investigate generative adversarial networks (GANs) to synthetically generate dataset that can mimic the realistic conditions in the LLSM 3D biofilm images[123]. By proposing a stochastic synthetic dataset quality assessment (SSQA)[123], we will be able to expand the training dataset for biofilm image segmentation with reliable images and annotations.

4.1 BACKGROUND ON IMAGE SYNTHESIS

The advent of deep learning brings opportunities to solve challenging problems in a variety of fields, including single-cell analysis. However, such data-driven approach requires a large quantity of labeled training data to achieve solid solutions[101]. The current gold standard for labeling biological images such as those of bacterial biofilms is manual annotation, which can be extremely time-consuming, especially for 3D data, and also inconsistent among different annotators. The process of manual annotation in microscopic fluorescent images is hard, because fluorescence intensity is often not uniform in cells. When cells are closely touching each other, the gaps between neighboring cells are often smaller than the diffraction-limited resolution (200-300 nm) and therefore hard to identify. It is even more difficult, or sometimes nearly impossible, to annotate non-spherical cells densely packed in 3D (**Fig. 4.3** first column), like bacterial cells. Even with manual annotation, it is difficult to delineate touching cells and to provide correspondence between different z-slices of the same cell. Therefore, an automatic data augmentation method is important for deep learning tasks to obtain sufficient training dataset with objective ground truth annotations.

There are three major strategies to expand the size of the training dataset for applying deep learning on the single-cell analysis task: performing classic transformation (e.g., scaling, translation, rotation, interpolation[1,91]) on limited manually annotated data, simulating the volumes using optical and biological knowledge[137,68,112], and generating synthetic datasets using generative adversarial networks (GANs)[33,35]. The first strategy, using the limited manually labeled data, is very error-prone and time-consuming, especially in the setting of 3D biofilms as mentioned above. Therefore, the ideal goal is to automatically expand the datasets with accessible ground truth labels that do not incorporate human subjective opinions.

The second approach, data simulation based on optical and biological knowledge, is a model-based data augmentation strategy. There are three general steps in prototyping microscopic images[137]. First, modeling the spatial structure of the objects in the image, e.g. spherocylindrical rods for bacterial cells. In[137], a computational modeling software, CellModeller[93], is used to synthesize the physical splitting and growing nature of biofilms in 3D, which generates binary images where bacterial cell volumes are foreground on an otherwise dark background. Then, virtual fluorophores are placed in the cell volumes to simulate the fluorescence emission[68], *i.e.*, the fluorophores give rise to overlapping focused and defocused point spread functions on the detector. Finally, read noise and photon shot noise of the specific microscope are convolved/added to mimic the real conditions in the images. Promising instance segmentation results (categorizing each cell region as an individual segment) were achieved with simulated bacterial biofilm data in[137], but observable differences between simulated and real data still exist, because the simulated images cannot take into account uncalibrated image aberrations and illumination/emission heterogeneity. Perhaps more importantly, CellModeller only simulates spherocylindrical cells, such that curved or bent bacterial cells are never encountered in the training step. Thus, data augmentation method that can learn the realistic experimental microscopic image conditions is needed for in-depth analysis of biological/medical images[101].

The third strategy, using generative adversarial networks (GANs), is a recent more popular solution because it can generate images that reproduce realistic scenarios in experimental datasets. GANs attempt to maximize the probability in discriminating between real and fake images, as well as minimize the difference between real and generated images[44]. There are many modalities in the literature that learn the translation between images, such as *pix2pix* for learning paired natural images[54], *CycleGAN* for unpaired image-to-image scenario[140], *DCGAN* on CT images[38], and *Ising-GAN*[29] on microscopic images. Among these, *CycleGAN* and its variations[39,95,139] do not require the input and output images to be exactly paired. The paired image-to-image translation means the input and output domains are aligned to each other, for example translating a binary image to its corresponding fluorescence image. In another case, the unpaired task translates unrelated images from different domains where the ground truth annotation (input image) for the corresponding output microscopic images does not exist.

CycleGAN generates images via mapping the distribution in the input image domain to another unpaired output image domain. Most importantly, such a model can bring back the input image from the reverse direction in the network by a cycle consistency constraint[140]. This technique was tested for a cell label-to-image translation task in[39], but spatial drifting of cell location was observed within the generated images. Here, a label-to-image translation is defined as mapping the distribution in the labels to another distribution in the image. The authors in[39] added spatial consistency loss to reduce the drift of generated cells from their labels. This work successfully generated 3D synthetic nuclei cells images in a rat kidney, and motivates the learning task in this paper. While works like[39] and[95] perform training on 2D slices in the 3D images to generate synthetic 3D stacks, they cannot provide sufficient axial (along *z-axis*) signal continuity in the images for our bacterial biofilm data, because there is not a large size of z stack in the biofilms to train the 2D networks as the others did. There are also works extending 2D GANs to 3D[2,139], but they do not have the learning needs to transfer between unmatched labels and images. Besides, our biofilm data presents different challenges than the data in the literature in the differing cell size, anisotropic cell shape, and different noise levels, so an adapted generation workflow is desired to fulfill the needs for the biofilm learning and augmentation task.

4.2 IMAGE SYNTHESIS BY LEARNING LLSM IMAGES WITH 3D CYCLIC GAN

The image synthesis pipeline proposed in this dissertation exploits the ideas of unpaired image translation with cycle consistent GANs[140,39], and extends the networks to 3D for learning the microscopic bacterial biofilm data. Additionally, different GAN loss functions are compared to achieve optimal augmented biofilm data in 3D when evaluating multiple image quality assessment metrics. Since the ground truth annotation is not available in most of the unpaired image-to-image translation tasks, a stochastic synthetic data quality assessment scheme (SSQA) is also proposed in the next section. Our solution, *3D Cyclic GAN*[123] along with SSQA, will answer the following questions:

- How to learn the axial correspondence in terms of image signal continuity along the *z axis* for 3D images compared to those generated by the 2D GANs?
- What is the most appropriate loss function that provides both faithful distribution and spatial consistency in the 3D biofilm images?
- How to evaluate the quality of generated synthetic dataset without corresponding ground truth?

Correspondingly, a workflow that summarizes the learning and evaluation pipeline in this paper is shown in **Fig. 4.1**.

4.2.1 DATASET

The 3D experimental dataset of biofilm is obtained by lattice light-sheet microscopy (LLSM) built in the Cell Imaging at Nanoscale lab at the University of Virginia[137]. LLSM is able to look into the dense aggregations of cells *in vivo* because of its lower phototoxicity and higher spatial and axial resolution compared to that of traditional confocal microscopy[40]. *E.coli* bacterial cells are used in this experiment with cytosolic expression of green fluorescent protein. There are 300

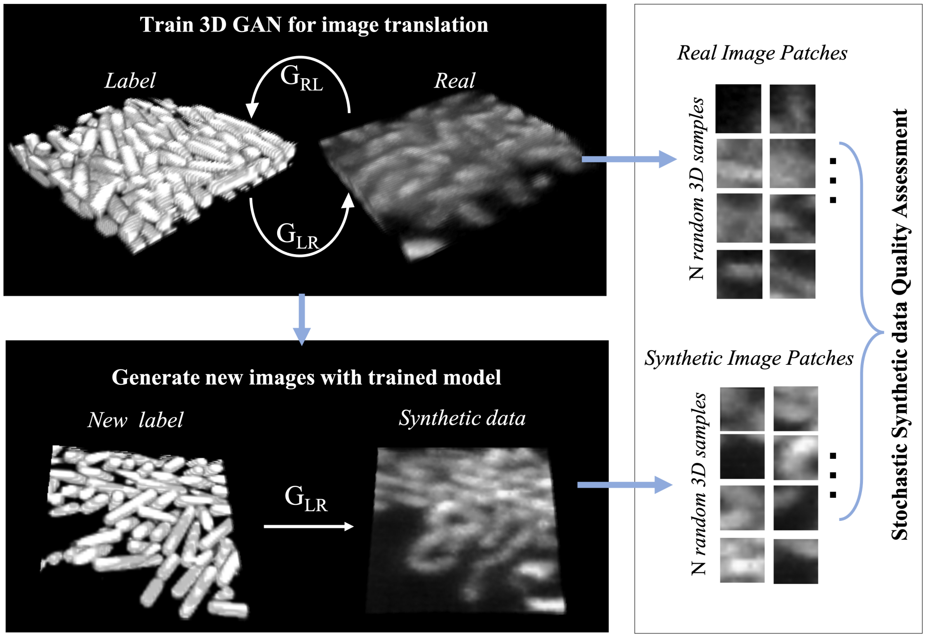


Figure 4.1: Pipeline for learning and evaluating the LLSM 3D bacterial biofilm images. All the images shown are 3D rendered in ImageJ. **Step 1:** learning the translation from labels to real images using 3D cyclic GANs, which contains two generators (G_{LR}, G_{RL}), and two discriminators (not shown in the figure). **Step 2:** the learned translation from labels to real images is used to generate synthetic data. **Step 3:** evaluating the quality of synthetic data compared to real images.

3D images with size $128 \times 128 \times 8$ voxels for training. These training images contain three time points in biofilm development (**Fig. 4.3** first column). Each image has voxel size of 100 nm ×100 nm ×100 nm. All the real images are pre-processed with normalization and contrast enhancement that saturates the bottom 1% and the top 1% of all voxel values. The label set is generated by CellModeller[93] with local densities that approximately match the real image conditions (**Fig. 4.3** second column). These images in the label set will be treated as the ground truth of synthetic cell location.

4.2.2 3D GAN ARCHITECTURE

The basic framework of the learning module in this paper follows the cyclic generative adversarial networks (CycleGAN)[140], which consists of four parts: two generators and two discrimi-

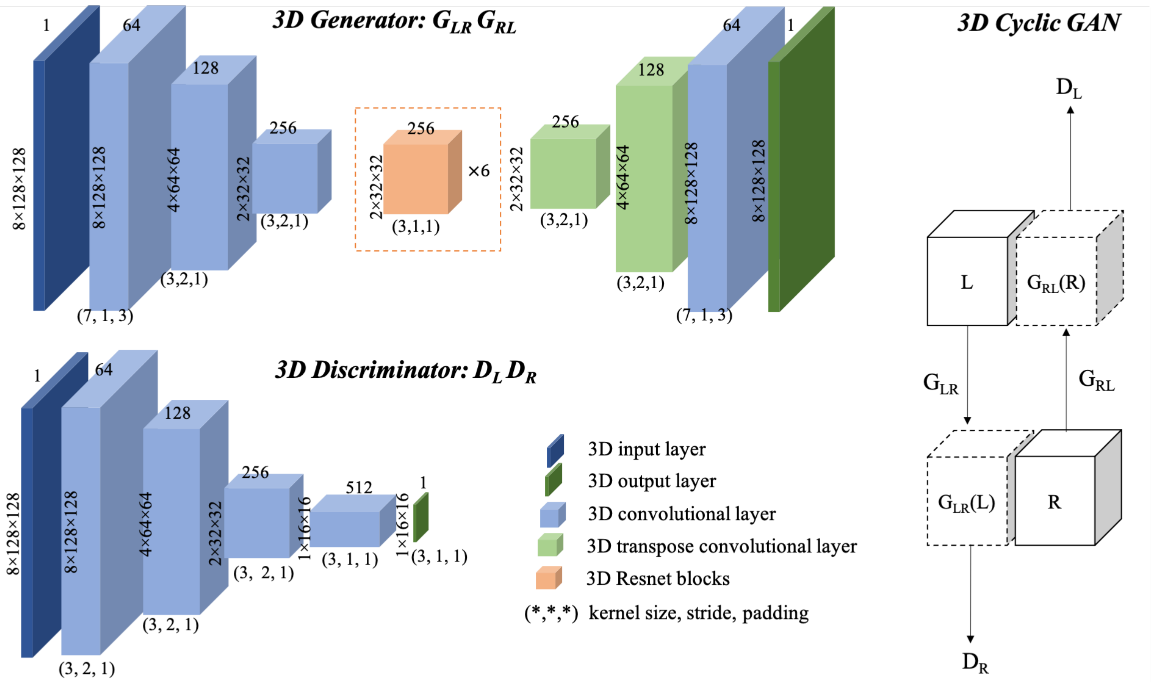


Figure 4.2: Architectures of networks in the 3D cyclic GAN for learning biofilm images, which replace all the 2D layers in [140] and [39] to 3D layers. The sizes of kernel, stride, and padding in the convolutional layers are cube-shaped with parameters indicated in the diagram.

nators (**Fig. 4.2**). The generators (G_{LR}, G_{RL}) attempt to generate synthetic LLSM images from a binary label set, while discriminators (D_L, D_R) try to distinguish generated synthetic data from the real data. We adopt the architectures of the downsampling/upsampling style generator with six residual blocks [50] at the bottleneck and patchGANs discriminator with three convolutional layers as used in [39], which presented promising results in 2D cases. In both networks, we change all the 2D layers to 3D ones. In addition, the original 70×70 2D patchGANs discriminator is changed to patch size $8\times65\times65$ (depth × height × width) for input of a biofilm image. Replication padding, instance normalization and the ReLu activation function are used in the generator with employment of the hyperbolic tangent function in the last layer, while the discriminator uses LeakyRelu(0.2) and batch normalization. Details about the input and output shapes of each layer implemented with PyTorch are shown in **Fig. 4.2**, where the number of channels, image tensor size, and other details of each convolutional layer are labeled in the diagram.

4.2.3 Loss function

To learn the mapping functions between labels ($l_i \in L$) and microscopic real images ($r_i \in R$), where $i \in [1, T]$ and T is the number of training data, the loss function in the original *CycleGAN* formation and several variations are tested for applying on biofilm data. Generally, the total loss function ($\mathcal{L}$) consists of three parts[140]:

$$\mathcal{L}(G_{LR}, G_{RL}, L, R) = \mathcal{L}_{\text{GAN}}(G_{LR}, D_R, L, R) + \mathcal{L}_{\text{GAN}}(G_{RL}, D_L, R, L) + \lambda \mathcal{L}_{\text{cyc}}(G_{LR}, G_{RL}) \quad (4.1)$$

where λ is a control parameter, which is set to 10 in the experiments such that all losses are on the same scale.

The first two GAN losses assess the mappings $L \rightarrow R$ and $R \rightarrow L$. These losses try to generate the similar images as the target output images, as well as maximize the performance of distinguishing the two classes (0 for generated, or 1 for original) by using D_R or D_L. The basic adversarial objective function, which was originally derived from log loss or binary cross-entropy in[140], is a min-max problem. However, as suggested in[75], such a regular GAN loss function with a sigmoid function in the last layer of the discriminators can cause the well-known vanishing gradients problem. Unstable results are observed both in the experiment in[140] and in our 3D biofilm situation. Thus, least-squares GAN loss function[75], which minimizes both generator $G_{LR \text{ or } RL}$ and discriminator $D_{L \text{ or } R}$, is more practically used. Consequently, the optimization of GAN loss yields:

$$\begin{aligned}
\min_{G_{LR}} \min_{D_R} \mathcal{L}_{\text{GAN}}(G_{LR}, D_R, R, L) = & \min_{G_{LR}} \frac{1}{T} \sum_{i=1}^{T} ||D_R(G_{LR}(l_i)) - 1||_2 \\
& + \min_{D_R} \frac{1}{T} \sum_{i=1}^{T} (||D_R(r_i) - 1||_2 + ||D_R(G_{LR}(l_i)) - 0||_2)
\end{aligned} \quad (4.2)$$

The last cycle consistency loss $\mathcal{L}_{\text{cyc}}$ tries to ensure that the cyclic translation process from one side to the other is able to bring back the original input image, *i.e.*, $G_{RL}(G_{LR}(l_i)) \approx l_i$. Mathematically, the cycle consistency loss aims to minimize the mean error between an input image and the cycled back image[140]. We generalize and modify the original loss formula as:

$$\begin{aligned} \mathcal{L}_{\text{cyc}}(G_{LR}, G_{RL}) =& \frac{1}{T}\sum_{i=1}^{T}(\alpha||G_{RL}(G_{LR}(l_i)) - l_i||_{p_1} \\ & + \beta||G_{LR}(G_{RL}(r_i)) - r_i||_{p_2}) \end{aligned} \tag{4.3}$$

where α and β are weighting factors. ℓ_1-norm, as $p_1 = p_2 = 1$ for both mapping directions, is used in[140] and many other single biomedical object translation applications (e.g. from fMRI to T1 maps[2]). However, the work on generating synthetic cell volumes from binary cell labels in[39] suggested that displacements in cell location was observed. Accordingly, the authors in[39] added a spatial consistency loss, which is an additional ℓ_2-norm that computes the pixel-wise mean-squared error between l_i and $H(G_{LR}(l_i))$. Here, H is the same structure as G_{LR}.

