Faculty of
Mathematics, Computer Science
and Natural Sciences

RWTH AACHEN UNIVERSITY

Geometry-aware Image Analysis for Microfluidic Live-cell Experimentation

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades einer Doktorin der Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von

Karina Ruzaeva
Master of Science (M.Sc.)
aus Iwantejewka, Russland

Berichter: Prof. Dr. rer. nat. Benjamin Berkels
Prof. Dr. rer. nat. Thomas Berlage
Prof. Dr. rer. nat. Wolfgang Wiechert

Tag der mündlichen Prüfung: 12.02.2025

Abstract

Time-lapse microscopy combined with advanced imaging techniques offers new opportunities to approach fundamental and applied biological questions. Specifically, microfluidic tools enable the study of living cells under precisely controlled environmental conditions (medium, temperature, light) and allow for their observation through live-cell microscopy. This approach enables the capture of time-resolved cellular responses, which are recorded as high-volume time-lapse image sequences. Processing these large-scale image sequences poses numerous challenges for image analysis, such as varying noise levels, low-intensity gradients and limited image capture rates. Addressing these challenges is crucial for developing biotechnological processes as it becomes increasingly necessary better to understand the behavior of microorganisms at the single-cell level. Advanced image-processing techniques play a critical role in extracting valuable insights from these datasets, which are essential for optimizing biotechnological applications.

The dissertation "Geometry-aware image analysis for microfluidic live-cell experimentation" emphasizes the importance of incorporating prior knowledge of microorganisms' geometry and behavior—specifically their shape, size, and division mechanisms—into image analysis techniques. In this dissertation, we present an image processing workflow comprising ground truth data generation, segmentation, and tracking. This ground truth generation and subsequent segmentation and tracking algorithms are informed by the geometry and behavioral characteristics of cells, which are determined by the selected microorganism.

To implement this workflow, this dissertation proposes ground truth data generation methods that include both synthetic image simulations and the processing of annotations from real data. For the segmentation task, we combine geometry-aware variational spline-based segmentation with machine learning-based detection to enhance the accuracy of cell identification. This approach is complemented by activity-based tracking that monitors cell behavior over time, enabling the extraction of critical parameters such as cell size, count, and dynamic behavior. The extracted data can be used to dynamically adjust bioprocess conditions to optimize growth and yield, leading to greater efficiency and productivity in biotechnological processes.

By integrating this geometry and behavior-aware image processing methods, including ground truth generation, geometry-aware segmentation, and activity-based tracking, this dissertation underscores the potential of precise image analysis in enhancing live-cell experimentation. The results not only improve the accuracy of single-cell analysis but also help to optimize biotechnological processes by understanding cellular behavior. The graphical abstract illustrating this workflow is presented in Figure 1.

Zusammenfassung

Die Zeitraffer-Mikroskopie in Kombination mit fortschrittlichen Bildgebungstechniken eröffnet neue Möglichkeiten, grundlegende und angewandte biologische Fragestellungen anzugehen. Insbesondere ermöglichen mikrofluidische Werkzeuge, die Entwicklung

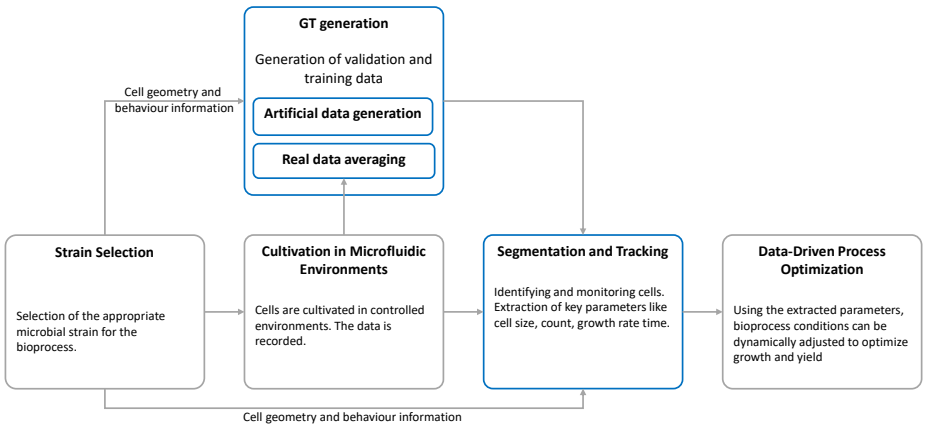


Figure 1.: Graphical Abstract. Integration of the proposed geometry-aware image analysis methods into the biotechnological process. The image analysis blocks are highlighted in blue.

lebender Zellen unter präzisen Umgebungsbedingungen (Medium, Temperatur, Licht) zu untersuchen und mittels Lebendzell-Mikroskopie aufzuzeichnen. Diese Methode ermöglicht die Erfassung zeitaufgelöster Zellantworten, die als hochvolumige Zeitraffer-Bildsequenzen aufgezeichnet werden. Die Verarbeitung der anfallenden hochvolumigen Bildsequenzen stellt zahlreiche Herausforderungen an die Bildanalyse, wie etwa variierende Rauschpegel, geringe Intensitätsgradienten und begrenzte Bildaufnahmeraten. Die Bewältigung dieser Herausforderungen ist entscheidend, da es für die Entwicklung biotechnologischer Prozesse zunehmend notwendig wird, das Verhalten von Mikroorganismen auf Einzelzellebene besser zu verstehen. Hier spielen fortschrittliche Bildverarbeitungstechniken eine entscheidende Rolle, um wertvolle Einblicke zu gewinnen, die für die Optimierung biotechnologischer Anwendungen unerlässlich sind.

Die Dissertation “Geometriebasierte Bildanalyse für mikrofluidische Lebendzell-Experimente” betont die Bedeutung der Einbeziehung von Vorwissen über die Geometrie und das Verhalten von Mikroorganismen—insbesondere ihrer Form, Größe und Teilungsmechanismen— in Bildanalysetechniken. In dieser Arbeit wird ein Bildverarbeitungs-Workflow vorgestellt, der die Generierung von Ground-Truth-Daten, Segmentierung und Nachverfolgung umfasst. Diese Ground-Truth-Generierung sowie die nachfolgende Segmentierungs- und Tracking-Algorithmen werden durch die geometrischen und verhaltensspezifischen Merkmale der Zellen, die durch den ausgewählten Mikroorganismus bestimmt werden, informiert.

Um diesen Workflow umzusetzen, schlägt diese Dissertation Methoden zur Generierung von Ground-Truth-Daten vor, die sowohl synthetische Bildsimulationen als auch die Prozessierung von Annotationen realer Daten umfassen. Für die Segmentierung kombinieren wir geometrie-bewusste variationale Spline-basierte Segmentierung mit maschinellem Lernen zur Erkennung, um die Genauigkeit der Zellerkennung zu verbessern. Dieser Ansatz wird durch ein aktivitätsbasiertes Tracking ergänzt, das das Verhalten von Zellen im Zeitverlauf überwacht und es ermöglicht, kritische Parameter wie Zellgröße,

Zellzahl und dynamisches Verhalten zu extrahieren. Die extrahierten Daten können zur dynamischen Anpassung der Bioprozessbedingungen verwendet werden, um das Wachstum und den Ertrag zu optimieren, was zu einer höheren Effizienz und Produktivität der biotechnologischen Prozesse führt.

Durch die Integration dieser geometrie- und verhaltensbewussten Bildverarbeitungsmethoden, einschließlich der Ground-Truth-Generierung, geometrie-bewusster Segmentierung und aktivitätsbasierter Nachverfolgung, unterstreicht diese Dissertation das Potenzial einer präzisen Bildanalyse zur Verbesserung von Lebendzell-Experimenten. Die Ergebnisse verbessern nicht nur die Genauigkeit der Einzelzell-Analyse, sondern tragen auch zur Optimierung biotechnologischer Prozesse bei, indem sie das zelluläre Verhalten besser verstehen. Die grafische Zusammenfassung, die diesen Workflow veranschaulicht, wird in Abbildung 1 dargestellt.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to Prof. Benjamin Berkels and Dr. Katharina Nöh for their exceptional supervision and guidance throughout my dissertation. Their expertise, patience, and encouragement were invaluable to the completion of my work.

I am deeply grateful to the HDS-LEE Graduate School for funding my research. Special thanks also go to Prof. Wolfgang Wiechert for giving me the opportunity to join both the graduate school and the IBG-1 Institute at Forschungszentrum Jülich, providing an excellent working environment.

I would like to extend my thanks to Prof. Thomas Berlage for kindly agreeing to review my dissertation, and to Prof. Johannes Stegmaier for serving as an additional reviewer. My sincere appreciation also goes to Prof. Arnold Reusken for chairing my defense.

Furthermore, I would like to thank my friends and family for their unwavering support and motivation throughout this journey. I am also grateful to all my colleagues at AICES, RWTH Aachen, and the Biotechnology Group (IBG-1) at Forschungszentrum Jülich for their collaboration and friendship.

This dissertation would not have been possible without the contributions and encouragement of everyone mentioned above, and for that, I am truly thankful.

ChatGPT (OpenAI) and Grammarly were used to assist in rephrasing and correcting grammatical errors. These tools were employed solely to enhance the clarity and readability of the text.