As the lack of paired data is particularly challenging in our case, and the noise levels and cell sizes differ significantly from those of the original data used in[39], simply utilizing the loss function does not accommodate our particular data augmentation problem. We aim to investigate different loss functions to determine which form provides both the best distribution and spatial consistency for biofilm data. The decision between choosing ℓ_1-norm and ℓ_2-norm in loss functions is always of critical interest. Although ℓ_1-norm is robust for data with outliers, ℓ_2-norm provides unique solutions. The output correspondence between labels and images is important for the data augmentation task for biofilms, as future researchers would like to use the augmented dataset with corresponding real ground truth labels for improving segmentation tasks. Thus, different combinations of p_1 and p_2 values, including the options listed above, are compared below. Particularly, we hold a hypothesis: changing $p_1 = 2$ and increasing the weighting of cycle mapping direction $L \rightarrow R \rightarrow L$ by a factor of $\alpha = 2$ can also provide spatial consistency. In this case, the need for training another network H in 3D will be reduced. Details about the comparison of different loss functions are included in the next section.

4.3 EVALUATING SYNTHETIC DATASET WITH SSQA

The training of 3D cyclic GANs for generating LLSM biofilm images follows the basic setup in [140], which uses Adam optimizer with batch size of 1 and learning rate of $2e^{-4}$. The learning rate decays linearly after half of the total epochs are completed.

Four loss functions are compared: the original *SpCycleGAN* [39] [*], the *SpCycleGAN* loss function applied in 3D GANs, the original *CycleGAN* extended to 3D, and the modified 3D cyclic GAN with unbalanced cycle consistency loss, as shown in **Fig. 4.3**. Other combinations of p_1 and p_2 values are also tested, but they did not provide reasonable outputs. *SpCycleGAN* was trained with all the 2D *z-sliced* images in the 3D stacks and the other networks are directly trained using 3D cyclic GANs with 300 training data. Each model is trained separately with one NVIDIA Titan v GPU, which takes about a day for training and few minutes for testing. The training time elapse for all the networks follows: *SpCycleGAN* $<$ *3D Cyclic GAN* (ℓ_{21}) $<$ *CycleGAN 3D* $<$ *SpCycleGAN 3D*. The testing, or image generation, is performed on another 3D dataset containing 300 images that have never appeared in the training set. Additionally, model-based simulated dataset and reference real images are also listed in **Fig. 4.3**, where all the images are one slice along *z-axis* in the 3D image volumes.

The goal of this section is to evaluate the GAN outputs, find the best loss function for the biofilm learning task, and potentially obtain the best synthetic biofilm dataset for future research.

---

[*] We thank Prof. Edward J. Delp and his laboratory at Purdue University for providing the code for DeepSynth (SpCycleGAN) in the work of this chapter.

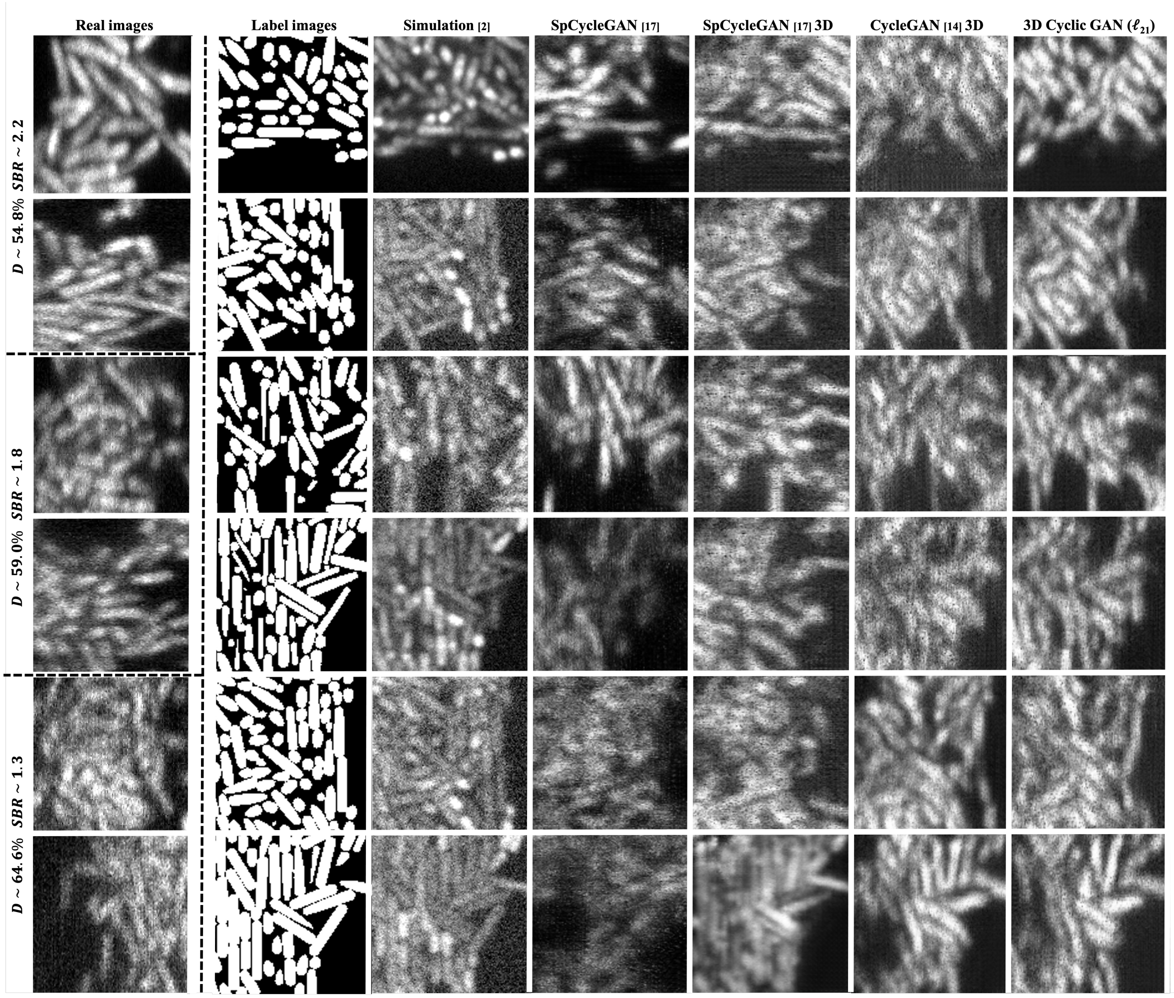


Figure 4.3: Examples for unpaired training dataset, and qualitative comparison of different augmented results. The images in each column are one slice in z*-axis* in the 3D volume. ***Real***: examples of real enhanced LLSM experimental 3D images, which contain three average biofilm local densities (D), 54.8%, 59.0%, and 64.6%, and three signal-to-background ratios (SBR), 2.2, 1.8, and 1.3[137]. ***Label***: example label images obtained by CellModeller with comparable local density conditions to the first column. The test label images for generating synthetic data do not exist in the label set for training. Please note, the *real* and *label* images are unpaired, as they do not have location correspondence to each other. The ***synthetic outputs*** for comparison are from model-based simulation[137], original *SpCycleGAN*[39], *SpCycleGAN* loss extended to 3D, *CycleGAN* loss[140] used in 3D, and our modified 3D cyclic GAN with $(p_1, p_2) = (2, 1)$ and $(\alpha, \beta) = (2, 1)$ in Eq. (3).

4.3.1 MOTIVATION FOR SYNTHETIC DATASET QUALITY ASSESSMENT

As mentioned earlier, we aim to study how well a GAN generated synthetic dataset creates a training set in which the images are similar to the real conditions with corresponding ground truth labels, as it is difficult to annotate single cell regions in experimental 3D biofilm images (see **Fig. 4.3** first column). The quantitative evaluation of the quality of a 3D synthetic dataset, without ground truth labels, is extremely challenging. For a paired image translation dataset, the ground truth images are used to compute scores such as pixel-wise mean-squared errors or class-wise Dice[107] scores. As a significant body of ground truth images does not exist, the quantitative quality evaluation metrics are limited, because there are not many referenceless metrics for 3D images. In[54], AMT perceptual studies were carried out to collect subjective feedback from human participants, but it is difficult to gather sufficient data for our particular experiment. In order to automatically assess and learn the conditions in 3D GAN outputs, this paper presents a comparison-based stochastic synthetic dataset quality assessment measure that evaluates relative intensity-wise quality of 2D/3D synthetic dataset compared to real dataset. Three other measures in the literature are also evaluated for comparison. These measures take into account different aspects of images: deep feature-based dataset similarity analysis with FID[52], distortion-based image quality evaluation with BRISQUE[78], and location correspondence evaluation with signal-to-background ratio (SBR) analysis.

4.3.2 SSQA

To evaluate the intensity-wise similarity between a generated synthetic dataset and the real LLSM biofilm dataset, a **S**tochastic patch-based **S**ynthetic dataset **Q**uality **A**ssessment (SSQA) method is proposed. When the two datasets for comparison are neither identical images nor corresponding volumes with different distortions, the regular fully-referenced image quality assessment (e.g. SSIM[126]) or comparison-based blind quality assessment methods (e.g. C-IQA[67]) cannot accommodate our application, because they rely on the structure and distortion within the whole image.

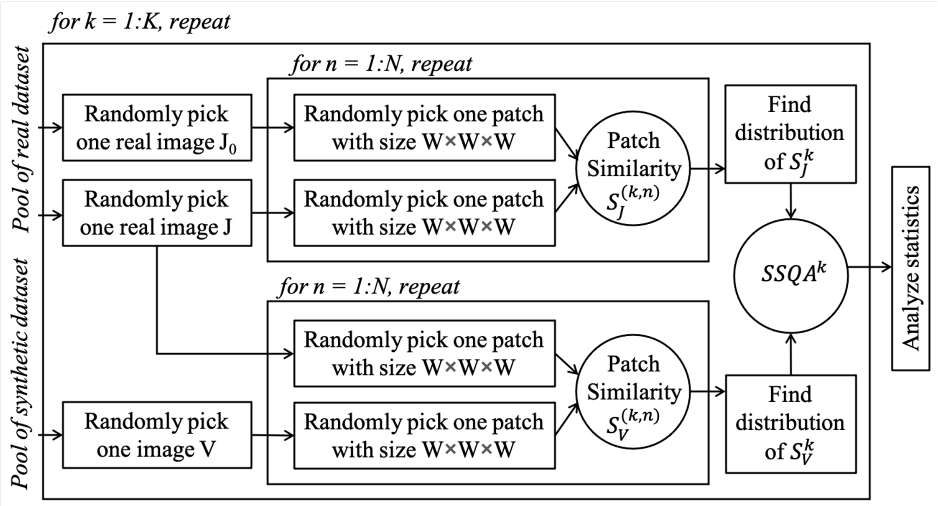


Figure 4.4: Flowchart to compute and analyze SSQA.

The proposed SSQA, as shown in **Fig. 4.4**, is a stochastic approach that takes random patches from random images for quality evaluation and statistical observation analysis. SSQA contains four stages. In the first stage, three images are randomly picked from the pool of two sets, J_0 and J from the real and V from the synthetic dataset. This stage will repeat K times and an SSQA score will be computed at each time. At the second stage, the comparison-based quality similarity measure $S_{Data}^{(k,n)}$ is calculated between two randomly cropped patches with size $W \times W \times W$ from two pairs of images: $J_0 \leftrightarrow J$ and $J \leftrightarrow V$. Here, the similarity score obtained from $J_0 \leftrightarrow J$ pair is treated as a reference intra-class (real image) comparison. $S_{Data}^{(k,n)}$ denotes the similarity score between the compared images $Data = \{J, V\}$ in the dataset and the images J in the real dataset at the n^{th} patch selection and the k^{th} image comparison. This score is formulated based on the luminance and contrast indices from [126]:

$$S_{Data}^{(k,n)} = \frac{2\mu_{data} \cdot \mu_j + C_1}{\mu_{data}^2 + \mu_j^2 + C_1} \cdot \frac{2\sigma_{data} \cdot \sigma_j + C_2}{\sigma_{data}^2 + \sigma_j^2 + C_2} \tag{4.4}$$

where μ_{data} and μ_j, and σ_{data} and σ_j are the means and standard deviations extracted from patches *data* and *j*, respectively. Patches *data* and *j* are randomly picked from *Data* and *J*. C_1 and C_2 are just constant values to avoid a zero denominator. Each patch similarity score is between zero and one. The value of W can change the values of SSQA, as it reflects the local features in intensity. For the choice in this paper, we are trying to compare the random regions that can include the interior intensity of a cell, so $W = 4$ pixels is chosen as the average radius (400 nm) of bacterial cells. In fact, the patterns of SSQA freqencies for different window sizes W were found to be similar to each other.

The third stage of SSQA is to repeat the second stage for N times, and then find the distribution of similarity S_{Data} at k^{th} image comparison. The current "real vs synthetic" $SSQA^k$ score from randomly picked images is the Bhattacharyya distance between two comparison-based similarity distributions, S_J^k and S_V^k. The $SSQA^k$ is defined as:

$$SSQA^k = -ln\left(\sum_{x=1}^{X}\sqrt{S_J^k(x)\cdot S_V^k(x)}\right) \tag{4.5}$$

where $x \in [0, X]$ is the domain of discrete probability distribution. Bhattacharyya distance[12] measures the overlapping amount of two distributions. It does not require an order of input distributions and can avoid a zero denominator when compared to KL-divergence. Its value ranges from zero to infinity, where a zero value means that the two distributions are identical. The distribution of patch similarity is discretized with 100 quantization levels, which results in SSQA being small ($\sim 10^{-2}$) as shown in **Fig. 4.7**.

Finally, the performances of K cross-dataset (real vs synthetic) SSQA scores are compared graphically with different data generation methods as shown in **Fig. 4.7**. To achieve statistically meaningful SSQA comparisons, $K = 600$ and $N = 10000$ are chosen in each evaluation. Statistical observations (e.g. mean, standard deviation) and intersection-over-union (IoU) of two SSQA histograms are analyzed as listed in **TABLE 4.3**. In terms of a single SSQA, a value closer to zero means better image relative quality. When we evaluate a unpaired dataset containing multiple image conditions, the comparison between intra-dataset (real vs real) and cross-dataset (real vs target) SSQA measures is needed. In this case, a smaller difference between the statistical observations compared to the reference (intra real dataset $SSQA_0$) denotes more similar dataset conditions. The mean values of SSQA, along with standard deviations (std), are shown in **TABLE 4.3**. The IoU values of SSQAs exhibit quantitative comparison of different synthetic datasets, where a value closer to 1 means the distributions of image intensity-wise qualities in the two datasets are more similar.

4.3.3 Empirical analysis of SSQA

In general, SSQA takes random patches from random images and analyzes the inter- and intra-datasets statistics. To extract the statistics, SSQA evaluates measures embedded in intensity and noisy background of each patch, in terms of illuminance and contrast. A question may be asked that "can we substitute the similarity measure by other existing scores?" In this section, further analysis on SSQA is performed to discuss the choices for parameters (e.g. patch size, sample numbers), and to compare different similarity measure metrics. It is worth mentioning that SSQA takes advantage of the stochastic statistical approach and does not need the two images to exactly match; while the other methods may require image to image structure correspondence. However, by taking the patch-based SSQA scheme, *i.e.*, embedding different similarity measures in our proposed SSQA framework, the limitations on structure correspondence can be diminished.

Parameter analysis: There are three major parameters in SSQA: patch/window size (W), the number of random patches (N), and the number of random images (K). **Table 4.1** demonstrates the comparison with different parameter settings and the corresponding SSQA scores. For varying patch size, images are fixed as shown in **Fig. 4.5**. For the other studies, there are twenty five 3D stacks from two SBR conditions in the model-based simulated datasets. Then, 2D images are selected from the same random depth in the 3D dataset with different SBR conditions. The

window size (W)	4	8	16	32
inter-dataset SSQA	0.1348	0.0931	0.0685	0.0851
intra-dataset SSQA	0.0025	0.0028	0.0026	0.0024
num. of images (K)	10	50	100	200
inter-dataset mean SSQA	0.0809	0.0941	0.0818	0.0882
intra-dataset mean SSQA	0.0025	0.0025	0.0025	0.0025
num. of patches (N)	50	500	5000	10000
inter-dataset mean SSQA	0.0983	0.1084	0.0900	0.0891
intra-dataset mean SSQA	0.0581	0.0279	0.0026	0.0025

Table 4.1: Parameter analysis in SSQA.

number of patches and the number of images are 10000 and 1. The fixed window size and the number of patches for analyzing varying number of images are 8 and 10000. When there are sufficient numbers of images and patches, both inter- and intra-dataset SSQA scores maintained a stable level of evaluation. Window size will affect the value of SSQA according to **Table 4.1**. However, when we repeat the validation on different sets of images, window size won't dramatically change the probability of SSQA as long as it is sufficient to describe the local characteristics in the image (e.g. around a bacterial diameter size).

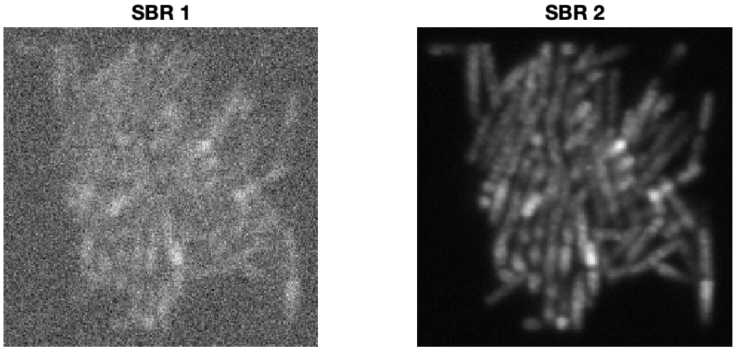


Figure 4.5: Images with different SBR conditions are used for the analysis of SSQA. For the validation of parameter setting with varying window size, SBR 2 image was used as a reference real image for intra-dataset comparison.

Comparison with different similarity measures: This section analyzes and compares different similarity measures to evaluate the image quality either for the entire image or for patch-wise similarity at the second stage in SSQA scheme. The similarity scores for comparison are normalized root mean-squared error (NRMSE), normalized cross-correlation analysis by evaluating the maximum value (max(NCC)), mutual information analysis (MI), and SSIM [126]. For NRMSE, the root of mean-squared error is normalized by the difference between the maximum and minimum intensities in the two patches. The parameter settings for the comparison are window patch size of 8, and the number of patches of 10000. The results shown in **TABLE 4.2** are comparisons using different quality assessment measures to evaluate qualities of the entire images (top three rows) and the mean qualities of random patches (bottom three rows). The number of bins for evaluating the similarity distribution is 100.

Our observation indicates that, when biofilm structure does not appear the same in the two images (see **Fig. 4.6** second and third columns), NCC, SSIM and MI fail to exhibit appropriate intra-dataset qualities, which are expected to be larger than inter-dataset values in these scores. Both NRMSE and SSQA present reasonable trends in the two comparisons, *i.e.,* larger values for inter-dataset evaluation than those for intra-dataset. While for SSQA, the inter- and intra-dataset scores are observed to be more diverse with a relative ratio of 100, which is larger than the ratio ($\sim$ 2 times) evaluated by NRMSE. In the cases when we analyze stochastic patch-wise similarities, the mean score closer to 1 for the S in SSQA denotes more similar qualities of the two images. Although the other scores, especially SSIM and MI, exhibit enhanced patterns that compare inter- and intra-dataset qualities, they are still lacking in diverse scores for different datasets.

For both studies in **Table 4.1** and **TABLE 4.2**, the original reference dataset quality affects the range and values of similarity scores. However, SSQA can still present a relative ratio of at least $\sim$ 30 between inter- and intra-dataset quality measures.

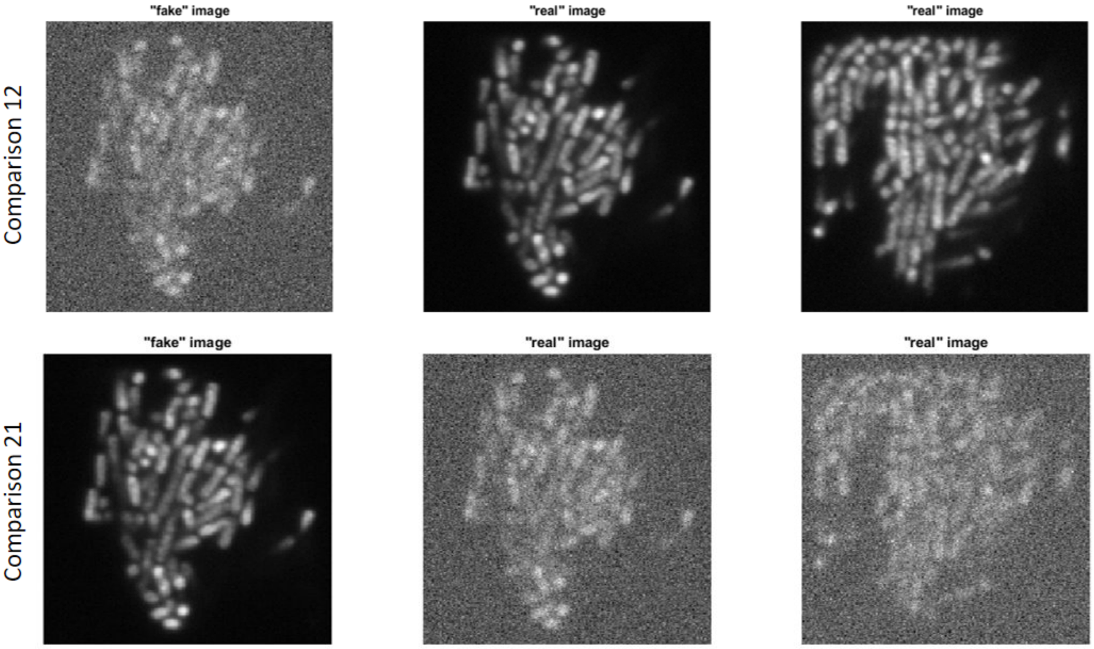


Figure 4.6: Images for the analysis and comparison of different similarity scores. In comparison 12, "real" images are SBR 2 images which are served as reference dataset; while in comparison 21, the reference dataset contains SBR 1 images.

	NRMSE	max(NCC)	SSIM	MI	SSQA
inter-dataset 21	0.3412	0.7086	0.1451	1.1283	1.0179
intra-dataset 11	0.1661	0.5607	0.0242	1.0999	0.0099
intra-dataset 22	0.1448	0.3197	0.0012	1.0507	0.0044
inter-dataset 21 (in SSQA scheme)	0.5770	0.5628	0.0294	1.8096	0.3178
intra-dataset 11 (in SSQA scheme)	0.4607	0.5563	0.0807	1.8101	0.4738
intra-dataset 22 (in SSQA scheme)	0.3041	0.3688	0.0945	1.8889	0.9651

Table 4.2: Comparison of different similarity scores to evaluate the quality of images or patches. The upper three rows compute the quality assessment scores using different methods between two images. The lower three rows evaluate the mean patch-wise similarity of two images. The patch-wise scores are averaged over 10000 comparisons. *Inter-dataset 21* denotes the comparison between SBR 2 and SBR 1 conditions (Fig. 4.5). *Intra-dataset 11* evaluates intra SBR 1 dataset quality and *intra-dataset 22* accounts for intra SBR 2 dataset comparison.

4.3.4 Other dataset quality assessment metrics

Fréchet inception distance (FID): FID score evaluates the quality of GAN outputs based on the statistics from the original real training dataset[52] as well as the statistics in the target outputs. It uses a pre-trained inception v3 model to extract deep features that discriminate real and generated images in terms of mean and covariance. Then, FID calculates the difference of two feature distributions using Fréchet distance. The lower the FID, the better the dataset quality. FID was found to be well-correlated with human judgment, but the inception v3 model was trained on ImageNet 2D images with no 3D interface as of yet. Therefore, only 2D slices in the 3D stacks were evaluated as shown in **TABLE 4.3**.

Blind/referenceless image spatial quality evaluator (BRISQUE): BRISQUE is an algorithm that evaluates the distortions in images without the need for corresponding reference images. It measures the deviation of the distributions in normalized distorted images from natural scene images which follow the Gaussian distribution[78]. Features in intensity and pixel neighbors are extracted and further analyzed to get a quality score using a support vector regressor. This paper uses the original version of BRISQUE, which predicts the distortions in 2D images. When its value is smaller, the image quality is better. For the comparison of dataset quality, smaller difference of BRISQUE value between the reference and target synthetic dataset is better.

Location correspondence with SBR: The evaluation on location correspondence is performed by overlaying the ground truth labels on generated synthetic data. The values of mean intensity and standard deviation in "cell regions" and "background regions" are extracted. To compare the difference among different datasets, signal-to-background ratio (SBR) is calculated as described in[137] that takes the ratio of foreground to background mean intensities. When the location correspondence is lower, the mean intensity in the foreground gets lower due to the lack of foreground signals, and the mean intensity in the background is higher because of more wrongly-positioned signals. Thus, higher SBR that is closer to the reference value in **TABLE 4.3** is better.

4.3.5 COMPARISON

With regard to comparing the image conditions quantitatively in real dataset and target synthetic datasets (**TABLE 4.3**), *3D Cyclic GAN (ℓ_{21})* presented the overall optimal dataset quality

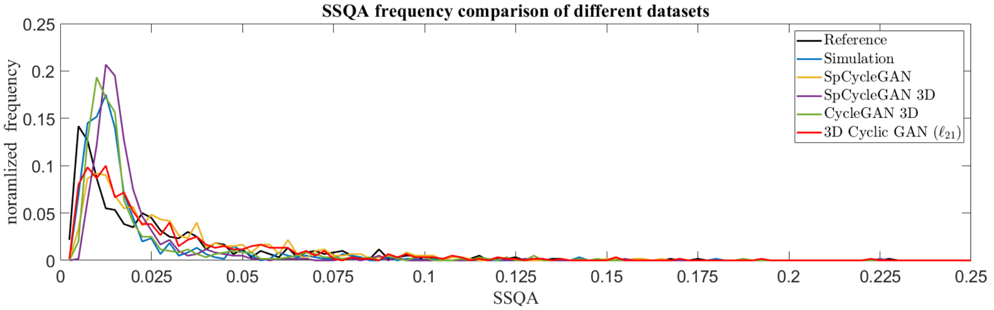


Figure 4.7: SSQA frequency comparison of different datasets. Histogram of SSQA for all the different outputs are normalized by the total number of K image comparisons to get the frequency of SSQA score in each dataset. The *Reference* plot indicates the stochastic comparison within the data pool of real images. The more similar the distribution compared to the reference, the better relative image quality of synthetic dataset.

Dataset	FID	BRISQUE (mean±std)	SBR	SSQA (mean±std)	IoU(SSQA, SSQA$_0$)
Reference	1.170e-10	36.252 ± 5.997	1.767	0.026 ± 0.029	1
Simulation	122.147	41.096 ± 3.269	**1.567**	0.019 ± 0.023	0.4670
SpCycleGAN	**45.188**	*34.124* ± 6.796	0.926	*0.030* ± 0.027	*0.6173*
SpCycleGAN 3D	103.074	38.478 ± 2.187	1.234	0.018 ± 0.018	0.3652
CycleGAN 3D	106.938	38.092 ± 5.439	1.364	0.021 ± 0.025	0.4371
3D Cyclic GAN (ℓ_{21})	*80.798*	**34.238** ± 5.155	*1.418*	**0.028** ± 0.029	**0.6575**

Table 4.3: Dataset quality assessment measures and their comparisons. Columns are quality assessment scores of different datasets. Fréchet inception distance (FID)[52] and blind/referenceless image spatial quality evaluator (BRISQUE) are performed on 2D slices in 3D stacks, while the others directly evaluate 3D images. Rows are different scores foreach dataset. The scores shown in BRISQUE and SSQA are mean and standard deviation (std) values. The SBR scores are also averaged over all the test images in the dataset. *Reference* denotes the intra real image data pool quality comparison or statistics. The closer to the *Reference* scores the better relative dataset image quality. The values highlighted in bold are best result and the underlined ones in italics are second best.

with best distortion-based measure (BRISQUE) and best intensity-wise comparisons (SSQA and IoU(SSQA,SSQA$_0$)). For the other two scores FID and SBR, *3D Cyclic GAN (ℓ_{21})* still achieve the second best. In visual inspection of different datasets as shown in **Fig. 4.3**, *3D Cyclic GAN (ℓ_{21})* also demonstrates reliable image details that mimic the microscopic image conditions in the real dataset. Especially in the background of the images, less regions of artifacts or over-smoothness are seen compared to some other augmented datasets. There are cell drifting problems for all the GAN augmented datasets, but the results of *3D Cyclic GAN (ℓ_{21})* are relatively better.

SpCycleGAN achieved the best FID score and comparable BRISQUE and SSQA scores to *3D Cyclic GAN* (ℓ_{21}), which indicates that the outputs from 2D GAN with the original spatial consistency loss provide decent 2D image intensity-wise and distortion-wise quality conditions. The visual results in **Fig. 4.3** also validate the intensity and distortion similarity compared to the real images. *SpCycleGAN* gets the lowest SBR, because it cannot preserve the axial signal correspondence along *z-axis* (**Fig. 4.8**) and has more regions of cell signals missing as shown in **Fig. 4.3**. When this spatial consistency loss is applied in 3D Cyclic GAN, better visual location correspondence, as quantified by SBR, is found in part of the *SpCycleGAN 3D* outputs, such as the first and last rows in **Fig. 4.3**. The outputs, however, cannot consistently yield realistic images, where the resultant images are over-smoothed and distorted outputs (see **Fig. 4.3** last row). *SpCycleGAN 3D* has the lowest mean of SSQA score over all the K image comparisons, but its SSQA frequency does not reflect the similar spread of different image conditions in the real LLSM microscopic image dataset (**Fig. 4.7**).

Model-based *simulation* outputs exhibit the best location correspondence in terms of SBR as shown in **Fig. 4.8**, due to the fact that the cell signals in the datasets are produced by incorporating the exact locations of cells in labels with theoretical fluorescent emission models, point spread functions, and noise conditions. These simulated datasets are suboptimal, because they cannot mimic the actual intensity and distortion statistics in real datasets with regards to FID, BRISQUE, and SSQA.