Faculty of
Mathematics, Computer Science
and Natural Sciences

RWTH AACHEN UNIVERSITY

Eidesstattliche Erklärung
Declaration of Authorship

I, Karina Ruzaeva, declare that this thesis and the work presented in it are my own and have been generated by me as the result of my own original research.
Hiermit erkläre ich an Eides statt / I do solemnly swear that:

1. This work was done wholly or mainly while in candidature for the doctoral degree at this faculty and university;

2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated.

3. Where I have consulted the published work of others or myself, this is always clearly attributed

4. Where I have quoted from the work of others or myself, the source is always given. This thesis is entirely my own work, with the exception of such quotations;

5. I have acknowledged all major sources of assistance;

6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;

7. Parts of this work have been published before as listed in Section Publications.

Contents

Outcome

Publications

This thesis is based on research performed by the author during her time at the Aachen Institute for Advanced Studies in Computational Engineering Science (AICES) at RWTH Aachen University, Germany, and the Institute of Biotechnology (IBG-1) at Forschungszentrum Juelich, between October 2019 and October 2022. Additionally, the author was part of the Helmholtz Graduate School for Data Science in Life, Earth, and Energy (HDS-LEE). Major parts of this thesis have appeared in previous publications and are reproduced here, in restructured, modified and/or extended form. In particular,

(i) **Ruzaeva, K.**, Nöh, K., & Berkels, B. (2021). Polar Space Based Shape Averaging for Star-shaped Biological Objects. Eurographics Workshop on Visual Computing for Biology and Medicine. https://doi.org/10.2312/VCBM.20211340

(ii) **Ruzaeva, K.**, Nöh, K., & Berkels, B. (2022). A Hybrid Multi-Object Segmentation Framework with Model-Based B-Splines for Microbial Single Cell Analysis. In 2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI). 2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI). IEEE. https://doi.org/10.1109/isbi52829.2022.9761575

(iii) **Ruzaeva, K.**, Kusters, K., Wiechert, W., Berkels, B., Oldiges, M., & Nöh, K. (2022). Automated Characterization of Catalytically Active Inclusion Body Production in Biotechnological Screening Systems. In 2022 44th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE. https://doi.org/10.1109/embc48229.2022.9871325

(iv) Sachs, C. C., **Ruzaeva, K.**†, Seiffarth, J.†, Wiechert, W., Berkels, B., & Nöh, K. (2022). CellSium: versatile cell simulator for microcolony ground truth generation. In A. Bateman (Ed.), Bioinformatics Advances (Vol. 2, Issue 1). Oxford University Press (OUP). https://doi.org/10.1093/bioadv/vbac053

(v) **Ruzaeva, K.**, Cohrs, J.-C., Kasahara, K., Kohlheyer, D., Nöh, K., & Berkels, B. (2022). Cell tracking for live-cell microscopy using an activity-prioritized assignment strategy. In 2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS). 2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS). IEEE. https://doi.org/10.1109/ipas55744.2022.10053036

(vi) Kasahara, K., Leygeber, M., Seiffarth, J., **Ruzaeva, K.**, Drepper, T., Nöh, K., & Kohlheyer, D. (2023). Enabling oxygen-controlled microfluidic cultures for spatiotemporal microbial single-cell analysis. In Frontiers in Microbiology (Vol. 14). Frontiers Media SA. `https://doi.org/10.3389/fmicb.2023.1198170`

† The authors contributed equally to this work.

Conference contributions

(i) **Ruzaeva, K.**, Nöh, K., Berkels, B. Polar Space Based Shape Averaging for Star-shaped Biological Objects (Presentation) 11th EG Workshop on Visual Computing for Biology and Medicine, VCBM 2021

(ii) **Ruzaeva, K.**, Küsters, K., Oldiges, M., Berkels, B., Nöh, K. A hybrid approach for automated characterization of CatIBs production in a biotechnological screening system. (Poster) BioImage Informatics 2021, Institut Pasteur, France, 2021.

(iii) **Ruzaeva, K.**, Nöh, K., Berkels, B. Geometry-aware image analysis for microfluidic live-cell experimentation (Scientific telegram) The first HIDA Annual Conference 2021

(iv) **Ruzaeva, K.**, Nöh, K., Berkels, B. A hybrid multi-object segmentation framework with model-based B-splines for single cell analysis (Poster) ISBI 2022: IEEE International Symposium on Biomedical Imaging

(v) **Ruzaeva, K.**, Küsters, K., Oldiges, M., Berkels, B., Nöh, K. Automated Characterization of Catalytically Active Inclusion Body Production in Biotechnological Screening Systems (E-poster) EMBC 2022: Annual International Conference of the IEEE Engineering in Medicine and Biology Society

(vi) **Ruzaeva, K.**, Cohrs J.-C., Kasahara, K., Kohlheyer, D., K., Berkels, B. Cell tracking for live-cell microscopy using an activity-prioritized assignment strategy (Presentation) IPAS 2022: IEEE International Image Processing Applications and Systems (*Best paper award of the session "Scene Analysis for Activity Recognition"*)

Acronyms

C. glutamicum *Corynebacterium glutamicum*

E. coli *Escherichia coli*

S. cerevisiae *Saccharomyces cerevisiae*

AM Activity Map

AMD Average Multi-object Dice

CatIB Catalytically active Inclusion Body

FD Foreground Dice

GFP Green Fluorescent Protein

GT Ground Truth

HIPP Hybrid Image Processing Pipeline

IoU Intersection over Union

LoC Lab-on-Chip

mAP mean Average Precision

ML Machine Learning

NMS Non Maximum Suppression

PCC Pearson Correlation Coefficient

RMSE Root-Mean-Square Error

STAPLE Simultaneous Truth And Performance Level Estimation

WRMSE Weighted Root-Mean-Square Error

Thesis outline

The dissertation is organized into four distinct parts: Introduction (Chapter 1 and Chapter 2), Ground truth generation (Chapter 3 and Chapter 4), Segmentation (Chapter 5 and Chapter 6) and Tracking (Chapter 7).

Chapter 1 provides a general introduction to biotechnology and the various scales of biotechnological processes. It discusses the experimental pipelines, the target microorganisms, and the role of live-cell imaging and microfluidics in biotechnology.

Chapter 2 covers image processing methods used in microfluidic single-cell datasets. This chapter provides an overview of both model-based and data-driven image processing methods and also delves into the technical details of the data that we aim to analyze in the following thesis.

The following two Chapters 3 and 4 discuss the Ground Truth (GT) generation problem, which is a crucial step in analyzing single-cell data. In particular, in Chapter 3, a method that creates GT by averaging real data is discussed, while Chapter 4 describes software for artificial GT generation.

Chapters 4 and 5 focus on cell segmentation, with Chapter 4 discussing the segmentation problem in Lab-on-Chip devices and Chapter 5 discussing the segmentation problem in flow chambers. Chapter 6 provides a detailed description of the proposed activity-based tracking method used for time analysis of cell colonies. By leveraging these image analysis tools, we aim to provide insights into the behavior of individual cells in microfluidic environments, with potential applications in various fields, including biotechnology and biomedical research. Finally, Chapter 7 draws overall conclusions and summarizes the findings. The thesis is a useful resource for researchers and students interested in microfluidic single-cell analysis, providing a comprehensive overview of the various techniques and methods used in this field.

Given that all chapters are built upon previously published articles, it is important to acknowledge the occasional redundancy in the content. This allows reading the chapters independently.

Part I.

Introduction

1. Introduction into biotechnology

Industrial biotechnology employs microorganisms to produce chemicals, pharmaceuticals, biofuels, and amino acids. It includes using whole microorganisms or cell-free synthesis to generate industrially relevant products more efficiently (e.g., less energy or fewer by-products production) or creating substances and chemical building blocks with specific capabilities that conventional processes cannot provide. That makes industrial biotechnology one of the most promising new approaches to pollution prevention, resource conservation, and cost reduction, offering businesses a way to reduce costs and create new markets while protecting the environment at the same time [1].

There is a solid demand to improve existing bio-based production processes further, find new ways to use sustainable substrates, and sustainably produce new biocatalysts. This is complex as production dramatically depends on the organism used, whose output can vary due to the stochastic nature of the process itself, converting a substrate into the desired product.

There is a strong interest in the efficient and cost-effective biotechnological production of bulk chemicals (fuels, solvents, plastics) from renewable plant biomass. There is a long tradition of microbial fermentations in food applications, such as producing organic acids, alcohols, amino acids, and vitamins. Still, many microbial products can also be found in other segments, such as antibiotics. Strain development (i.e., the improvement of microbial production strains) and the continuous improvement of biobased production processes are important and essential for optimizing biotechnological production processes [2, 3].

1.1. Bioprocess development

The classical development pipeline of novel bioprocesses typically follows a four-stage workflow:

Stage 1 The creation and identification of the production strain that produces the compound of interest is performed at laboratory-scale experiments.

Stage 2 Once a suitable strain has been identified, the production of the desired compound is enhanced using different microbial and molecular biological techniques.

Stage 3 The growth media composition and cultivation conditions are optimized. This stage involves optimizing various parameters, such as temperature, pH, and oxygen supply, to maximize product yield and minimize costs.

Stage 4 The process gets transferred from the laboratory scale to the production scale [4, 3].