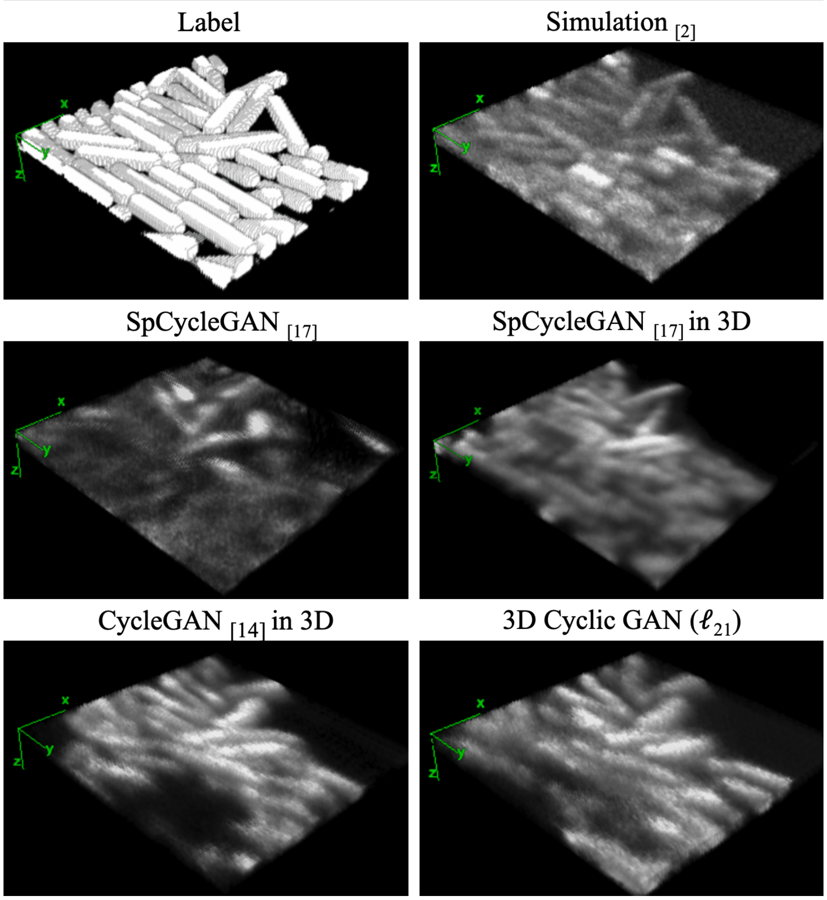
In summary, *3D Cyclic GAN* (ℓ_{21}) generates the overall most realistic dataset with qualities better than other GAN outputs in terms of intensity, distortion, and location correspondence. Additionally, the proposed 3D Cyclic GAN (ℓ_{21}) generated a diverse dataset with a similar mixture of different image conditions as compared to the reference dataset. This diversity in image conditions is observed by the mean and standard deviation values in BRISQUE and SSQA.

4.3.6 DISCUSSION

The following discussions are to analyze the limitations in the current GAN-based networks for biofilm application and other experiments to improve the GANs outputs. In the last paragraph, feasible solutions to overcome the limitations are discussed as future work, taking advantage of the proposed SSQA.

Although 3D Cyclic GAN successfully reproduces the LLSM microscopic distributions in the real experimental biofilms to generate 3D synthetic datasets, the spatial drifting of each bacterial cell region within the generated images is a limitation of the current GAN-based unpaired biofilm augmentation workflows (**Fig. 4.8**). Preliminary 3D segmentation performance with u-net[91] using a set of GAN generated synthetic dataset did not provide accurate segmentation outputs, as part of the training images and labels for segmentation are not perfectly matching to each other (**Fig. 4.8**). A possible reason for spatial drifting is that since the synthetic data are generated from a label consisting of all the cell locations as one joint block, the lack of spatial consistency is exacerbated as the noise level of the input data and the density of cells increase. The lack of spatial consistency means that the synthetic data generated may shift, distort, obscure, or disregard the cell regions compared to the original input label locations. **Fig. 4.8** shows an example with missing cells and location drifting rendered in 3D. It is worth mentioning that generating a synthetic dataset with unpaired images is challenging in nature as only about 50% pixel-wise accuracy was observed in the original unpaired image translation paper[140], and is already the best associated performance.

Additional experiments aiming to improve the dataset quality in intensity, distortion and spatial consistency were carried out with the presented 3D Cyclic GAN. These efforts include changing the architecture in terms of the number of layers and parameters in both the generator and discriminator, expanding the training dataset by flipping and cropping, and tuning basic training parameters. For example, we tested learning rate from a scale of 10^{-2} to 10^{-6}, added and


Label

Simulation [2]

SpCycleGAN [17]

SpCycleGAN [17] in 3D

CycleGAN [14] in 3D

3D Cyclic GAN (ℓ_{21})

Figure 4.8: A comparison of GAN outputs, which shows the preservation of axial (along *z-axis*) signal continuity with 3D GANs, as well as problems in location drift and missing cells associated with *SpCycleGAN* [39], *SpCycleGAN* loss extended to 3D, *CycleGAN* loss [140] used in 3D, and the proposed 3D cyclic GAN (ℓ_{21}). The ground truth label volume and model-based simulation result [137] are shown for reference. All the images shown are 3D rendered in ImageJ.

removed up to two layers, and tuned the parameters in the convolutional layers. We also modified the activation layers with sigmoid, and varied batch size from 1 to 5. None of these trials provided a better synthetic microscopic biofilm output than the current setup. The setup is using the architecture as shown in Fig. 4.2, the training set with 300 image volumes, batch size of 1, and learning rate of $2e^{-4}$.

However, with the 3D synthetic dataset quality evaluation scheme (SSQA) proposed in this paper, a future work can be done on filtering out the low quality and low label-to-image correspondence pairs to increase the location correspondence in the data. Other experiments can be explored by adding a small amount of manual annotation of real images to improve the overall GAN location correspondence for the biofilm learning task. Besides, the current biofilm data do not contain enough thick clusters of cells in *z-direction* that follow the density in CellModeller outputs, so we only selected 8 slices of them with comparable number of cells to label-set images. Collection of more LLSM biofilms data with thicker cluster size may be helpful as well to provide more and larger 3D volumes for learning the label-to-image spatial and axial translation.

4.4 SUMMARY

This chapter explored data augmentation options using generative adversarial networks to learn densely packed LLSM microscopic 3D biofilm images at the diffraction limit. The findings and contributions are concluded as follows, as are also the answers to the questions discussed in the beginning of this chapter:

- By modifying the original 2D layers to 3D layers in cyclic GANs, axial continuity in signals of cell regions along *z-direction* is better preserved when compared to 2D GAN outputs in both visual comparison and location correspondence evaluation with respect to the signal-to-background mean intensity ratio ($\text{SBR} > 1.2$).
- The most preferable synthetic dataset that mimics the realistic image conditions in the real dataset is produced by 3D cyclic GAN with unbalanced cycle consistency loss ℓ_{21}, which emphasizes the loss in the forward cycle direction that minimizes the ℓ_2-norm of the input labels and the cycled back generated labels. This unbalanced 3D Cyclic GAN achieved the best results in terms of three out of five scores in quantitative evaluation, and scored the second best in the rest two metrics. It also takes the shortest training time with 3D GAN models compared to the other loss function options in prior works.
- The proposed stochastic synthetic dataset quality assessment scheme, abbreviated as SSQA, provides a quantitative option for comparison-based 3D GAN output evaluation

when the corresponding ground truth images are not available. With adequate comparisons in terms of number of patches and images (e.g. 10000×600 in this paper), SSQA reveals statistically meaningful trends of intensity-wise cross-dataset quality.

The overall learning and evaluating pipeline presented in the chapter will be able to assist in generating more realistic images for future work on improving the single-cell segmentation work.

Imagination is more important than knowledge. Knowledge is limited. Imagination encircles the world.

Albert Einstein

5

Analysis of bacterial biofilms

Computation of bio-informatic measures from images of bacterial biofilms can provide cell biologists significant insights to understand and control the growth of biofilms. With the output of automatically identified cells from the previous chapters with *LCuts* and *m-LCuts*, we are able to extract some basic bio-infomatics regarding cellular position, cell length, orientation, and density. In this chapter, we will describe in detail how we reconstruct the biofilm models by geometrical models and refine the reconstructions using deformable active models. Preliminary shape-based alternatives, *e.g.*, the extension of our work *shape filter*[121] to biofilm scenario, to analyze, visualize, and compare bacterial cells are also presented in this chapter, as a gateway to the future work on statistic analysis of biofilm shapes in 3D.

5.1 BIOFILM RECONSTRUCTION

Biofilm reconstruction is an important step for LCuts and m-LCuts, especially in the mode of medial axis point cloud data, to recover the shape of bacteria cells. A pipeline of the current approach in this research is shown in **Fig. 5.1**.

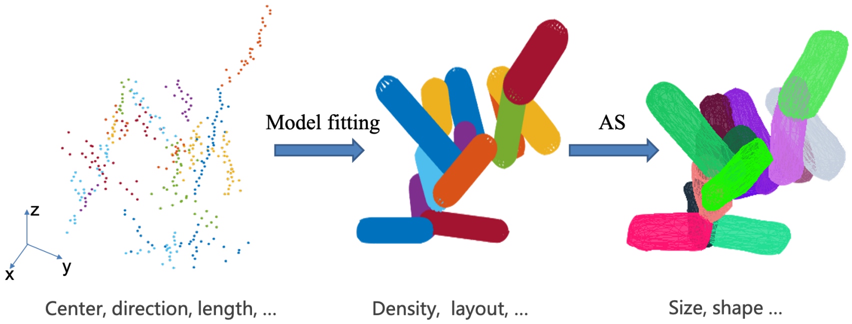


Figure 5.1: Pipeline of biofilm reconstruction. Panel 1: Result after LCuts and m-LCuts (under-segmented mask mode) is a point cloud data with different groups in different colors. Panel 2: Model fitting with spherocylinders, where cell length, position, and orientation are from the previous result. Panel 3: Further reconstruction can be realized by an active surface model to get the actual shape of bacterial cell in the intensity image.

5.1.1 MODEL FITTING

A computational efficient solution is provided by fitting a geometrical model, *i.e.,* a spherocylinder, to the actual position and orientation of a medial axis. A sherocylinder consists of a cylinder in the middle and two half sphere on each side of the cylinder. We first compute the centroid, orientation and length of a cell segment from LCuts and m-LCuts outputs. Then, a point cloud data of spherocylinder model is initialized at the origin, which is vertical to the *xy*-plane. By shifting and rotating, the final location of a estimated bacterial volume can be determined, as shown in the last row in **Fig. 5.2**.

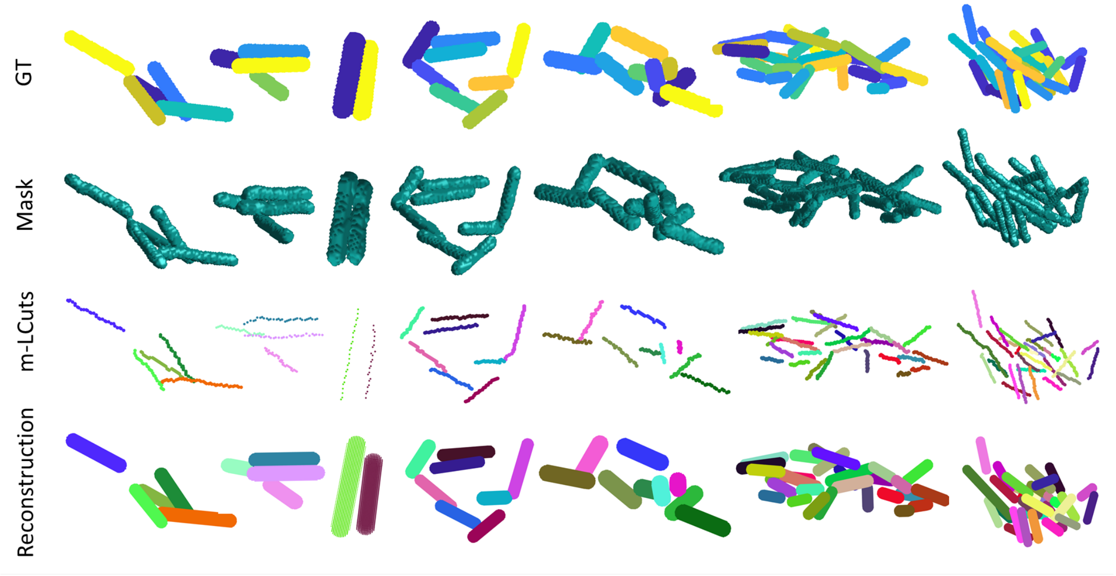


Figure 5.2: Examples of geometrical model fitting. GT is the ground truth layout of the cells during simulation. Mask is provided by initial segmentation method using u-net. m-LCuts outputs identify each cell as a central axis of a cell and reconstruct the volumes by fitting geometrical models to replace each medial axis data.

Evaluation of the performance on geometric model fitting uses two scores: single-cell grouping/segmentation accuracy (SGA) and single-cell boundary F1 score (SBF1) from Chapter 3. We randomly selected some examples as shown in **Fig. 5.2** to demonstrate the reconstruction comparison. SGA and SBF1 compares each cell in GT to a corresponding cell in reconstruction, while the former compares the voxel-level accuracy of the whole volume and the later compares the boundary accuracy of each cell segment. The scores are shown in **Table 5.1**. Although the geometric model fitted volumes cannot perfectly align with the ground truth cell layouts (e.g. SGA = 100%), the current performance with SGA $\sim$ 0.7 and SBF1 $\sim$ 0.9 is sufficient to identify the population of cells (for cell counting purpose) and to estimate the density of the cells in the biofilm.

ROI No.	1	2	3	4	5	6	7
SGA	0.756	0.825	0.649	0.799	0.790	0.707	0.752
SBF1	0.918	0.933	0.888	0.916	0.929	0.918	0.916

Table 5.1: Examples of evaluation on geometrical model fitting. SGA and SBF1 are defined in Chapter 3.

5.1.2 MODEL REFINEMENT WITH DEFORMABLE ACTIVE SURFACE

An advanced biofilm reconstruction is explored by employing parametric active contour/surface driven by vector field convolution (VFC)[65]. Different from the geometric active contour model as discussed in Chapter 2, a parametric active contour model cannot adapt to topological changes, but it can avoid merging of two contours from different identified cell segments.

A parametric active contour is usually initialized with landmarks, for example, some points sampled along the boundary of a circle or a surface. An illustration is shown in **Fig. 5.3**. Then, these landmarks will be attracted to the actual object boundary in a force field. In a closed curve/surface situation, the ordering of these landmarks is important as the active model will also try to keep the smoothness of the contour/surface that connects all the landmarks.

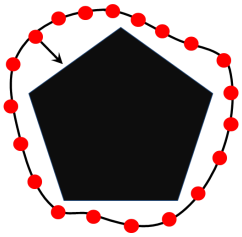


Figure 5.3: An illustration of a parametric active contour. The red dots are landmarks on the initialized active contour. The parametric active contour tries to approach the object boundary driven by a force field.

The force field is defined by utilizing vector field convolution (VFC)[65]*. An illustration that shows how we generate the force field in the biofilm application is shown in **Fig. 5.4**. In the current preliminary work, the edge map of a LLSM biofilm image is generated by sobel edge detection. It will then be convolved with the vector field kernel (VFK) defined in[65] to generate the force field, in which the flow of force points towards the detected edges, for our bacterial biofilm images. Mathematically, a VFK is defined as $\mathbf{k}(x, y, z) = m(x, y, z)\mathbf{n}(x, y, z)$, where m is the magnitude of the vectors in the kernel and $\mathbf{n}$ is the normal vector pointing to the kernel origin.

---

[*] Another successful application of VFC can be found in a collaboration manuscript, named C3VFC, with T. Ly, et al.

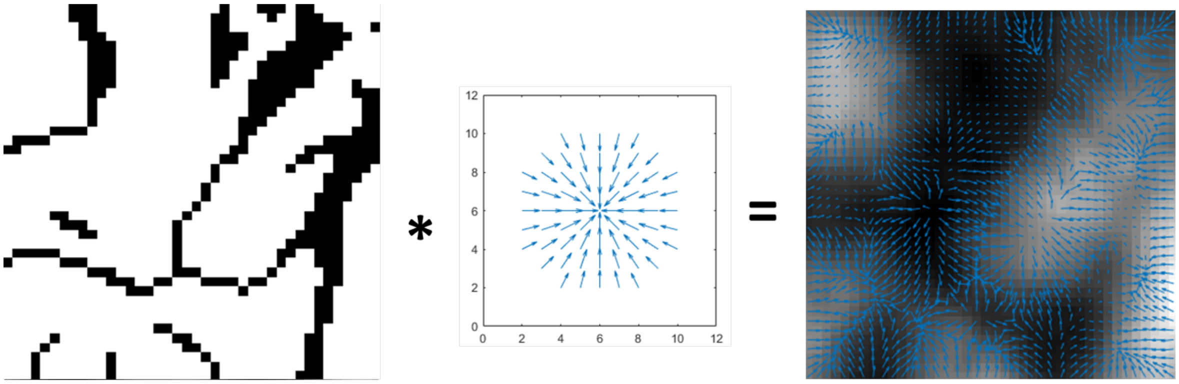


Figure 5.4: An illustration of the force field for model refinement using VFC[65]. The images from left to right are edge map of a biofilm image in a 2D view, the 2D view of a vector field kernel (VFK) with a radius of 10 as defined in[65], and the resultant velocity field (blue arrows) aligned with the orignal 2D view of the intensity image.