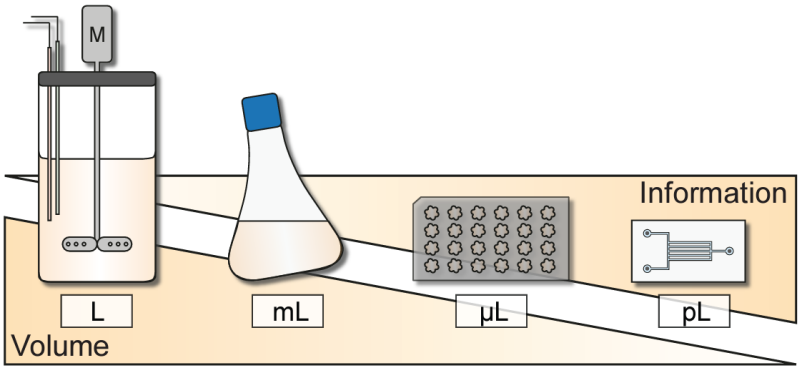


Figure 1.1.: Different scales of microbial analysis. Left to right: Lab-scale bioreactor, Shaking flask, Microtiter plates, Microfluidic Lab-on-Chip (LoC) devices. The figure is taken from [5].

The four-stage workflow in bioprocess development provides a structured approach to developing and optimizing a bioprocess for producing a desired product by minimizing development time and costs while maximizing product yield and quality. The scale-up is still one of the significant issues in bioprocess development. The different laboratory scales of the bioprocess will be discussed in the next sections.

1.1.1. Different scales of the bioprocess development

As mentioned above, laboratory-scale experiments are essential for bioprocess design. Different laboratory-scale approaches can be used to develop and optimize bioprocesses (Figure 1.1).

(i) **Bioreactor** A laboratory-scale (liter-scale) bioreactor is a device used in a laboratory setting to cultivate and grow microorganisms or cells for various applications. It provides a controlled environment for growing cells or microorganisms by regulating multiple parameters such as temperature, pH, nutrient supply, and oxygen availability. Bioreactors play an essential role in microbiological, biotechnological, and biochemical research. They have a high level of automation and control, reducing the need for manual intervention and increasing the reproducibility of results.

Small-volume bioreactors enable biomass growth, extracellular metabolite production, solid-state fermentation, and biocatalytic processes with increased experimental throughput compared to larger-scale bioreactors. This allows for more experiments to be conducted in a shorter time than larger-scale bioreactors.

The lab-scale (liter-scale) bioreactor processes can be scaled up to a pilot level or scaled down to study the impact of the heterogeneity of the system performance and its effect on the producer. Given their geometric similarity to large-scale production plants and the excellent possibilities for precise process control and monitoring, they still represent the gold standard for bioprocess development at a laboratory

scale. However, the large (liters) volume makes obtaining detailed information on individual cells or microscale interactions challenging [6].

(ii) **Shake flask**

For the past decades, scientists have used cell cultivation in shake flasks in the bioprocess development on a small scale, with volumes ranging from ca. 10 ml to 500 ml. They have been estimated to be used for over 90% of all culture experiments across industry and academia, cultivating a wide range of microorganisms, e.g., bacteria, fungi, and yeasts, as well as mammalian cells in suspension [7, 8].

Shake flasks come in a variety of shapes and materials. Factors that affect shake flask cultivations include vessel size and geometry, fill volume, construction material, shaking frequency, and type of plug used to seal the vessel. It is easy to see why they are so widely used: they are an inexpensive and effective way of reproducibly performing many industrially-relevant cell cultivation tasks. However, significant inter-flask variations, often observed in Erlenmeyer flask cultivations, make comparing product levels during fermentation difficult and less reliable [9].

They are easy to use and inexpensive, making them suitable for basic applications such as organism screening, media design, and early process development. This very simplicity, however, is the most significant disadvantage. Shake flasks do not provide detailed information about culture performance or the opportunity to monitor and control the process parameters. Therefore, cultivation processes in shake flasks are never under accurate control. These systems do not allow online measurements of critical process parameters such as temperature, oxygen levels, or optical density because the flasks need to be equipped with sets of sensors. Direct control of these parameters is also impossible. Controlling a bioprocess in shake flasks would involve significant manual labor, such as taking samples, making offline measurements, and adjusting levels manually. Scaling up a bioprocess requires bioreactors that allow for predictable scalability. Nevertheless, shake flasks are suitable for screening, media optimization, and early process development [10]. But, like in bioreactors, detailed information on individual cells is limited due to the larger volume.

(iii) **Microtiter plates**

Microtiter plates also referred to as microplates or multiwell plates, are essential instruments in laboratory settings that facilitate a wide range of experimental procedures and assays. These plates comprise numerous small wells, typically made of plastic or glass, which can be sealed to protect samples from contamination and evaporation. Designed to accommodate small volumes of liquid, they are well-suited for conducting multiple experiments in parallel. Depending on the experimental requirements, the number of wells can range from a few to several hundred.

Various biotechnological workflows are now established and automated in microtiter plates. What makes them so valuable is their ability to allow the simultaneous analysis of many samples in a single experiment, which greatly boosts efficiency and productivity in the lab.

For example, the BioLector system enables 48 parallel cultivations with online monitoring of various population bioprocess variables, such as biomass, dissolved oxygen, pH, and fluorescence. Specialized microtiter plates, like FlowerPlates®, are tailored for high-throughput screening of microbial strains [11, 12, 6].

While microtiter plates reduce the volume compared to bioreactors and shaking flasks, single-cell analysis is limited, but the integration of a robotic platform for handling the experiments in parallel and flow chamber microscopy experiments can unlock the potential for high-throughput single-cell analysis (described in detail in Section 1.2.2).

(iv) **Lab-on-chip devices**

Lab-on-Chip (LoC) devices are miniaturized platforms integrating multiple laboratory functions on a single chip. LoC devices operate at a microscale, typically in the range of microliters to picoliters [13]. They provide precise control over microenvironments: fluid flow, temperature, oxygen concentration, and nutrient delivery, being able to mimic the natural conditions in which the cells grow while also allowing for real-time single-cell analysis [14].

Microfluidic devices can capture individual cell behavior, interactions, and responses to dynamic conditions. They are designed to perform various analytical and biological tests using small sample volumes, making them particularly useful in high-throughput cell screens [15, 16, 17].

In these cultivation devices, the morphology of microbes—specifically their geometry—plays a crucial role in determining their productivity. In these cultivation devices, the morphology of microbes—specifically their geometry—plays a crucial role in determining their productivity. However interestingly, even in genetically identical populations (isogenic), significant morphological diversity can be observed. Differences in environmental conditions can cause diversity within a microbial population, manifesting as variations in cell shape and splitting behavior. These morphological changes, in turn, lead to differences in productivity within the same population. Understanding and tracking these differences at the single-cell level is essential for optimizing bioprocesses.

As the scale decreases, there is an opportunity to gain more detailed information at the level of individual cells or microscale interactions (Figure 1.1). Microfluidic LoC devices provide the unique advantage of single-cell analysis, such as single-cell growth tracking, which can be challenging or impossible at the other three scales, making them well-suited for studying cell heterogeneity and microscale phenomena. However, it's important to note that each scale, from macroscale to microscale, offers distinct advantages and limitations for research. The choice of scale should align with the research objectives—whether the goal is to study broad population-level dynamics at the macroscale or to focus on detailed cellular interactions and heterogeneity at the microscale. Researchers must choose the appropriate scale based on their specific research goals and the level of detail required for their analysis.

1.2. Live-cell imaging and microfluidic cultivation in biotechnology

Live-cell imaging is a crucial technique that allows us to observe living microorganisms in real time, providing invaluable insights into their behavior, morphology, and interactions within various environments. With advanced microscopes, we can image living specimens under relevant conditions, capturing the behavior and morphology of microorganisms. Automated live-cell imaging systems are employed to monitor microbial behavior and morphology effectively. These systems can be integrated with different cultivation platforms, such as LoC devices and Robotic platform with flow cell chamber setup, which are described below, each offering unique capabilities. The robotic platform allows for the imaging of larger volumes (milliliters), making them suitable for studying microbial populations in more complex, flowing environments. On the other hand, LoCs operate at much smaller scales (nanoliters to picoliters), providing high-throughput capabilities and finer control over environmental conditions.

In this work, we focus on both flow cell chambers and LoCs to leverage their respective strengths. Flow cell chambers are advantageous for their ability to simulate more natural, dynamic environments where microbes can be observed in a continuous flow. This is particularly useful for studying how microbial communities behave under conditions that mimic industrial bioprocesses. Conversely, LoCs offer a highly controlled, miniaturized environment where individual cells can be isolated and observed with great precision. This makes them ideal for single-cell analysis, where understanding the subtle differences in morphology and behavior among seemingly identical cells can provide insights into population heterogeneity.

Each of these systems, LoC devices, and the robotic platform, equipped with the flow chamber, presents specific challenges in image analysis, which will be tackled in the following chapters of this dissertation. LoCs, while providing high-resolution data, require precise segmentation techniques to accurately identify and track cells in densely packed, confined spaces. Flow cell chambers, with their larger volumes and flowing cells, generate complex datasets where distinguishing individual in-focus cells can be difficult, especially under varying flow rates.