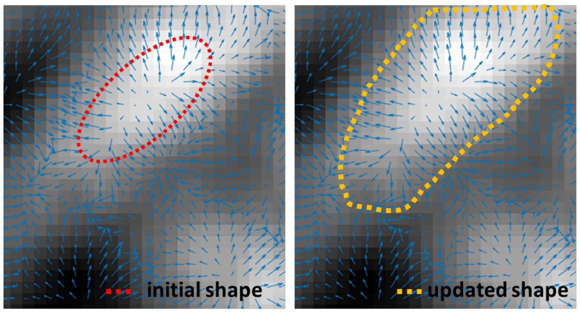


Figure 5.5: An illustration of the evolution of the active surface from geometrical model fitted shape to the actual bacterial shape in the image, driven by the force field. The dashed contours represent the parametric active surface in a 2D slice.

When the VFC-driven active surface is applied to refine the model at each cell location, an illustration of the evolution process is shown in **Fig. 5.5**. The expected result is an updated active surface evolved from the initialized location with geometrical model fitted shape. At the initialization step of the 3D active surface, each bacterial shape in 3D is parameterized by a set of points and triangular connections on the surface. Mathematically, the evolution of active surface is achieved by minimizing an energy functional, which has two parts: $E_{total} = E_{internal}(X, Y) + E_{external}(X, Y)$. The internal energy minimization will yield a smooth contour with a preferred shape and size, while the external energy minimization will yield agreement with the actual boundary of cells in the image, where the net force at these locations are minimal.

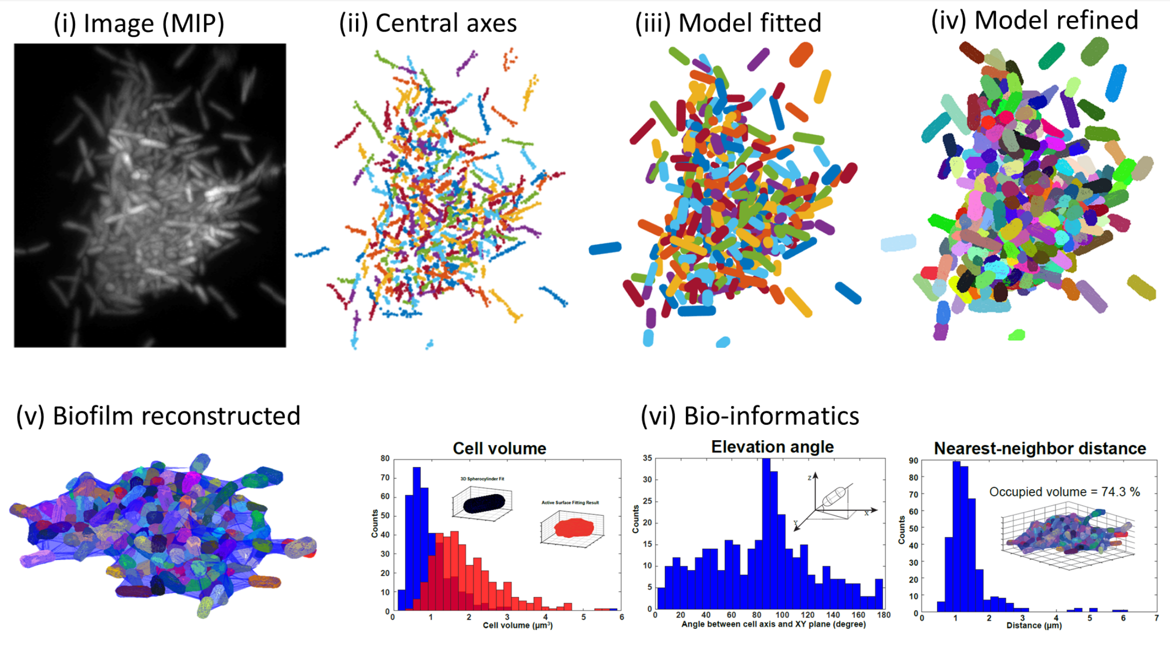


Figure 5.6: An example of biofilm reconstruction and the bio-informatics behind it. (i) The maximum intensity projection along *z*-axis of the deconvolved 3D *Shewanella* image. (ii) The central axes manually labeled along *x*-, *y*-, and *z*-axes. (iii) Model fitted with spherocylinder models. (iv) Model refined with VFC driven active surface. (v) Biofilm reconstructed by fitting a bounding surface that can enclose all the cells (with the *boundary* function in MATLAB). (vi) Bio-informatics that are interesting to domain area experts. The hitogram diagrams in (vi) are provided by M. Zhang. The cell volume diagram compares the distribution of cell volumes in geometrical model fitted biofilm and VFC surface refined model. The elevation angle diagram demonstrates the histogram of orientations of cells in the biofilm. The nearest-neighbor distance diagram computes the histogram on cell-to-cell distance in the biofilm.

Preliminary results: A preliminary result that applies VFC-driven active surface to refine geometrical models of cells is presented in **Fig. 5.6**. In this experiment, one model was evolved at each time in the VFC field, and the force at the area that was already occupied by this model was set to zero to avoid overlapping with other cells. As shown in **Fig. 5.6 (vi)**, the volumes, elevation angles, and cell-to-cell distances in the biofilm are interesting bio-informatics for the experts in cell biology. The refined models reveal a shifting in the hitogram of cell volume compared to the original geometrical model fitted biofilm, which may be caused by the lack of minimized force at the weak edges so that the active surface kept growing to a larger size. An improvement on the force field to stop the evolution of the active surface, even when the edges were not clearly presented in the image, can be a future work to explore. The current results still exhibit meaningful refinement on the shapes of bacterial cells, as the bending morphometry of some cells are restored.

5.2 SHAPE FILTER

With the proposed methods in this dissertation, biofilm segmentation are improved and refined to reconstruct the actual cell morphology. Next, we can analyze the statistics on the cell morphology by shape analysis. A shape is usually a representation of an object boundary, regardless of the texture, color, and scaling information. The statistic in shape can provide a novel aspect to analyze the deformation of bacterial cells along time and quantitatively compare the shapes of different bacterial species.

In this section, we introduce an application of our work on shape, named *shape filter*[121]. It was proposed to repair segmentation results obtained in calcium imaging of neurons *in vivo*. We extract the contour from an initially segmented cell in the time series images and exploit square-root velocity (SRV) transform[109] to represent the cells in shape space. The goal of *shape filter* is to generate a smooth path on the shape space to represent the evolution of a cell segment through time without any prior knowledge. *Shape filter* can also automatically filter out outlier segmentation results, in which case several cells maybe connected as one component.

Under the SRV representation, each shape is defined on a Riemannian manifold $\mathcal{M}$ with locally assembled SRV Euclidean coordinates q, defined as follows:

$$q(s) = \frac{\dot{\beta(s)}}{\sqrt{||\dot{\beta(s)}||}} \tag{5.1}$$

where $\beta(s)$ represents the shape, which is a closed curve parameterized by equal distant arc length $s \in [0, 2\pi]$. $\dot{\beta(s)}$ shows the gradient of this curve and $||\cdot||$ denotes the Euclidean norm. Suppose

α is the geodesic path between two arbitrary shapes on the defined manifold. Then, the difference between two shapes can be evaluated using the geodesic distance between the two SRV-transformed shapes:

$$d_g = \int_0^1 \sqrt{< \dot{\alpha(t)}, \dot{\alpha(t)} >} dt \tag{5.2}$$

where $\alpha(0)$ and $\alpha(1)$ represent the initial and final positions of the path, respectively. $\dot{\alpha(t)}$ shows the gradient of this geodesic path. When dt is infinitesimally small, the geodesic distance is approximately the integral length of each gradient magnitude.

After transforming the shape data into the SRV representation, an optimization problem is solved to filter out the extra components (*outliers*) in the preliminary result by interpolating the most possible shapes along the time-indexed calcium firing path.

We use φ to denote the original evolution path and γ to represent the filtered new path on the manifold. Both of these two paths are approximately differentiable and have the geodesic distance defined as aforementioned. Then, γ is estimated by optimizing the following regression problem, which is modified from De Boor's approach[27]:

$$\min_{\gamma} \rho \underbrace{\sum_{t=start}^{end} w(t)|\varphi(t) - \gamma(t)|^2}_{data\ term} + (1-\rho) \underbrace{\int |D^2\gamma(t)|}_{smoothness\ term} \tag{5.3}$$

This modified shape regression model aims to find a desired minimizer γ that balances the trade-off between approaching the original data and smoothing the filtered path. In equation 5.3, t is time index for each shape, which is also used in the following sections in this paper. $D^2\gamma(t)$ is the second derivative of path γ, which characterizes the changes of shapes along the fitted path. $\rho \in [0, 1]$ is the smoothing parameter that reflects the emphasis on data or smoothness. Note that, when ρ is close to 1, γ becomes the spline of the input data φ; and when ρ is getting smaller, γ will be smoother with fewer shape changes along time in terms of SRV transformed shape rep-

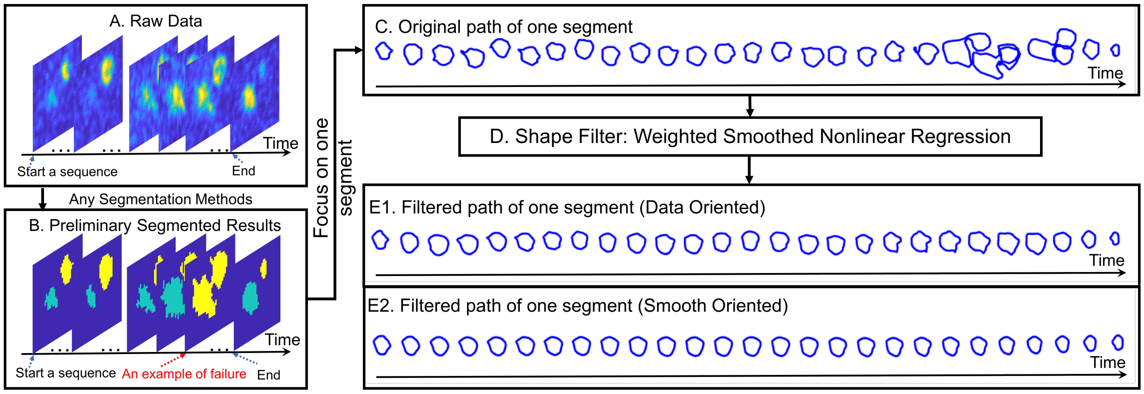


Figure 5.7: Flow chart of *shape filter* applied to filter calcium imaging of neurons.

resentation. $w(t)$ is the local weight to estimate the *outliers* automatically, for which we proposed two alternatives in the *shape filter* paper[121]. Details of the two weighting functions are introduced as follows.

Bi3 local shape weighting: Inspired by tricube kernel model and robust local regression model in[7], Bi3 local shape weight is defined as:

$$w_3(t) = A * (1 - (\frac{r(t)}{\text{median}(r(t)) + \tau})^3)^3 \tag{5.4}$$

where

$$r(t) = d_g(\delta(t), \varphi(t)) \tag{5.5}$$

$$\sigma_r = \frac{1}{N}\sum_{t=1}^{N}(|r(t) - median(r(t))|) \tag{5.6}$$

$$\tau = \sigma_r + (\sigma_r - min(r(t))) \tag{5.7}$$

where $r(t)$ is the residual geodesic distance from the true data to a fitted smooth spline δ with a small weight for the data term in equation (5.3). σ_r specifies the mean deviation of residuals from δ. $median(r(t))$ is treated as the mean shape in the sequence instead of the traditional mean because median is more robust with *outliers* in the data. τ is the tolerance for residual deviations.This τ will automatically assign a negative feedback to data term when the data is far away (more than tolerance) from the smoothed spline δ. The formula to calculate this value is inspired by the skew measure in statistical analysis. A is just a constant that can amplify the proportion of the data term whenever necessary. The choice of the bi-cube function is capable of constructing wider low pass range for data when the base of the inner cube is closer to zero[7].

Modified shape Gaussian (sGaussian) weighting: The Gaussian model is also a popular weighting selection in local regression problems, such as in[70]. To adapt the framework of Gaussian weighting to shapes, the weighting may be computed using:

$$w_4(t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{d_g^2(\varphi(t), q_{median})}{2\sigma^2}) \tag{5.8}$$

where

$$\sigma^2 = \frac{1}{N}\sum_{t=1}^{N}(d_g(\varphi(t), q_{median}))^2 \tag{5.9}$$

Here, q_{median} represents the Euclidean median shape along the input path α with N shapes in the SRV representation. The Euclidean distance in the normal Gaussian cases is replaced by the geodesic distance.

A flow chart of the application in the original paper is shown in **Fig. 5.7**, where the time sequence segmentation results with outliers are filtered and replaced with shape estimation along the neuron firing process.

5.3 EXTENDING THE SHAPE FILTER TO SHAPE-BASED BIOFILM ANALYSIS

A possibility that *shape filter* experiments bring to biofilm analysis is the ability to analyze bacterial shapes. More specifically, the possible analysis include fixing the rough and wrong segmented results, visualizing the deformation of cells, and quantifying the difference between different bacterial cells in the image. In this section, experiments on 2D shapes using *shape filter* will be presented. A potential solution to extend *shape filter* in 3D is also discussed.

5.3.1 Experiment on 2D shapes

The experiments on the 2D shapes of bacterial biofilm data are performed on the BCM3D processed LLSM time lapse *E.coli* dataset[137]. The maximum projection of these segmentation results are shown in **Fig. 5.8**. The biofilm images were captured every 30 minutes in 3D by the Nanoscale Lab at UVA. To demonstrate the shape analysis using *shape filter*, a target cell segment is selected as shown in the bottom panel of **Fig. 5.8**. It is notable to mention that there is a segmentation error at 510 mins, where two cells are under-segmented as one connected component. The corresponding cell segments are manually traced. The shapes of the segments are parameterized with 100 sampling landmarks along their contours.

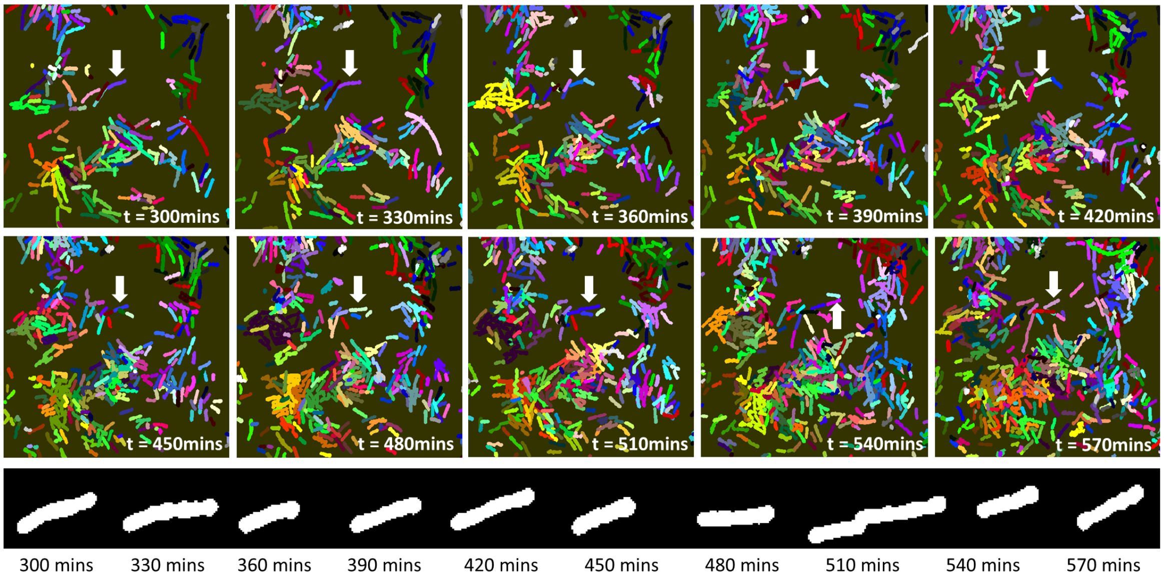


Figure 5.8: Segmented dataset for shape analysis with *shape filter*. The images are all maximum projection along *z*-axis of the original 3D image. The segmentation results are provided using BCM3D[137]. The cell marked with the white arrow is the target for shape analysis in this section. The target cell segment is also cropped out individually in the bottom panel in the figure. The experimental time lapse images are imaged at an interval of 30 minutes.