1.2.1. LoC devices

LoC devices have become important tools for studying single-cell behavior and cellular dynamics with remarkable precision and high throughput. These small, microfabricated devices combine microfluidic technology, which allows for carefully handling fluids in small volumes—from microliters down to picoliters—and advanced live-cell imaging techniques.

Microfluidic LoC devices were initially developed to miniaturize standard laboratory processes, condensing multiple functionalities onto a single chip the size of a coin. These chips consist of intricate microfluidic channels and chambers, which allow for the controlled delivery of fluids to cells or samples. Polydimethylsiloxane (PDMS) is a commonly used material for fabricating microfluidic chips due to its biocompatibility and

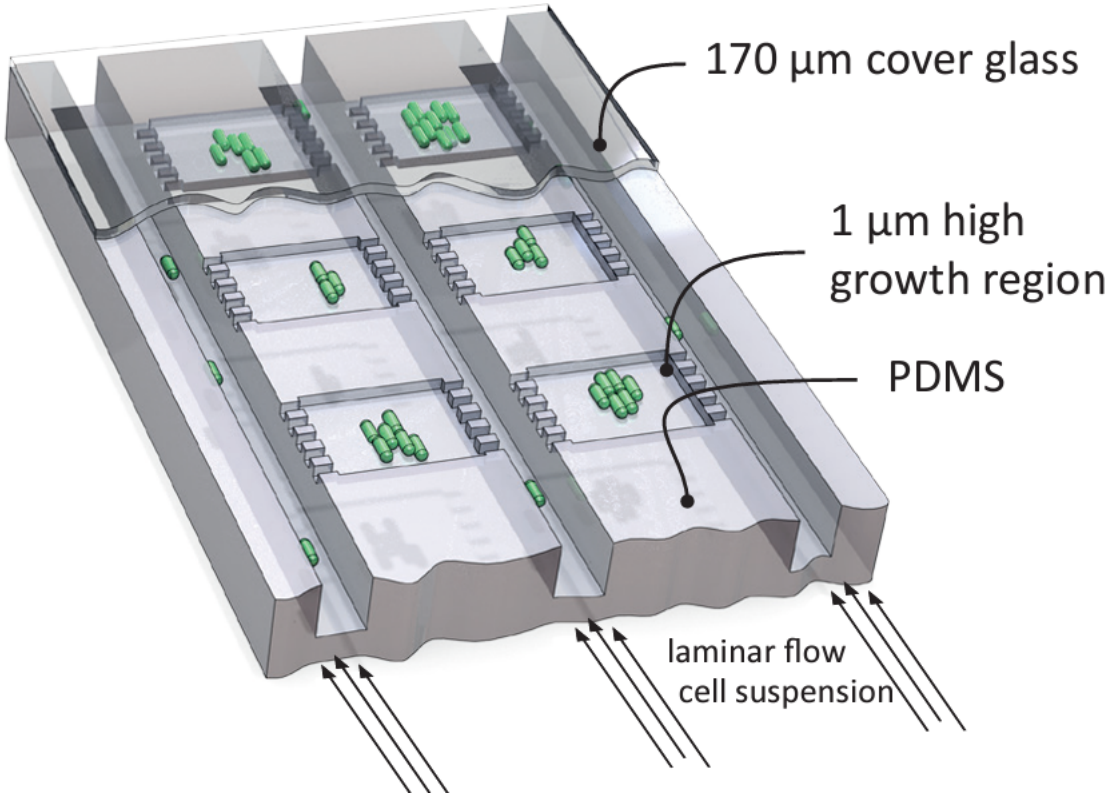


Figure 1.2.: Microfluidic growth chambers for bacterial cultivation. Figure is taken from[5].

ease of fabrication. The remainder of this chapter discusses the fundamental principles of microfluidics and the design considerations for constructing LoC devices to support single-cell analysis. A typical LoC device is illustrated in Figure 1.2 [5, 17].

One of the most significant advantages of LoC devices is their capacity for parallelization, as they typically house hundreds of microfluidic chambers on a single chip. This feature allows researchers to run many experiments in parallel, enhancing throughput and efficiency. Each microfluidic chamber can accommodate individual cells or samples, enabling researchers to study cell behavior in high resolution while analyzing cell-to-cell heterogeneity within a population [18].

Cell-to-cell heterogeneity, referring to the variations in behavior and responses among individual cells within a population, is a critical aspect of cellular dynamics. Traditional bulk measurements can overlook these nuances, delivering average values of process parameters such as the growth and production of billions of cells. Misleadingly, the average response of cells was, and still is, interpreted as the response of every single cell within a sample. The cell-to-cell heterogeneity of the microbial population remains hidden inside the obtained average. However, live-cell imaging in microfluidic LoC devices allows researchers to observe and track individual cells over time, unveiling heterogeneity within cell populations (Figure 1.3) [19, 3].

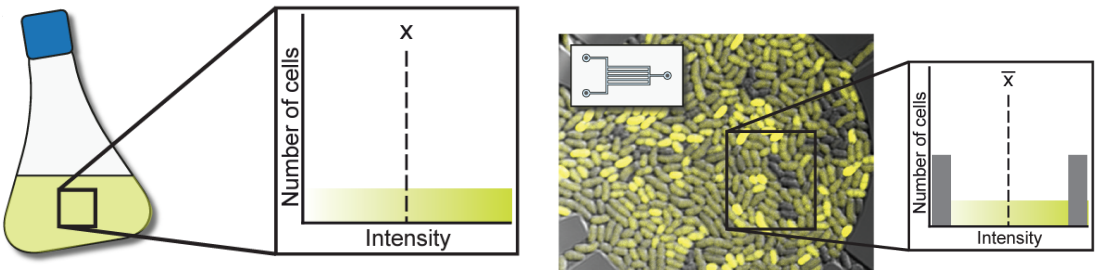


Figure 1.3.: Population heterogeneity at different resolutions. Bulk measurements extract one averaged fluorescence intensity value for all cells (left), while the single-cell analysis (right) enables the extraction of spatially-resolved information from individual cell. The resulting population distribution has the same mean value as extracted from bulk analysis but resolves two subpopulations. Figure is taken from [5].

Microscopes Live-cell imaging is a powerful technique that offers dynamic insights into cellular processes by enabling real-time observation of living cells. Within microfluidic LoC devices, live-cell imaging is facilitated by incorporating microscopy capabilities onto the chip.

Microscopes play an essential role in microfluidic single-cell cultivation, enabling researchers to visualize and analyze individual cells in real-time. Microscopes' high magnification and resolution allow researchers to observe individual cells and track their movements, interactions, and reactions. Additionally, advanced microscopy techniques, such as fluorescence microscopy, can be used to monitor specific molecular interactions and cellular processes.

To conduct an experiment, the sample or device must be placed on the motorized stage under the microscope. The researcher can then program the microscope to move the stage to a specific location, adjust the focus, and capture images or videos of the sample at different magnifications and resolutions. In this context, a microscope is a collection of single hardware components (listed below) that must be orchestrated meaningfully to perform a desired task. A modern, high-end research microscope that is used with LoCs is highly automated and includes various parts (Figure 1.4):

- **Objective revolver:** Contains various selectable objectives, contrast techniques, or magnifications.
- **Condenser wheel:** Different objectives require different condensers.
- **Filter wheels:** Contain dichroic mirrors for different fluorescence emissions.
- **Switchable optical paths:** Allow selection between multiple cameras or eyepieces.
- **Stage in Z direction:** Enables focusing.
- **Stage in X/Y direction:** Moves the sample across the field of view.

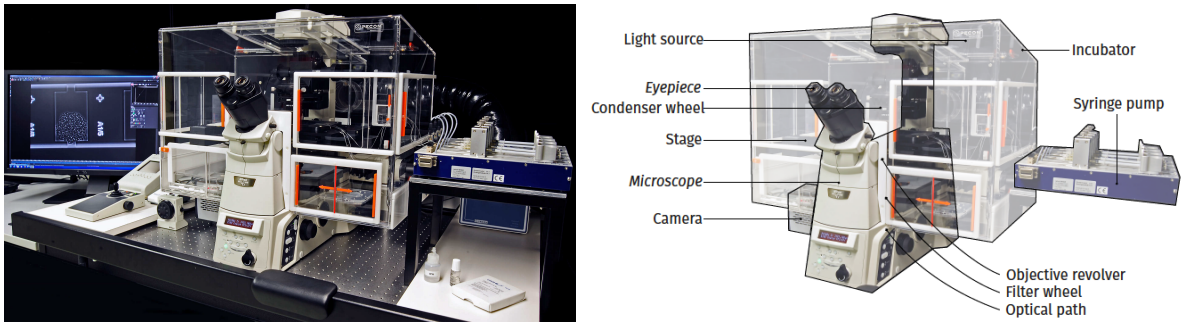


Figure 1.4.: A photo of a research microscope setup (left): with additional components for live cell imaging, such as an incubator and syringe pump system. The individual parts are shown in the right image. The figures are taken from [15].

- **Focus drift compensation:** Automatically maintains the correct focus during imaging sessions (e.g., Nikon Perfect Focus System (PFS), Leica Adaptive Focus Control, Zeiss Definite Focus, Olympus Z drift compensator).
- **Shutter:** Switches illumination on or off depending on the light source, without turning the source off entirely.
- **Light source:** Provides illumination.
- **Fluorescence light source:** Provides excitation light; typically includes a shutter and neutral density (ND) filters.
- **Cameras:** Connected to the microscope to record images [15].