By using the *shape filter*, the segmentation error can be filtered out as shown in **Fig. 5.9** with both of the proposed weighting functions. According to the resulting paths, Bi3 local shape weighting (Bi3) provides a path that is closer to the shapes in the original path, compared to shape Gaussian weighting (sGaussian). Note here, the same parameter $\rho = 0.9$ in **eq. (5.3)** is used. The corresponding geodesic paths of the three shape paths along time are projected on a dimension reduction plane, using the isomap method[115]. As all the shapes along the time sequence are also labeled with orders in **Fig. 5.9**, we are also able to demonstrate the relative quantitative difference between two bacterial shapes. **Fig. 5.9** also indicates that the paths after nonlinear filtering with *shape filter* are smoother and denser than the original path, while Bi3 weighting can keep more information from the original path.

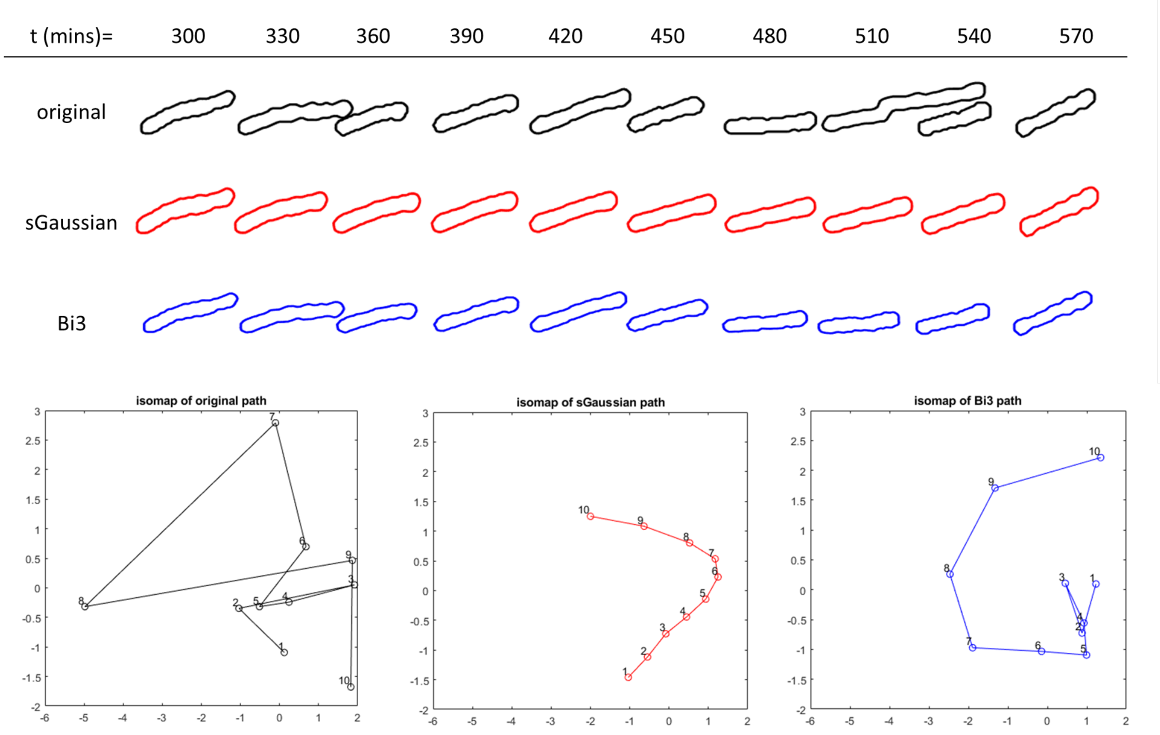


Figure 5.9: Filtered paths with *shape filter* and projection of the corresponding geodesic path on a dimension reduction plane. The filtered paths are outputs using the proposed shape Gaussian weighting (sGaussian) and Bi3 local shape weighting (Bi3). The geodesic path of each shape sequence is projected on a 2D plane using isomap dimensionality reduction[115]. Each circle markers on the isomap indicates the ordered shape in the time sequence. The *x* and *y* are relative geodesic distance projected on the 2D plane.

The *shape filter* can also provide estimations of shapes/segments that may not be originally imaged at the experimental imaging step, and potentially provide insights for biofilm statistic analysis. An experiment of shape interpolation and an analysis on tracking the change of volume along time are shown in **Fig. 5.10**. It is observed that *shape filter* can provide reasonable estimation of shapes even though the images were not obtained during the experimental imaging process. Given the estimations, a potential analysis that *shape filter* can provide to the cell biologists is a track of volume variations along time. It is interesting to find a periodic change of volume of the target cell in **Fig. 5.10**, as it approximately agrees with the cell dividing situation observed in the real time lapse images.

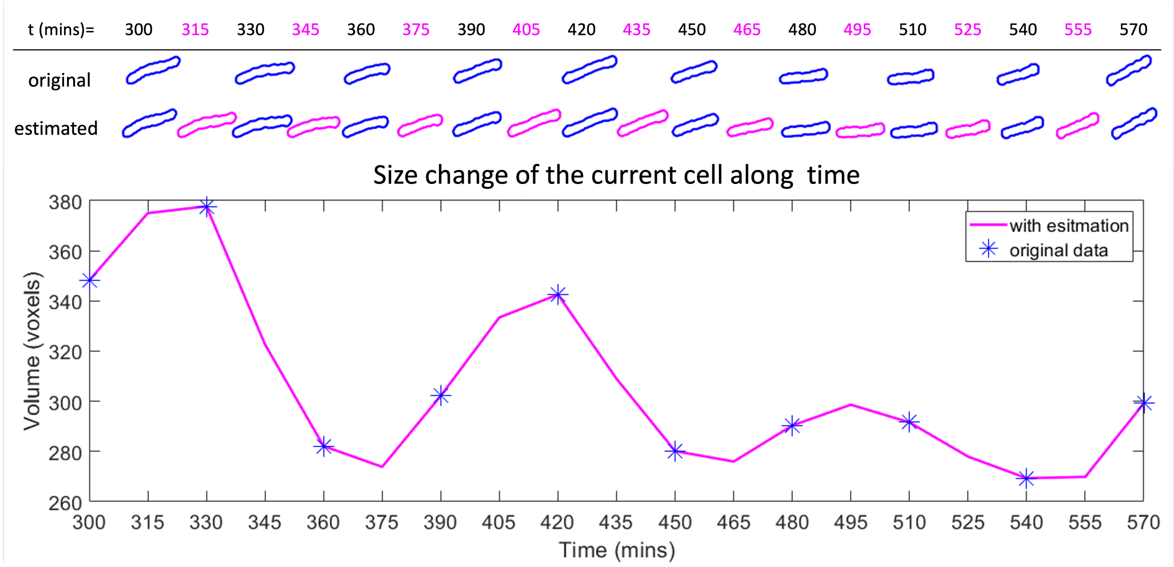


Figure 5.10: Estimation of shapes and potential biofilm analysis on cell volume provided by *shape filter*. The top panel demonstrates the path estimated in *shape filter* by interpolation. The shapes in blue are original shapes (after Bi3 filtering) in the experimental imaging sequence, and the ones in magenta are new estimation/interpolations of shapes. The bottom diagram reveals the track of volume changes of the target cell along time (with interpolations). The original data at original time scales are marked with blue asterisk.

5.3.2 Potentials for 3D shapes

The 2D experiments exhibit encouraging findings as discussed in the last section, while the 3D representation of cells can definitely bring more insights for biofilm analysis. A barrier in conducting 3D shape experiments, at the current stage, is the lack of a suitable parameterization method for 3D shapes that can be analyzed in Riemannian manifold as *shape filter* utilized. In order to find an elastic shape representation in Riemannian manifold for the 3D bacterial shapes, a potential solution is in progress, collaborating with T. Toma in the VIVA lab and shape expertise from the Florida State University (FSU), A. Srivastava and his student Y. Wu. In their recent paper, a square root normal field (SRNF) is presented as a solution to describe 3D shapes, along with all the potential elastic shape analysis statistics[62,55].

Based on the theoretic papers[62,55], the 3D surface of each bacterial cell is a function f defined on domain $\mathcal{D} \rightarrow \mathbb{R}^3$, where each point s on the surface $s \in \mathcal{D}$ is a point with three-dimensional coordinates. Then, the representation of a 3D surface with SRNF is:

$$q(s) = \frac{n(s)}{\sqrt{n(s)}} \tag{5.10}$$

where $n(s)$ is the normal vector at each point s on the shape surface; $q(s)$ is an $\mathbb{L}_2$-metric under the SRNF representation. Thus, SRNF transform preserves the properties in the original SRVF for comparing shape difference in the differentiable shape space. A preliminary SNRF-transformed bacterial shape, provided by Y. Wu, is shown in **Fig. 5.11**. With the supporting statistics from SRNF, a future work can target on extending *shape filter* to the 3D space. The higher dimensional shape data points that represent 3D surfaces will be potentially fitted in the data filtering scheme by adjusting the weighting functions to a higher dimension.

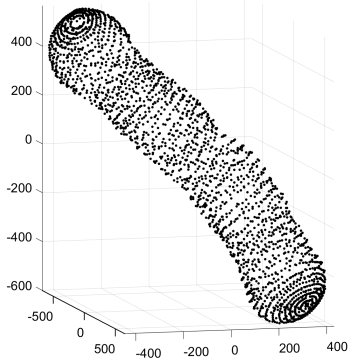


Figure 5.11: An example of 3D shape parameterized with square root normal field[62].

5.4 SUMMARY

In this chapter, biofilm reconstruction using geometrical model fitting and active surface-based model refinement was firstly presented. These biofilm reconstruction methods can smoothly link up the central axis point cloud outputs from *LCuts* and *m-LCuts* and thus provide bio-informatics of cells in the biofilms, such as cell position, orientation, biofilm density, *etc.* By combining the advances in biofilm image analysis algorithms presented in this dissertation, the single cell segmentation results can be improved, which further enables shape-based biofilm analysis using *shape filter*. Based on the preliminary experiments in this chapter, *shape filter* is able to fix the segmentation errors along time, visualize and estimate the deformation of cells, and quantify the morphometry difference (e.g. volume, shape) of different bacterial cells in the image. A future work will be on extending the current shape analysis workflow to 3D shape space.

Nothing is impossible. The word itself says 'I'm possible!'

Audrey Hepburn

6

Conclusions

6.1 CONTRIBUTIONS

This dissertation presents four thrusts from Chapters 2 to 5. Several automated algorithms were proposed in each thrust, which include the graph-based segmentation variations, biofilm image synthesis, evaluation and biofilm analysis solutions. This research breaks through the inability in resolving and analyzing individual cells in densely-packed biofilm. It also enables single-cell and population-level studies of biofilms, incorporating both classic and machine learning image analysis techniques, as a gateway for future work on biofilm analysis. The major contributions are listed below.

Contribution I: **Graph-theoretic automated bacterial cell segmentation with *L-Cuts*.** The first thrust is directed toward the segmentation of individual bacterial cells in dense biofilm. ***Bact-3D*** and ***LCuts*** were presented in Chapter 2 as single-cell segmentation solutions. *Bact-3D* is the first effort of this biofilm research, and gives an edge-based level-set segmentation method with local velocity constraints to prevent the merging of two cell contours. Although promising results for multi-layered biofilm 3D data were achieved, the performance degrades on more complex biofilm images as the gaps between neighboring cells are harder to identify. Therefore, *LCuts* was presented to detect each cell by analyzing its central axis, even when the cell boundary is ambiguous.

In *LCuts*, the segmentation of bacterial cells is transformed to a key point clustering problem via recursive graph cuts. It computes the local intensity maxima to delineate the central axis of each cell; then it automatically divides these point cloud data to different collinear groups to represent each rod-shaped bacterial cell. Distance, linearity, and intensity characteristics of the point cloud data are utilized to define the new graph node similarity measure. *LCuts* outperforms the existing methods in the majority cases in 2D. Experiments in 3D with *LCuts* indicate the necessity to improve the automatic point cloud data generation process that can provide data that exhibit explicit collinearity features, for which the second part of this research was carried out.

Contribution II: ***m-LCuts*** **for generalized collinear data clustering with application to post-processing.** In the second thrust (Chapter 3), the extension from *LCuts* to masked *LCuts* (***m-LCuts***) was presented to maximize the single cell identification performance and to generalize the method to be able to combine with any other segmentation pipelines. By utilizing preliminarily segmented outputs (masks) from the general segmentation approaches, the point cloud data for *m-LCuts* are extracted using either radius-constrained medial axis extraction method on under-segmented masks or supernode generation workflow for over-segmented masks. In these cases, the collinearity in the data can be preserved and utilized for further data clustering. Compared to the initial *LCuts*, *m-LCut*s algorithm reports new sub-algorithms in graph construction, node feature extraction, node similarity measure, and stopping criterion for recursion.

The validation experiments are twofold. Firstly, the incorporation of *refined LCuts*, the intermediate version, with deep neural networks was investigated in ***BCM3D***. *Refined LCuts* can further split the connected components where multiple cells merge into one cluster in the CNN-trained segmentation outputs. Secondly, experiments were conducted to generalize ***m-LCuts*** as it can post-process segments with different segmentation errors with one unified theory. In addition, to deal with the under-segmented clusters, *m-LCuts* can also remedy the situation in which a single cell is over-segmented into several parts. In such a case, the segmentation outputs (masks) can be provided with any segmentation approaches, e.g. machine learning-based facilities or traditional computational image processing solutions.

Furthermore, *m-LCuts* is a broadly applicable data clustering method, potentially extendable to many other scenarios. For example, in the experiments of shape classification, the masks are binary image of shapes and *m-LCuts* can find the classes of shapes using their collinearity in shape space. With each of the *LCuts* variation, there is no need to specify the number of cells to be segmented.

Contribution III: 3D GAN image synthesis and dataset quality assessment with SSQA for bacterial biofilms. The third thrust of this dissertation focuses on the generation of biofilm images (Chapter 4). A machine learning solution using generative adversarial networks (GANs), named ***3D Cyclic GAN (ℓ_{21})*** with unbalanced loss functions, was designed to generate synthetic 3D images that mimic the actual LLSM biofilm image conditions. The proposed workflow is a solution to potentially increase the segmentation performance on real experimental biofilm images, and to overcome the current limitation in data-driven segmentation pipelines that the model-based simulated training data may not fully reproduce the complicated image conditions in the biofilm dataset. The unbalanced loss function tested in *3D Cyclic GAN (ℓ_{21})* provides the best synthetic data quality compared to model-based image generation and other GAN loss options, in terms of intensity-wise and distortion-based image quality assessment measures.

To fill in the lack in assessing 3D synthetic dataset quality, a stochastic synthetic dataset quality assessment scheme was proposed, abbreviated as ***SSQA***. *SSQA* is a stochastic approach that takes random patches from random images for quality evaluation and statistical observation analysis. It provides a quantitative option for comparison-based 3D GAN output evaluation when the corresponding ground truth images are not available. With adequate comparisons in terms of number of patches and images (e.g. 10000×600 in this paper), SSQA revealed statistically meaningful trends of intensity-wise cross-dataset quality. We believe that this approach is applicable to other bioimage data generation problems for which blind quality assessment is required.