All these components should be carefully controlled to maximize the microscope's capabilities. Microscopy software orchestrates acquiring various types of images, such as brightfield, fluorescence, phase-contrast, and confocal images. These images are often captured in multiple dimensions—spatial, temporal, and spectral and will be described in more detail in Chapter 2.

The data collected by the microscope can be analyzed and processed using image processing software or other analysis tools. This enables researchers to extract relevant information from the images or videos, such as cell count, size, and behavior.

In this thesis, we propose live-cell image analysis tools for analyzing data collected from microfluidic single-cell cultivation experiments. These tools can be applied to microfluidic single-cell datasets to extract valuable information regarding cell behavior and interactions, and they are discussed in the following chapters.

1.2.2. Live cell imaging in flow cell chambers

An additional approach for observing individual cells involves cultivating them in a microbioreactor and subsequently transferring them to a flow chamber under a microscope using a robotic platform. This method offers the advantage of easy scalability and the

ability to mimic industrial processes. Moreover, it enables the cultivation of a larger cell population while preserving their viability and allows for experimentation with a diverse range of growth conditions, as explained in detail below.

In our institute's experiments, we use the BioLector, equipped with a 48-well FlowerPlate® system, which allows the simultaneous online measurement of several process parameters, and the plate reader is used to monitor the growth of the microorganisms in the Biolector during the cultivation process. In the experimental pipeline, cells were cultivated in the Biolector system and then harvested from each well of the FlowerPlate®. The harvested samples were then stored in a deep well plate at 4°C on the robotic deck until automated microscopy was performed. This ensures the preservation and stability of the samples until the analysis is performed.

A robotic manipulator, a part of the Tecan robotic platform, was used to handle the samples during the automated microscopy workflow. It was responsible for transferring the samples from the deep well plate to the injection station and later to the flow chamber for imaging. For image acquisition, a self-built injection station was used, which was connected via tubing to a flow chamber (Figure 1.5).

The flow chamber was mounted on a microscope equipped with an oil objective. The microscope is placed next to the liquid handling platform. The sample, followed by water, was injected into the injection station. After the injection of the majority of the suspension, the flow was set to zero to allow image acquisition. A pulse with a 4-minute delay was performed several times to flush new cells into the flow chamber.

The images were taken of each sample in one channel of the flow chamber using a Thorlabs camera. This process enables efficient and automated screening and analysis of single-cell analysis and their dynamical behavior under realistic bioprocess conditions. Combined with an automated image analysis concept, this enabled an automated morphology analysis pipeline [12, 20].

1.3. Biotechnological platform organisms

Many industrially relevant microbes are rod-shaped, and their splitting behavior—whether symmetric or asymmetric—can vary, impacting their growth and productivity. By analyzing these behaviors in detail, we can better understand how to optimize them for industrial applications. In the following subsections, we focus on rod-shaped microorganisms that are highly prevalent in biotechnology, *Escherichia coli* (*E. coli*) and *Corynebacterium glutamicum* (*C. glutamicum*), starting with their biotechnological relevance and then exploring their morphology and behavior.

1.3.1. *C. glutamicum*

Biotechnological relevance

C. glutamicum is a gram-positive, non-pathogenic soil bacterium that is one of the major platform organisms for biotechnological processes [22, 23, 24], particularly in the following applications:

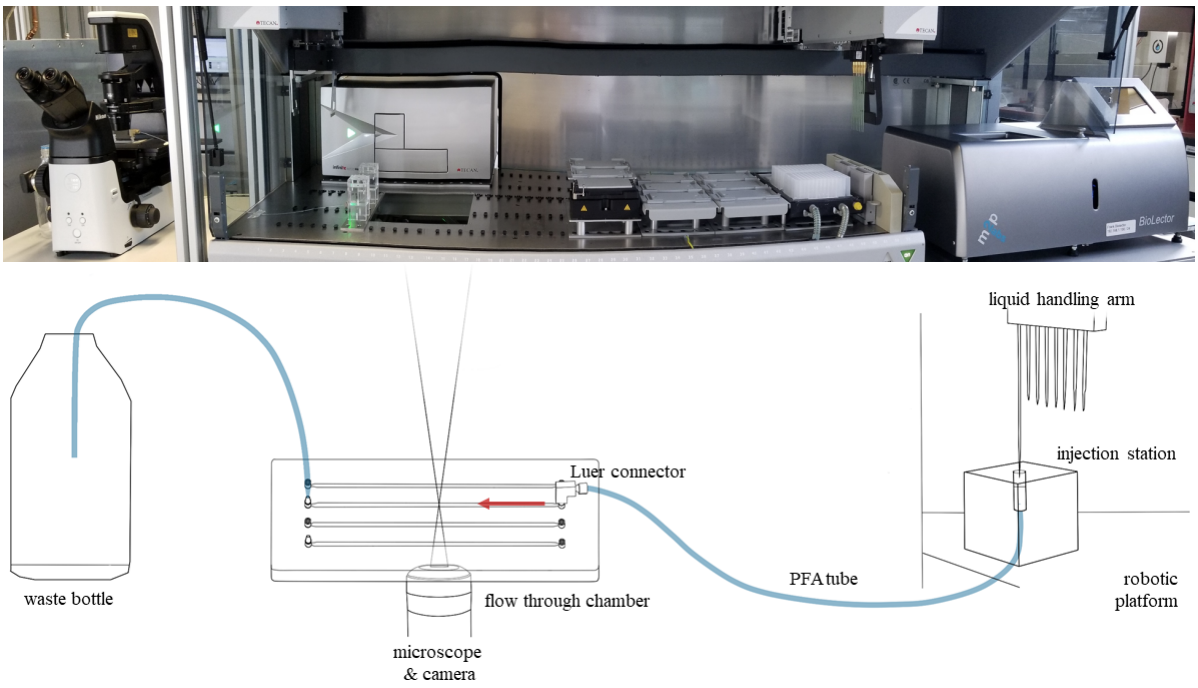


Figure 1.5.: Schematic representation of the robotic setup for automated microscopy (top) and liquid handling system (bottom). The figures are taken from [21].

- Production of amino acids: *C. glutamicum* is utilized for amino acid synthesis, especially L-glutamate, a key flavor enhancer used in the food sector. It can also synthesize other valuable amino acids like L-valine and L-lysine, which are commonly used in food and feed supplements [25].
- High efficiency: *C. glutamicum* demonstrates outstanding efficiency and output, making it a cost-effective option for major industrial operations. Its competitiveness in the biotechnological market is enhanced by efficiently utilizing a range of carbon sources such as sugars and organic acids [26, 27].
- Non-pathogenic: *C. glutamicum* is non-pathogenic, unlike other Corynabacteria species, meaning it does not cause diseases in humans or animals. This characteristic makes it safe to use in industrial processes and ensures the products derived from it are safe [28].
- Genetic manipulation: Over the years, extensive research has led to a deep understanding of *C. glutamicum*'s genetics and metabolism. This allowed genetic manipulation and metabolic engineering to enhance productivity and tailor its capabilities for specific applications [29].
- Bioprocess optimization: Due to its robustness and stability, *C. glutamicum* can thrive in various bioprocess conditions, including high temperatures and pH ranges. This adaptability simplifies the optimization of bioprocess parameters for efficient fermentation and product formation [30].

- Versatility: Besides amino acids, *C. glutamicum* can be engineered to produce various other valuable compounds, such as nucleotides, vitamins, and bio-based chemicals. Its flexibility in biosynthetic pathways makes it an attractive candidate for sustainable production strategies [31].

Morphology and behavior

C. glutamicum is a non-motile, rod-shaped bacterium that typically measures 1-5 µm in length and 0.5-0.8 µm in width. It forms single cells, pairs, or short chains and sometimes exhibits branching or irregular shapes. Snapping division is a unique mode of cell division observed in Corynebacteria bacteria, including *C. glutamicum*. In this division mode, the cell appears to snap or break in the middle, resulting in two unequal daughter cells.

The snapping division of *C. glutamicum* is characterized by forming a constrictive ring, which appears at a specific site on the cell surface, usually at one-third or one-fourth of the cell length. This ring contracts and tightens until the cell breaks in the middle, resulting in two daughter cells of unequal sizes [32].

The snapping division of *C. glutamicum* has been proposed to play a role in adaptation to changing environments. By producing unequal daughter cells, *C. glutamicum* can generate a diverse population of cells with different sizes and phenotypes, which may have advantages in certain environments [33]. However, this consequence of the growth process, i.e., snapping division in the biology of *C. glutamicum*, is still under investigation [34].