Contribution IV: Shape-based biofilm reconstruction and analysis. By providing the image analysis and machine learning advances to improve the single-cell segmentation performance as mentioned above, biofilm analysis is enabled for researchers in the biofilm community. As discussed in Chapter 5, biofilm reconstructions using ***geometrical model fitting*** and ***active surface-based model refinement*** are both provided with preliminary results to demonstrate the handling of *LCuts* and *m-LCuts* outputs. They are able to provide statistics in cell location, orientation, size, biofilm density, and even shapes for biofilm biologists to perform analysis. Furthermore, ***shape filter*** was applied to analyze bacterial shapes along time. Experiments demonstrate the potential that the *shape filter* brings to the biofilm analysis. For example, the morphological tools allow one to visualize the shape deformation of cell growing-and-dividing process and to quantify the difference in cell morphometry spatially and temporally.

6.2 Future work

One of the future directions of this work will focus on biofilm analysis with the extension of a *shape filter* in 3D. As also mentioned in Chapter 5, a collaborative project on this topic is actually in progress, using an elastic shape representation of bacterial cells in 3D. In addition, for biofilm reconstruction, further improvements will be explored on the active surface-based model refinement, such as adding shape prior on each active surface model, which may also include systematic experiments on evaluating the corresponding biofilm reconstruction accuracy.

For biofilm image generation, future work may involve the improvement of GAN pipeline to decrease the problems in location drifting of cells in the generated images compared to input binary annotations. On the other hand, experiments will be performed using *SSQA* and the adjusted SBR score from Chapter 4 to filter out the images with low quality and low location correspondence. Then, the segmentation pipelines can be refined to improve the single-cell identification accuracy. These works will further assist in accurate single-cell tracking in the biofilms along time.

Another branch of future work will be devoted to broader applications of the proposed graph-based clustering method, *m-LCuts*. A preliminary result on shape classification is already shown in Chapter 3. Works will be further carried out to define a stopping criterion that can automatically terminate each recursion, for the detection of different collinear shape groups in shape space. In addition, the goal to further generalize *m-LCuts* may also require future modifications in the algorithm and optimization of the runtime. Then, more experiments will be performed to validate the performance of the developments and to compare with the current state of the art.

6.3 List of publications

C1 **J. Wang**, R. Sarkar, A. Aziz, A. Vaccari, A. Gahlmann, and S. T. Acton. "Bact-3D: A level set segmentation approach for dense multi-layered 3D bacterial biofilms." In *2017 IEEE*

International Conference on Image Processing, pp. 330-334. (2017)

C2 **J. Wang**, Z. Fu, N. Sadeghzadehyazdi, J. Kipnis, and S. T. Acton. "Nonlinear Shape Regression for Filtering Segmentation Results from Calcium Imaging." In *2018 25th IEEE International Conference on Image Processing*, pp. 738-742. (2018)

C3 **J. Wang**, T. Batabyal, M. Zhang, J. Zhang, A. Aziz, A. Gahlmann, and S. T. Acton. "LCuts: Linear Clustering of Bacteria using Recursive Graph Cuts." In *2019 IEEE International Conference on Image Processing*, pp. 1575-1579. (2019) **Listed in the Top 10% of papers in ICIP 2019.**

J1 M. Zhang†, J. Zhang†, Y. Wang†, **J. Wang**, A. Achimovich, S. T. Acton, and Andreas Gahlmann, "Non-Invasive Single-Cell Morphometry and Tracking in Living Bacterial Biofilms." *Nature Communications*, 11(1), pp.1-13. (2020)

J2 N. Tabassum, **J. Wang**, M. Ferguson, J. Herz, M.l Dong, A. Louveau, J. Kipnis, and S. T. Acton. "Image Segmentation for Neuroscience: Lymphatics." *Journal of Physics: Photonics*, 3(3). (2021)

J3 **J. Wang**, M. Zhang, J. Zhang, Y. Wang, Andreas Gahlmann, and S. T. Acton, "Graph-theoretic Post-processing of Segmentation with Application to Dense Biofilms." *IEEE Transaction on Image Processing*. (2021)

J4 T. T. Ly, **J. Wang**, K. Bisht, U. Eyo, and S. T. Acton. "C3VFC: A Method for Tracing and Quantification of Microglia in 3D Temporal Images." *Applied Sciences*, 11(13), 6078. (2021)

J5 **J. Wang**, N. Tabassum, T. T. Toma, Y. Wang, A. Gahlmann, and S. T. Acton. "3D GAN Image Synthesis and Dataset Quality Assessment for Bacterial Biofilm". *Submitted to TIP*. (2021)

J6 **J. Wang**[†], T. T. Ly[†], N. Tabassum[†], and S. T. Acton. "Review on Level Set Methods in Biological Imaging." *Biological Imaging*. *in preparation*. (2021)

J7 T. T. Toma, Y. Wu, **J. Wang**, A. Srivastava, A. Gahlmann, and S. T. Acton. "Realistic-shape Bacterial Biofilm simulator for Deep Learning-based 3D Single-Cell Segmentation." submitted to *ISBI 2022*. (2022)

A1 M. Zhang, J. Zhang, **J. Wang**, A.M. Achimovich, A.A. Aziz, J. Corbitt, S. T. Acton and A. Gahlmann. "3D Imaging of Single Cells in Bacterial Biofilms using Lattice Light-sheet Microscopy." *Biophysical Journal*, 116(3), p.25a. (2019)

A2 J. Zhang, M. Zhang, Y. Wang, **J. Wang**, S. T. Acton, Andreas Gahlmann, "Non-Invasive Single-Cell Morphometry and Tracking in Living Bacterial Biofilms." *Biophysical Journal*, 120 (3), 358a. (2020)

[*] [†]

---

[*] C: Conference, J: Journal, A: Abstract

[†] The authors have equal contribution.

References

[1] Abdollahi, B., Tomita, N., & Hassanpour, S. (2020). Data augmentation in training deep learning models for medical image analysis. In *Deep Learners and Deep Learner Descriptors for Medical Applications* (pp. 167–180).

[2] Abramian, D. & Eklund, A. (2019). Generating fmri volumes from t1-weighted volumes using 3d cyclegan. *arXiv preprint arXiv:1907.08533*.

[3] Ackermann, M. (2015). A functional perspective on phenotypic heterogeneity in microorganisms. *Nature Reviews Microbiology*, 13(8), 497–508.

[4] Acton, S. T. & Ray, N. (2009). Biomedical image analysis: Segmentation. *Synthesis Lectures on Image, Video, and Multimedia Processing*, 4(1), 1–108.

[5] Akram, S. U., Kannala, J., Eklund, L., & Heikkilä, J. (2016). Cell proposal network for microscopy image analysis. In *2016 IEEE International Conference on Image Processing* (pp. 3199–3203).

[6] Ali, S. & Madabhushi, A. (2012). An integrated region-, boundary-, shape-based active contour for multiple object overlap resolution in histological imagery. *IEEE Transactions on Medical Imaging*, 31(7), 1448–1460.

[7] Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3), 175–185.

[8] Atta-Fosu, T., Guo, W., Jeter, D., Mizutani, C. M., Stopczynski, N., & Sousa-Neves, R. (2016). 3D clumped cell segmentation using curvature based seeded watershed. *Journal of Imaging*, 2(4), 31.

[9] Bannon, D., Moen, E., Borba, E., Ho, A., Camplisson, I., Chang, B., Osterman, E., Graf, W., & Van Valen, D. (2018). Deepcell 2.0: Automated cloud deployment of deep learning models for large-scale cellular image analysis. *BioRxiv*, (pp. 505032).

[10] Berg, S., Kutra, D., Kroeger, T., Straehle, C. N., Kausler, B. X., Haubold, C., Schiegg, M., Ales, J., Beier, T., Rudy, M., et al. (2019). Ilastik: interactive machine learning for (bio) image analysis. *Nature Methods*, (pp. 1–7).

[11] Beucher, S. et al. (1992). The watershed transformation applied to image segmentation. *Scanning Microscopy Supplement*, (pp. 299–299).

[12] Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, 35, 99–109.

[13] Bouman, C. A., Shapiro, M., Cook, G., Atkins, C. B., & Cheng, H. (1997). Cluster: An unsupervised algorithm for modeling gaussian mixtures.

[14] Boykov, Y. & Funka-Lea, G. (2006). Graph cuts and efficient nd image segmentation. *International Journal of Computer Vision*, 70(2), 109–131.

[15] Boykov, Y. Y. & Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Proceedings 2001 IEEE International Conference on Computer Vision*, volume 1 (pp. 105–112).

[16] Cavalier-Smith, T. (2002). The neomuran origin of archaebacteria, the negibacterial root of the universal tree and bacterial megaclassification. *International Journal of Systematic and Evolutionary Microbiology*, 52(1), 7–76.

[17] Chan, T. F. & Vese, L. A. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, 10(2), 266–277.

[18] Chen, B.-C., Legant, W. R., Wang, K., Shao, L., Milkie, D. E., Davidson, M. W., Janetopoulos, C., Wu, X. S., Hammer, J. A., Liu, Z., et al. (2014). Lattice light-sheet microscopy: imaging molecules to embryos at high spatiotemporal resolution. *Science*, 346(6208), 1257998.

[19] Choi, H. I., Choi, S. W., & Moon, H. P. (1997). Mathematical theory of medial axis transform. *Pacific Journal of Mathematics*, 181(1), 57–88.

[20] Choudhry, P. (2016). High-throughput method for automated colony and cell counting by digital image analysis based on edge detection. *PloS One*, 11(2), e0148469.

[21] Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). 3d u-net: learning dense volumetric segmentation from sparse annotation. In *2016 International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 424–432).

[22] Costerton, J. W., Stewart, P. S., & Greenberg, E. P. (1999). Bacterial biofilms: a common cause of persistent infections. *Science*, 284(5418), 1318–1322.

[23] Croft, M. T., Lawrence, A. D., Raux-Deery, E., Warren, M. J., & Smith, A. G. (2005). Algae acquire vitamin b 12 through a symbiotic relationship with bacteria. *Nature*, 438(7064), 90–93.

[24] Csurka, G., Larlus, D., Perronnin, F., & Meylan, F. (2013). What is a good evaluation measure for semantic segmentation? *BMVC*, 27(2013).

[25] Culver, T., Keyser, J., & Manocha, D. (2004). Exact computation of the medial axis of a polyhedron. *Computer Aided Geometric Design*, 21(1), 65–98.

[26] Dataset (Retrieved 2020). http://www.dabi.temple.edu/s̃hape/mpeg7/dataset.html. *MPEG-7 CE-Shape-1 dataset.*

[27] De Boor, C. (2001). A practical guide to splines, revised edition, vol. 27 of applied mathematical sciences. *Mechanical Sciences*.

[28] Delorme, C. & Poljak, S. (1993). Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming*, 62(1), 557–574.

[29] Dimitrakopoulos, P., Sfikas, G., & Nikou, C. (2020). Ising-gan: Annotated data augmentation with a spatially constrained generative adversarial network. In *2020 IEEE 17th International Symposium on Biomedical Imaging* (pp. 1600–1603).

[30] Dodd, M. S., Papineau, D., Grenne, T., Slack, J. F., Rittner, M., Pirajno, F., O'Neil, J., & Little, C. T. (2017). Evidence for early life in earth's oldest hydrothermal vent precipitates. *Nature*, 543(7643), 60–64.

[31] Dogra, J., Jain, S., & Sood, M. (2018). Segmentation of mr images using hybrid kmean-graph cut technique. *Procedia Computer Science*, 132, 775–784.

[32] Duan, Q. (Retrieved 2020). Densityclust (https://www.mathworks.com/matlabcentral/fileexchange/53922-densityclust). *MATLAB Central File Exchange*.

[33] Dunn, K. W., Fu, C., Ho, D. J., Lee, S., Han, S., Salama, P., & Delp, E. J. (2019). Deepsynth: Three-dimensional nuclear segmentation of biological images using neural networks trained with synthetic data. *Scientific Reports*, 9(1), 1–15.

[34] Eltanboly, A., Ghazal, M., Hajjdiab, H., Shalaby, A., Switala, A., Mahmoud, A., Sahoo, P., El-Azab, M., & El-Baz, A. (2019). Level sets-based image segmentation approach using statistical shape priors. *Applied Mathematics and Computation*, 340, 164–179.

[35] Eschweiler, D., Klose, T., Müller-Fouarge, F. N., Kopaczka, M., & Stegmaier, J. (2019a). Towards annotation-free segmentation of fluorescently labeled cell membranes in confocal microscopy images. In *International Workshop on Simulation and Synthesis in Medical Imaging* (pp. 81–89).

[36] Eschweiler, D., Spina, T. V., Choudhury, R. C., Meyerowitz, E., Cunha, A., & Stegmaier, J. (2019b). Cnn-based preprocessing to optimize watershed-based cell segmentation in 3d confocal microscopy images. In *2019 IEEE 16th International Symposium on Biomedical Imaging* (pp. 223–227).

[37] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96 (pp. 226–231).

[38] Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *Neurocomputing*, 321, 321–331.

[39] Fu, C., Lee, S., Joon Ho, D., Han, S., Salama, P., Dunn, K. W., & Delp, E. J. (2018). Three dimensional fluorescence microscopy image synthesis and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 2221–2229).

[40] Gahlmann, A. & Moerner, W. (2014). Exploring bacterial cell biology with single-molecule tracking and super-resolution imaging. *Nature Reviews Microbiology*, 12(1), 9.

[41] Gamarra, M., Zurek, E., Escalante, H. J., Hurtado, L., & San-Juan-Vergara, H. (2019). Split and merge watershed: A two-step method for cell segmentation in fluorescence microscopy images. *Biomedical Signal Processing and Control*, 53, 101575.

[42] Gibson, E., Li, W., Sudre, C., Fidon, L., Shakir, D. I., Wang, G., Eaton-Rosen, Z., Gray, R., Doel, T., Hu, Y., et al. (2018). Niftynet: a deep-learning platform for medical imaging. *Computer Methods and Programs in Biomedicine*, 158, 113–122.

[43] Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1440–1448).

[44] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139–144.

[45] Greig, D. M., Porteous, B. T., & Seheult, A. H. (1989). Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society: Series B (Methodological)*, 51(2), 271–279.

[46] Hafiz, A. M. & Bhat, G. M. (2020). A survey on instance segmentation: state of the art. *International Journal of Multimedia Information Retrieval*, (pp. 1–19).

[47] Hall-Stoodley, L., Costerton, J. W., & Stoodley, P. (2004). Bacterial biofilms: from the natural environment to infectious diseases. *Nature Reviews Microbiology*, 2(2), 95–108.

[48] Hartmann, R., Jeckel, H., Jelli, E., Singh, P. K., Vaidya, S., Bayer, M., Vidakovic, L., Díaz-Pascual, F., Fong, J. C., Dragos, A., et al. (2019a). Biofilmq, a software tool for quantitative image analysis of microbial biofilm communities. *BioRxiv*, (pp. 735423).

[49] Hartmann, R., Singh, P. K., Pearce, P., Mok, R., Song, B., Díaz-Pascual, F., Dunkel, J., & Drescher, K. (2019b). Emergence of three-dimensional order and structure in growing biofilms. *Nature Physics*, 15(3), 251–256.

[50] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).

[51] He, Y., Gong, H., Xiong, B., Xu, X., Li, A., Jiang, T., Sun, Q., Wang, S., Luo, Q., & Chen, S. (2015). icut: an integrative cut algorithm enables accurate segmentation of touching cells. *Scientific Reports*, 5(1), 1–17.

[52] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30.

[53] Hu, P., Shuai, B., Liu, J., & Wang, G. (2017). Deep level sets for salient object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 2300–2309).

[54] Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 1125–1134).

[55] Jermyn, I. H., Kurtek, S., Laga, H., & Srivastava, A. (2017). Elastic shape analysis of three-dimensional objects. *Synthesis Lectures on Computer Vision*, 12(1), 1–185.

[56] Jung, C. & Kim, C. (2010). Segmenting clustered nuclei using h-minima transform-based marker extraction and contour parameterization. *IEEE Transactions on Biomedical Engineering*, 57(10), 2600–2604.

[57] Kim, T., Kim, D., & Lee, S. (2020). Cell counting algorithm using radius variation, watershed and distance transform. *Journal of Information Processing Systems*, 16(1), 113–119.