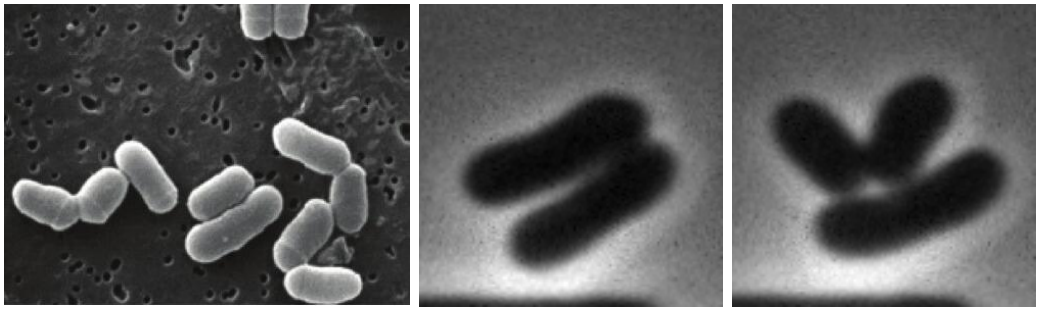


Figure 1.6.: Left to right: Scanning electron micrograph of *C. glutamicum* (The figure is taken from [23]), and two consecutive phase-contrast images, that illustrate *C. glutamicum* division event.

1.3.2. *E. coli*

Biotechnological relevance

E. coli is another significant bacterium with diverse biotechnology and industrial microbiology applications. Here are some key reasons for its importance:

- Protein expression and Biopharmaceuticals: *E. coli* is widely used for recombinant protein expression, making it an essential organism in biopharmaceutical production.

Its ability to efficiently produce proteins at high yields simplifies the production of antibiotics, vaccines, and enzymes [35].

- Molecular biology and Genetics: *E. coli* has been extensively studied, and its genetics and molecular biology are well understood. This knowledge allows researchers to manipulate its genome easily, enabling the development of genetically modified strains tailored for specific tasks. *E. coli* is a widely used model organism for gene expression studies and basic research in molecular biology. Its simple genetics and well-characterized regulatory mechanisms provide valuable insights into cellular processes [36].

- Vaccine Development: *E. coli*-based vaccines, particularly in veterinary medicine, have been developed. *E. coli* can be used as a carrier for antigens to stimulate an immune response [37].

- Metabolic engineering: *E. coli*'s metabolism is highly adaptable, and through metabolic engineering, it can be tailored to produce various compounds of industrial interest, such as biofuels, organic acids, and bioplastics [38].

- High Growth Rate: *E. coli* possesses a rapid growth rate, accelerating the fermentation process and reducing production time and cost. Its fast replication cycle is advantageous for large-scale industrial applications [39].

- Cost-effectiveness: *E. coli*'s cost-effectiveness as a microbial platform is another crucial factor in its importance. Its low nutrient requirements and straightforward cultivation make it a cost-efficient choice for various biotechnological applications [40].

- Recycling: In environmental biotechnology, certain strains of *E. coli* are employed for nutrient recycling and wastewater treatment, contributing to sustainable solutions for waste management [41].

Morphology and behavior

E. coli cells are typically rod-shaped, with a length of 2-6 µm and a width of 0.5-1 µm. The cells are often slightly curved, giving them a slightly helical appearance. The cell's elongated shape allows it to occupy and adapt to its environment efficiently. The cell wall of *E. coli* comprises peptidoglycan, providing structural integrity and protection.

E. coli reproduces through binary fission, a process where one parental cell divides into two daughter cells. This division behavior is highly coordinated and regulated to ensure the faithful transmission of genetic material to the next generation.

Before division, *E. coli*, as the other microorganisms, must duplicate its genetic material. As the cell prepares for division, it undergoes a period of growth and synthesis of cellular components, including proteins, ribosomes, and other organelles. During division, a septum (cell wall partition) forms at the midsection of the cell. The septum is composed of new cell wall material synthesized by enzymes. The septum grows inward until it

pinches the cell into two daughter cells, each containing a copy of the genetic material. The final division stage is the so-called cytokinesis, where the two daughter cells physically separate from each other. The process results in two genetically identical daughter cells.

Unlike *C. glutamicum*, *E. coli* does not "snap" or form V-shaped bulges during cell division.

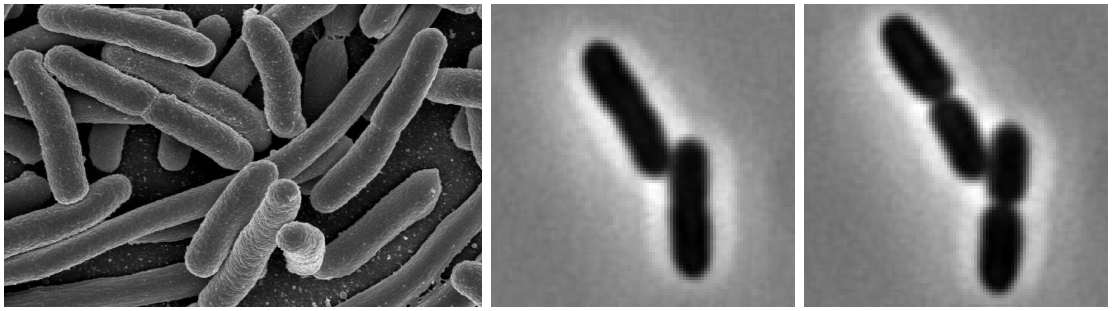


Figure 1.7.: Left to right: Scanning electron micrograph of *E. coli* (The figure is taken from [42]), and two consecutive phase-contrast images that illustrate *E. coli* division events.

2. Image analysis of microbial single cells

Image processing is an essential step in single-cell analysis, which involves the extraction of quantitative information from microscopic images of individual cells. In the described above automated microscopy processes, images of the samples are acquired using a microscope and a camera.

Phase-contrast microscopy is one of the most widely used imaging techniques in live-cell imaging [43]. It is based on the conversion of phase shifts of light passing through a specimen to brightness changes in the resulting image. Especially when working with unstained biological specimens, the contrast between a cell and the surrounding is low when using bright field illumination [44]. Moreover, alongside phase-contrast imaging, fluorescence microscopy serves as another invaluable tool in biological research. By utilizing a fluorescence excitation light source, fluorescence microscopy enables the visualization of specific molecules or structures within cells, providing crucial insights into various cellular processes [16].

In this thesis, our primary focus is on the processing of phase-contrast images. While fluorescence microscopy undoubtedly holds significance in cellular imaging, it's important to acknowledge that not every experiment involves fluorescence. Therefore, our attention is dedicated to the methodologies and algorithms employed in analyzing phase-contrast images.

The resulting phase-contrast image is recorded as a 2-D array of values, where each value represents a pixel in the acquired image (Figure 2.1).

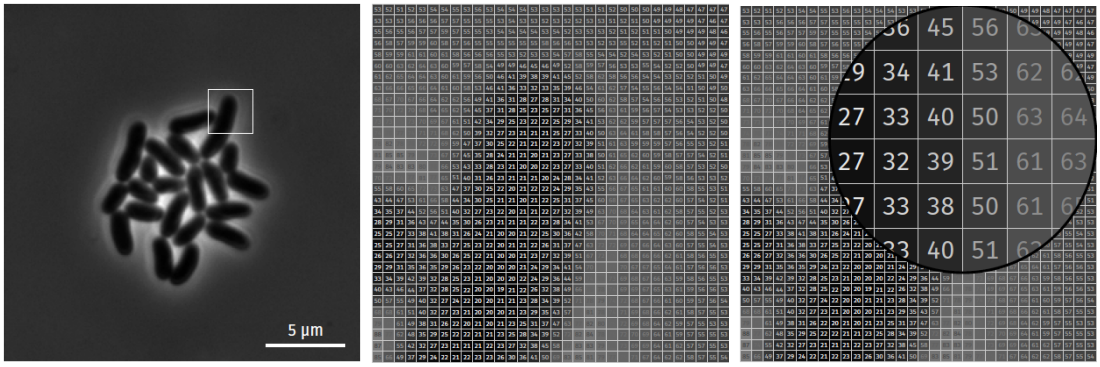


Figure 2.1.: Left to right: Image of *C. glutamicum* cells, 32 × 32 2-D array of the intensity values of the image crop, zoomed-in intensity array. The figures are taken from [15].

These arrays can then be further analyzed using various image analysis techniques to extract more information about the samples described below.

2.1. Preprocessing

Image preprocessing is a critical step in microfluidic live-cell experiments to enhance the quality of the images obtained and improve downstream analyses' accuracy. Some common image preprocessing steps in microfluidic live-cell experiments include:

- Image registration: This step involves aligning images acquired at different time points or from different channels (i.e., phase contrast and fluorescence) to correct for the shifts or movements of the microscope when switching between chambers (Figure 1.2) [45]. Image registration is important for accurate tracking of cells over time.
- Background subtraction: Microfluidic live-cell experiments may suffer from uneven illumination or background noise, which can interfere with image analysis. Background subtraction methods are used to remove the background signal from the image, enhancing cell visibility and improving the accuracy of subsequent steps of the image analysis [46].
- Denoising: Images may contain unwanted noise, which can distort the signal and reduce the accuracy of downstream analysis. Denoising techniques, such as median filtering or wavelet denoising, can be used to remove noise from the images [47].
- Contrast enhancement: Contrast enhancement techniques, such as histogram equalization or gamma correction, can be used to enhance the contrast of the images and improve the visibility of features in the image [48].
- Image cropping: Image cropping in cell microscopy is essential for efficient memory utilization, restricting the analysis to specific regions of interest and reducing computational load [49].

In this dissertation, the preprocessing steps, such as image registration and cropping, are not covered and were performed by the experimental scientist using ImageJ [50].