[58] Kong, H., Gurcan, M., & Belkacem-Boussaid, K. (2011). Partitioning histopathological images: an integrated framework for supervised color-texture segmentation and cell splitting. *IEEE Transactions on Medical Imaging*, 30(9), 1661–1677.

[59] Kouzuma, A., Kasai, T., Hirose, A., & Watanabe, K. (2015). Catabolic and regulatory systems in shewanella oneidensis mr-1 involved in electricity generation in microbial fuel cells. *Frontiers in Microbiology*, 6, 609.

[60] Kowal, M., Żejmo, M., Skobel, M., Korbicz, J., & Monczak, R. (2020). Cell nuclei segmentation in cytological images using convolutional neural network and seeded watershed algorithm. *Journal of Digital Imaging*, 33(1), 231–242.

[61] Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2), 83–97.

[62] Laga, H., Xie, Q., Jermyn, I. H., & Srivastava, A. (2017). Numerical inversion of srnf maps for elastic shape analysis of genus-zero surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12), 2451–2464.

[63] Lee, S., Salama, P., Dunn, K. W., & Delp, E. J. (2017). Segmentation of fluorescence microscopy images using three dimensional active contours with inhomogeneity correction. In *IEEE 14th International Symposium on Biomedical Imaging* (pp. 709–713).: IEEE.

[64] Levinshtein, A., Stere, A., Kutulakos, K. N., Fleet, D. J., Dickinson, S. J., & Siddiqi, K. (2009). Turbopixels: Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12), 2290–2297.

[65] Li, B. & Acton, S. T. (2007). Active contour external force using vector field convolution for image segmentation. *IEEE Transactions on Image Processing*, 16(8), 2096–2106.

[66] Li, G., Liu, T., Tarokh, A., Nie, J., Guo, L., Mara, A., Holley, S., & Wong, S. T. (2007). 3d cell nuclei segmentation based on gradient flow tracking. *BMC Cell Biology*, 8(1), 40.

[67] Liang, H. & Weller, D. S. (2016). Comparison-based image quality assessment for selecting image restoration parameters. *IEEE Transactions on Image Processing*, 25(11), 5118–5130.

[68] Lindén, M., Ćurić, V., Boucharin, A., Fange, D., & Elf, J. (2016). Simulated single molecule microscopy with smeagol. *Bioinformatics*, 32(15), 2394–2395.

[69] Liu, D., Zhang, D., Song, Y., Huang, H., & Cai, W. (2020). Cell r-cnn v3: A novel panoptic paradigm for instance segmentation in biomedical images. *arXiv preprint arXiv:2002.06345*.

[70] Loader, C. (2006). *Local regression and likelihood*. Springer Science & Business Media.

[71] Logan, B. E. (2009). Exoelectrogenic bacteria that power microbial fuel cells. *Nature Reviews Microbiology*, 7(5), 375–381.

[72] Lu, X., Xie, Q., Zha, Y., & Wang, D. (2018). Fully automatic liver segmentation combining multi-dimensional graph cut with shape information in 3d ct images. *Scientific Reports*, 8(1), 1–9.

[73] Ly, T., Thompson, J., Harris, T., & Acton, S. T. (2018). The coupled tuff-bff algorithm for automatic 3d segmentation of microglia. In *2018 25th IEEE International Conference on Image Processing* (pp. 121–125).

[74] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1 (pp. 281–297).

[75] Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., & Paul Smolley, S. (2017). Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2794–2802).

[76] Matula, D. W. & Shahrokhi, F. (1990). Sparsest cuts and bottlenecks in graphs. *Discrete Applied Mathematics*, 27(1-2), 113–123.

[77] McQuin, C., Goodman, A., Chernyshev, V., Kamentsky, L., Cimini, B. A., Karhohs, K. W., Doan, M., Ding, L., Rafelski, S. M., Thirstrup, D., et al. (2018). Cellprofiler 3.0: Next-generation image processing for biology. *PLoS Biology*, 16(7), e2005970.

[78] Mittal, A., Moorthy, A. K., & Bovik, A. C. (2012). No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12), 4695–4708.

[79] Molnar, C., Jermyn, I. H., Kato, Z., Rahkama, V., Östling, P., Mikkonen, P., Pietiäinen, V., & Horvath, P. (2016). Accurate morphology preserving segmentation of overlapping cells based on active contours. *Scientific Reports*, 6, 32412.

[80] Mukherjee, S. & Acton, S. T. (2015). Region based segmentation in presence of intensity inhomogeneity using legendre polynomials. *IEEE Signal Processing Letters*, 22(3), 298–302.

[81] Mukherjee, S., Condron, B., & Acton, S. T. (2014). Tubularity flow field—a technique for automatic neuron segmentation. *IEEE Transactions on Image Processing*, 24(1), 374–389.

[82] Nadell, C. D., Drescher, K., & Foster, K. R. (2016). Spatial structure, cooperation and competition in biofilms. *Nature Reviews Microbiology*, 14(9), 589.

[83] Nerurkar, P., Shirke, A., Chandane, M., & Bhirud, S. (2018). Empirical analysis of data clustering algorithms. *Procedia Computer Science*, 125, 770–779.

[84] Osher, S. & Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1), 12–49.

[85] O'Toole, G., Kaplan, H. B., & Kolter, R. (2000). Biofilm formation as microbial development. *Annual Reviews in Microbiology*, 54(1), 49–79.

[86] Phoulady, H. A., Goldgof, D. B., Hall, L. O., & Mouton, P. R. (2016). Nucleus segmentation in histology images with hierarchical multilevel thresholding. *Medical Imaging 2016: Digital Pathology*, 9791(979111).

[87] Puchkov, E. (2019). Quantitative methods for single-cell analysis of microorganisms. *Microbiology*, 88(1), 1–14.

[88] Reyer, M. A., McLean, E. L., Chennakesavalu, S., & Fei, J. (2017). An automated image analysis method for segmenting fluorescent bacteria in three dimensions. *Biochemistry*, 57(2), 209–215.

[89] Rodriguez, A. & Laio, A. (2014). Clustering by fast search and find of density peaks. *Science*, 344(6191), 1492–1496.

[90] Rojas, D., Rueda, L., Ngom, A., Hurrutia, H., & Carcamo, G. (2011). Image segmentation of biofilm structures using optimal multi-level thresholding. *International Journal of Data Mining and Bioinformatics*, 5(3), 266–286.

[91] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 234–241).

[92] Roszkowiak, L., Korzynska, A., Pijanowska, D., Bosch, R., Lejeune, M., & Lopez, C. (2020). Clustered nuclei splitting based on recurrent distance transform in digital pathology images. *EURASIP Journal on Image and Video Processing*, 2020(1), 1–16.

[93] Rudge, T. J., Steiner, P. J., Phillips, A., & Haseloff, J. (2012). Computational modeling of synthetic microbial biofilms. *ACS Synthetic Biology*, 1(8), 345–352.

[94] Sadanandan, S. K., Ö. Baltekin, Magnusson, K. E. G., et al. (2016). Segmentation and track-analysis in time-lapse imaging of bacteria. *IEEE Journal of Selected Topics in Signal Processing*, 10(1), 174–184.

[95] Sandfort, V., Yan, K., Pickhardt, P. J., & Summers, R. M. (2019). Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in ct segmentation tasks. *Scientific Reports*, 9(1), 1–9.

[96] Schmidt, U., Weigert, M., Broaddus, C., & Myers, G. (2018). Cell detection with star-convex polygons. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 265–273).

[97] Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems*, 42(3), 1–21.

[98] Shen, S. P., Tseng, H., Hansen, K. R., Wu, R., Gritton, H. J., Si, J., & Han, X. (2018). Automatic cell segmentation by adaptive thresholding (acsat) for large-scale calcium imaging datasets. *Eneuro*, 5(5).

[99] Shi, J. & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 888–905.

[100] Shih, F. Y. & Pu, C. C. (1995). A skeletonization algorithm by maxima tracking on euclidean distance transform. *Pattern Recognition*, 28(3), 331–341.

[101] Shorten, C. & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 1–48.

[102] Shuvaev, S. A., Lazutkin, A. A., Kedrov, A. V., Anokhin, K. V., Enikolopov, G. N., & Koulakov, A. A. (2017). Dalmatian: an algorithm for automatic cell detection and counting in 3d. *Frontiers in Neuroanatomy*, 11, 117.

[103] Soille, P. (2013). *Morphological image analysis: principles and applications*. Springer Science & Business Media.

[104] Solomon, C. & Breckon, T. (2011). *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*. John Wiley & Sons.

[105] Solorzano, C., Malladi, R., Lelievre, S., & Lockett, S. (2001). Segmentation of nuclei and cells using membrane related protein markers. *Journal of Microscopy*, 201, 404–415.

[106] Song, Y., He, L., Zhou, F., Chen, S., Ni, D., Lei, B., & Wang, T. (2016). Segmentation, splitting, and classification of overlapping bacteria in microscope images for automatic bacterial vaginosis diagnosis. *IEEE Journal of Biomedical and Health Informatics*, 21(4), 1095–1104.

[107] Sørensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biol. Skr.*, 5, 1–34.

[108] Spilger, R., Schwackenhofer, T., Kaspar, C., Bischofs, I., & Rohr, K. (2020). Deep segmentation of bacteria at different stages of the life cycle. *Bildverarbeitung für die Medizin*, (pp. 8–13).

[109] Srivastava, A., Klassen, E., Joshi, S. H., & Jermyn, I. H. (2010). Shape analysis of elastic curves in euclidean spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7), 1415–1428.

[110] Stringer, C., Wang, T., Michaelos, M., & Pachitariu, M. (2021). Cellpose: a generalist algorithm for cellular segmentation. *Nature Methods*, 18(1), 100–106.

[111] Stuart, T. & Satija, R. (2019). Integrative single-cell analysis. *Nature Reviews Genetics*, 20(5), 257–272.

[112] Svoboda, D. & Ulman, V. (2016). Mitogen: a framework for generating 3d synthetic time-lapse sequences of cell populations in fluorescence microscopy. *IEEE Transactions on Medical Imaging*, 36(1), 310–321.

[113] Tabassum, N., Vaccari, A., & Acton, S. T. (2015). Speckle removal by distance-driven anisotropic diffusion of sar temporal stacks. In *2015 49th Asilomar Conference on Signals, Systems and Computers* (pp. 1584–1588).

[114] Tabassum, N., Wang, J., Ferguson, M., Herz, J., Dong, M., Louveau, A., Kipnis, J., & Acton, S. T. (2021). Image segmentation for neuroscience: lymphatics. *JPhys Photonics*, 3(3).

[115] Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500), 2319–2323.

[116] Van Valen, D. A., Kudo, T., Lane, K. M., Macklin, D. N., Quach, N. T., DeFelice, M. M., Maayan, I., Tanouchi, Y., Ashley, E. A., & Covert, M. W. (2016). Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. *PLoS Computational Biology*, 12(11), e1005177.

[117] Vincent, L. & Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6), 583–598.

[118] Voleti, V., Patel, K. B., Li, W., Campos, C. P., Bharadwaj, S., Yu, H., Ford, C., Casper, M. J., Yan, R. W., Liang, W., et al. (2019). Real-time volumetric microscopy of in vivo dynamics and large-scale samples with scape 2.0. *Nature Methods*, 16(10), 1054–1062.

[119] Wang, C., Oda, M., Hayashi, Y., Yoshino, Y., Yamamoto, T., Frangi, A. F., & Mori, K. (2020). Tensor-cut: A tensor-based graph-cut blood vessel segmentation method and its application to renal artery segmentation. *Medical Image Analysis*, 60, 101623.

[120] Wang, J., Batabyal, T., Zhang, M., Zhang, J., Aziz, A., Gahlmann, A., & Acton, S. T. (2019). Lcuts: Linear clustering of bacteria using recursive graph cuts. In *2019 IEEE International Conference on Image Processing* (pp. 1575–1579).

[121] Wang, J., Fu, Z., Sadeghzadehyazdi, N., Kipnis, J., & Acton, S. T. (2018a). Nonlinear shape regression for filtering segmentation results from calcium imaging. In *2018 25th IEEE International Conference on Image Processing* (pp. 738–742).

[122] Wang, J., Sarkar, R., Aziz, A., Vaccari, A., Gahlmann, A., & Acton, S. T. (2017). Bact-3d: A level set segmentation approach for dense multi-layered 3d bacterial biofilms. *2017 IEEE International Conference on Image Processing*, (pp. 330–334).

[123] Wang, J., Tabassum, N., Wang, Y., Gahlmann, A., & Acton, S. T. (2021a). 3d gan image synthesis and dataset quality assessment for bacterial biofilm. *(submitted)*.

[124] Wang, J., Zhang, M., Zhang, J., Wang, Y., Gahlmann, A., & Acton, S. T. (2021b). Graph-theoretic post-processing of segmentation with application to dense biofilms. *IEEE Transaction on Image Processing (in revision)*.

[125] Wang, X., Allen, W. E., Wright, M. A., Sylwestrak, E. L., Samusik, N., Vesuna, S., Evans, K., Liu, C., Ramakrishnan, C., Liu, J., et al. (2018b). Three-dimensional intact-tissue sequencing of single-cell transcriptional states. *Science*, 361(6400).

[126] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612.

[127] Weigert, M., Schmidt, U., Haase, R., Sugawara, K., & Myers, G. (2020). Star-convex polyhedra for 3d object detection and segmentation in microscopy. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 3666–3673).

[128] Whitman, W. B., Coleman, D. C., & Wiebe, W. J. (1998). Prokaryotes: the unseen majority. *Proceedings of the National Academy of Sciences*, 95(12), 6578–6583.

[129] Wollmann, T., Gunkel, M., Chung, I., Erfle, H., Rippe, K., & Rohr, K. (2019). Gruunet: Integrated convolutional and gated recurrent neural network for cell segmentation. *Medical image analysis*, 56, 68–79.

[130] Wu, Z. & Leahy, R. (1993). An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11), 1101–1113.

[131] Yan, J. & Bassler, B. L. (2019). Surviving as a community: antibiotic tolerance and persistence in bacterial biofilms. *Cell Host & Microbe*, 26(1), 15–21.

[132] Yan, J., Sharo, A. G., Stone, H. A., Wingreen, N. S., & Bassler, B. L. (2016). Vibrio cholerae biofilm growth program and architecture revealed by single-cell live imaging. *Proceedings of the National Academy of Sciences*, 113(36), E5337–E5343.

[133] Yan, T.-Q. & Zhou, C.-X. (2012). A continuous skeletonization method based on distance transform. In *International Conference on Intelligent Computing* (pp. 251–258).

[134] Yi, F. & Moon, I. (2012). Image segmentation: A survey of graph-cut methods. In *2012 International Conference on Systems and Informatics* (pp. 1936–1941).

[135] Yi, J., Wu, P., Jiang, M., Hoeppner, D. J., & Metaxas, D. N. (2018). Instance segmentation of neural cells. *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*.

[136] Yu, Y. & Acton, S. T. (2002). Speckle reducing anisotropic diffusion. *IEEE Transactions on Image Processing*, 11(11), 1260–1270.

[137] Zhang, M., Zhang, J., Wang, Y., Wang, J., Achimovich, A. M., Acton, S. T., & Gahlmann, A. (2020). Non-invasive single-cell morphometry in living bacterial biofilms. *Nature Communications*, 11(1), 1–13.

[138] Zhang, Q., Wang, T., Zhou, Q., Zhang, P., Gong, Y., Gou, H., Xu, J., & Ma, B. (2017). Development of a facile droplet-based single-cell isolation platform for cultivation and genomic analysis in microorganisms. *Scientific Reports*, 7, 41192.

[139] Zhang, Z., Yang, L., & Zheng, Y. (2018). Translating and segmenting multimodal medical volumes with cycle-and shape-consistency generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 9242–9251).

[140] Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2223–2232).

THANK YOU FOR READING THIS DISSERTATION. It is a summary of my 20's with the best Virginia Image and Video Analysis Laboratory, in this lovely Charlottesville. Words are powerless at this moment to express my gratitude to everyone and everything I encountered. I treasure these priceless gains and thank to the growing in the losses. No matter how my life was or how it will be, I hope I can always stay positive and hopeful, with a pure grateful heart.

Sincerely,
Jie Wang