2.2. Segmentation

Segmentation is a crucial step in single-cell image analysis that involves identifying and separating individual cells, each representing an instance, from an image or a series of images, where each cell constitutes a distinct segment in the context of image segmentation. This process is necessary to extract quantitative information about each cell, such as its morphology, size, intensity, location, and fluorescence.

In single-cell image analysis, there are two main types of segmentation: manual and automated. Manual segmentation involves a human expert manually drawing a contour

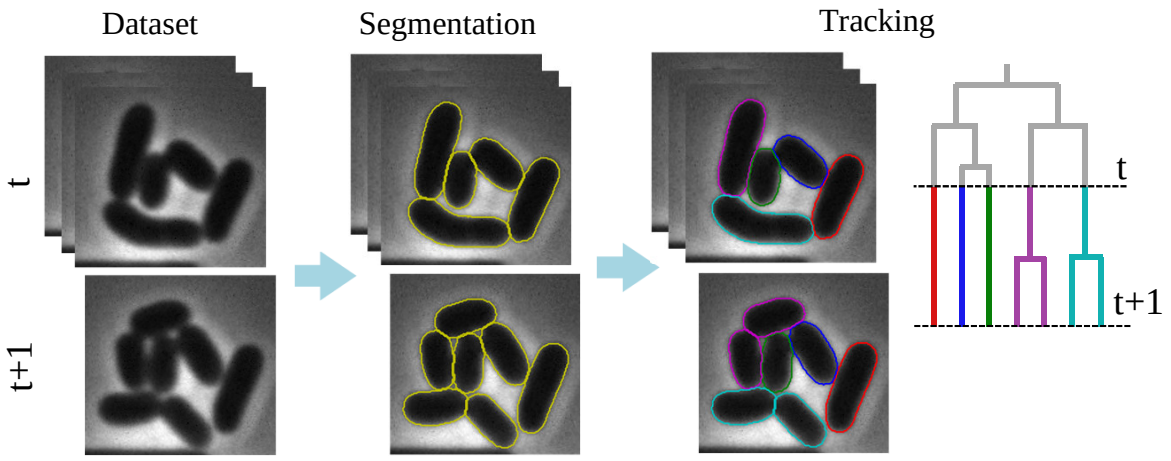


Figure 2.2.: Image processing pipeline for a single-cell analysis that includes instance segmentation and tracking with the lineage tree generation. The pipeline is shown for two consecutive frames (t and $t+1$) of the dataset.

around each cell in an image, which is time-consuming and subject to inter-observer variability (Part II). Automated segmentation, one of the main focuses of the dissertation, on the other hand, uses computer algorithms to identify and separate individual cells from an image or a series of images.

There are several algorithms used for automated segmentation in single-cell analysis, including thresholding, watershed segmentation, active contours/variational methods, and Machine Learning (ML)-based methods, which are described below.

Segmentation accuracy of the methods varies depending on the quality of the image, the algorithm used, and the level of complexity of the cell structures. To ensure accuracy, manual verification and correction of the segmentation results may be required.

2.2.1. Histogram-based methods. Thresholding and watershed

Thresholding is a simple and widely used segmentation method that involves setting a threshold value for the intensity to separate the cells (foreground) from the background based on the pixel intensity. In this method, the pixel intensities above the threshold value are assigned to the cells (foreground), and those below the threshold are assigned to the background. Essentially, it assumes a clear separation between these segments based on pixel intensity values. In the context of cell segmentation, thresholding results in a single segment containing all the cells. However, it's essential to note that thresholding is not an instance-based method; it doesn't distinguish individual cells but segments the entire image into two regions. Later steps might be necessary to refine this segmentation and delineate individual cells.

In digital image processing, thresholding is the simplest method for image segmentation. While in some cases, the threshold can be selected manually by the user, it can also be automatically set by an algorithm.

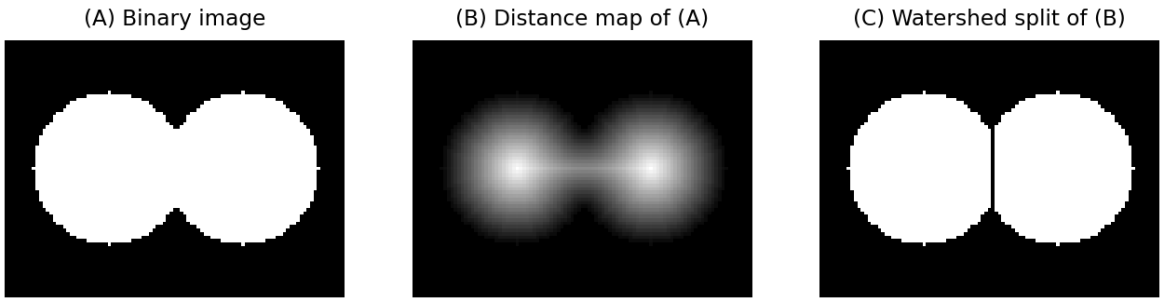


Figure 2.3.: The watershed segmentation of the binary image of two merged circles. The figure is taken from [55].

There are many types of automatic thresholding methods, with Otsu's method [51] being one of the most well-known and widely applied.

Typically, these methods use a uniform threshold for all the pixels in an image. However, in certain cases, it can be more effective to apply different thresholds to various parts of the image, depending on local pixel values, i.e., in the presence of an intensity gradient. This category of methods is called local or adaptive thresholding. They are particularly adapted to cases where images have inhomogeneous lighting. In those cases, a neighborhood is defined, and a threshold is computed for each pixel and its neighborhood, i.e., Niblack [52] or the Bernsen [53] algorithms. The local thresholding methods serve as an alternative to preliminary background subtraction, mentioned earlier, to address the challenges posed by varying background intensities.

Thresholding will work best under certain conditions: low noise level; higher intra-class variance than inter-class variance, i.e., the clear contrast between the cells and the background; homogeneous cell intensities, homogeneous lighting; sparsely distributed (untouching) cells.

In cases where cells are merged (about to divide) or touching in an image, standard thresholding techniques alone may not be sufficient to separate individual cells accurately. After applying thresholding to generate binary masks, the watershed algorithm [54] can be employed to segment and split merged regions within these masks accurately. Watershed algorithm, also known as Watershed segmentation, is a more sophisticated image processing method compared to simple thresholding methods that can effectively address this challenge by separating different objects (instances) in an image. The name watershed refers metaphorically to a geological watershed, or drainage divide, which separates adjacent drainage basins. The watershed transformation treats the image it operates upon like a topographic map, with the brightness of each point representing its height, and finds the lines that run along the tops of ridges (Figure 2.3).

To implement watershed segmentation, a distance map (Figure 2.3 B) is calculated from the binary mask (Figure 2.3 A). The distance map assigns each pixel in the binary image a value corresponding to the distance to the nearest cell boundary. This distance information helps to determine the locations of potential cell centers. Finally, the watershed algorithm is applied to the distance map [55]. This allows for counting

the objects or further analyzing the separated objects. This is a simple way to split a binary segmentation of cell images into individual cells. Still, it can result in unrealistic cell boundaries and merged or unnecessary split segmentation masks and often require techniques like Gaussian filtering or morphological operations, which might struggle with the complexities of cell morphology in microfluidic devices. Additionally, expert editing may be necessary in some cases to refine the segmentation, therefore more complex methods are crucial to address these challenges effectively.

2.2.2. Variational methods

Variational segmentation methods are a class of more complex algorithms used for segmentation. Variational methods, in general, aim to find a solution that optimizes a specific objective function. This objective function is constructed to quantify how well certain conditions or criteria are met. Consider the general optimization problem:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) \tag{2.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function, and $x \in \mathbb{R}^n$ is the variable to be optimized.

The solution to this optimization problem is given by:

$$x^* \in \arg\min_x f(x) \tag{2.2}$$

where x^* is a value of x that minimizes the function f.

In the context of variational segmentation, the goal is to find the segmentation that minimizes a cost function composed of two main terms: a data term and a regularization term. The data term measures how well the segmentation fits the image data, while the regularization term enforces smoothness and continuity of the segmentation [56, 57].

One popular variational segmentation method used in single-cell analysis is the Chan-Vese model [58]. This method is based on a level set representation of the segmentation, where the binary mask is described in terms of a level set function ϕ so that the super level-set $\{\phi > 0\}$ describes the foreground and $\{\phi \leq 0\}$ describes the background. It seeks to minimize a cost function that balances the difference in intensity values between the object and the background and the smoothness of the object boundary. The Chan-Vese functional is defined as follows:

$$E(\phi, c_1, c_2) = \mu\cdot\text{Length}(\phi)+\nu\cdot\text{Area}(\phi)+\int_{\{\phi>0\}} |I(x)-c_1|^2\,\mathrm{d}x+\int_{\{\phi\leq 0\}} |I(x)-c_2|^2\,\mathrm{d}x \tag{2.3}$$

where ϕ is the level set function representing the evolving boundary between regions, c_1 and c_2 are the average intensities inside the segmented regions, μ, ν are user-defined parameters, $\text{Length}(\phi)$ measures the length of the evolving curve, $\text{Area}(\phi)$ calculates the area enclosed by the curve, $I(\mathbf{x})$ is the intensity function of the image at position $\mathbf{x}$.

One important aspect of the Chan-Vese model is expressing these terms using the Heaviside function H, which is defined as:

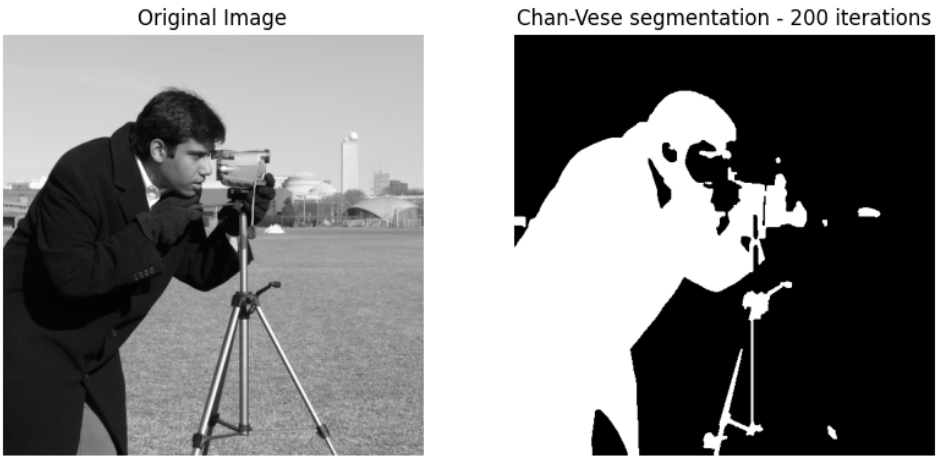


Figure 2.4.: The example of Chan-Vese segmentation of the cameraman image. The figure is taken from [59].

$$H(\phi) = \begin{cases} 0 & \text{if } \phi < 0, \\ 1 & \text{if } \phi \geq 0. \end{cases} \tag{2.4}$$

$$\begin{aligned} E(\phi, c_1, c_2) =& \mu \cdot \int_\Omega |\nabla H(\phi)| d\mathbf{x} + \nu \cdot \int_\Omega H(\phi) d\mathbf{x} \\ &+ \int_\Omega |I(\mathbf{x}) - c_1|^2 H(\phi) d\mathbf{x} + \int_\Omega |I(\mathbf{x}) - c_2|^2 (1 - H(\phi)) d\mathbf{x} \end{aligned} \tag{2.5}$$

where Ω represents the image domain. This representation using the Heaviside function replaces the need for sub-level and super-level sets, providing a unified mathematical formulation. The goal is to find the optimal values of ϕ, c_1, and c_2 that minimize the energy functional. This is expressed as:

$$(\phi^*, c_1^*, c_2^*) \in \arg\min_{\phi, c_1, c_2} E(\phi, c_1, c_2) \tag{2.6}$$

The solution (ϕ^*, c_1^*, c_2^*) represents the optimal level set function and region average intensities. It provides a segmentation that accurately captures object boundaries while maintaining smooth regions. This balance is achieved through the objective function, which comprises both a data term and a regularizer. The parameters associated with these terms allow users to fine-tune the compromise between data fidelity and the segmentation's smoothness.

Variational segmentation methods are valuable because they allow researchers to incorporate the geometric properties of the object into the loss function. These methods aim to optimize a functional that measures the agreement between the segmentation result and the true object boundaries (by the data term) while also considering the

smoothness and geometric properties of the segmented regions (by the regularization term(s)). For example, a higher μ value in the regularization term emphasizes shorter, smoother contours, reducing noise sensitivity and avoiding over-segmentation. Conversely, a lower μ value allows the contour to adapt more closely to the image data, potentially capturing more intricate details of the object boundaries but at the risk of increased noise sensitivity and over-segmentation.

By incorporating geometric properties, such as shape priors or curvature constraints, variational methods can lead to more accurate and natural-looking segmentation results compared to the methods discussed in Section 2.2.1. They can provide better regularization of the segmentation process, reducing artifacts and producing more coherent object boundaries.

However, variational methods may face challenges when dealing with images containing a large number of objects present simultaneously. As the number of objects increases, the optimization process's computational complexity grows significantly with the growing number of variables and the terms in the loss function. This can make the segmentation problem computationally expensive and time-consuming, especially for real-time or large-scale applications.

Moreover, in cases where objects overlap or are close together, the optimization process can struggle to distinguish and accurately segment individual objects, leading to segmentation errors or inaccuracies.

To address these challenges, researchers often combine variational methods with other techniques, such as watershed segmentation or ML-based approaches. Such a combination can provide more robust and efficient solutions for segmenting complex images with multiple objects. Such a hybrid approach will be introduced in Chapter 5.

2.2.3. Supervised ML-based methods

Supervised machine learning neural network-based methods have become prominent in image segmentation tasks in recent years. An artificial neural network is a biologically inspired computational model formed from hundreds of single units (artificial neurons) connected with coefficients (weights) that constitute the neural structure [60]. They offer a data-driven approach, meaning that the GT data has to be provided such that a neural network can automatically learn to identify and segment objects from images. This approach eliminates the need for explicitly providing a model, unlike in variational methods, and instead learns from the examples from the provided data.

Supervised machine learning involves learning a mapping $F : X \to Y$ from example pairs $(x_i, y_i)_{i=1}^n$, where $y_i = F(x_i)$. In this process, we aim to approximate this mapping using a parametrized function $\hat{F}[\phi] : X \to Y$, which is often represented by a neural network, where ϕ denotes the network parameters (e.g., weights). The goal is to find the optimal parameters ϕ such that $\hat{F}[\phi](x_i)$ approximates y_i for all i. This optimization process involves adjusting the parameters ϕ iteratively using techniques such as gradient descent to minimize the difference between the predicted outputs of the network and the actual labels (ground truth) in the training data. The neural network learns to generalize from the examples and predict outputs for new, unseen inputs through this iterative

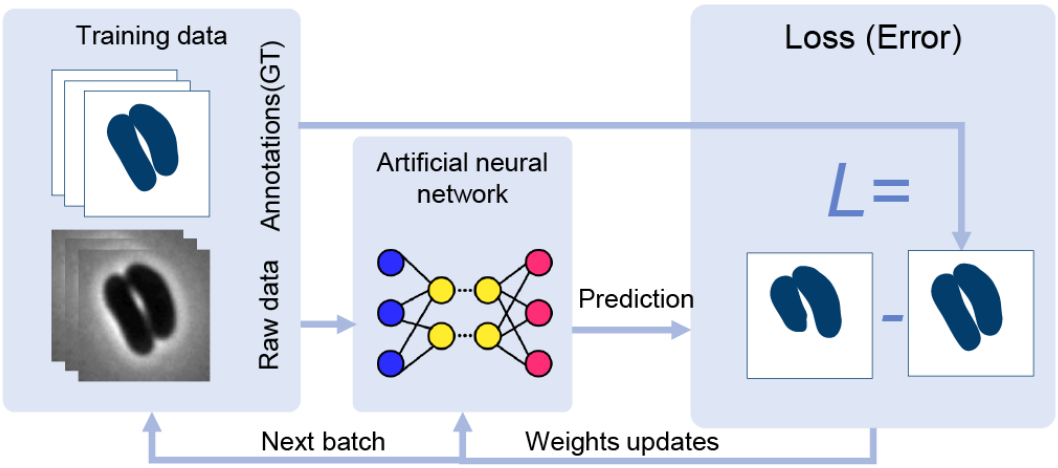


Figure 2.5.: The schematic representation of training pipeline for a neural network-based segmentation.

optimization [61].

The basic concept of supervised learning involves several key steps (Figure 2.5):

(i) Forward propagation: During forward propagation, the input image x is fed into the neural network, where $\hat{F}[\phi]$ represents the neural network with parameters (weights) ϕ. The network processes the image through multiple layers of neurons. Each neuron performs a weighted sum of its inputs, applies an activation function σ, and passes the output to the next layer. This process continues through the network until the final output layer is reached.

(ii) Loss calculation: The neural network's output $\hat{y} = \hat{F}[\phi](x)$ is compared to the ground-truth segmentation y of the image to calculate a loss or error value $L(\hat{y}, y)$. The loss function L quantifies the dissimilarity between the predicted and actual segmentation, measuring how well the network performs.

(iii) Loss minimization: The goal of training the neural network is to minimize the calculated loss L. This is achieved through an optimization algorithm, such as gradient descent. The optimization algorithm adjusts the weights of the network's neurons to reduce the error and improve the accuracy of the segmentation.

(iv) Updating weights: During the training process, the network parameters ϕ are updated iteratively based on the gradient of the loss function with respect to the network parameters, denoted as $\nabla_\phi L$. The optimization algorithm determines the direction and magnitude of the weight updates, gradually guiding the network toward better segmentation performance.

(v) Iterative training: The training process involves feeding multiple images (batch) into the neural network and repeatedly updating the weights to minimize the loss

across the entire dataset. This iterative training allows the network to learn from the data and improve its segmentation ability over time.

These data-driven approaches have advantages for cell segmentation, but they come with challenges. Producing the training data, which requires manual annotation of cells in images, can be laborious and time-consuming. Quality and diversity of the training data are crucial for the model's performance [62, 63].

2.3. Tracking

Tracking refers to the process of following and monitoring the movement and behavior of objects or entities over time in a sequence of images or videos. It involves detecting objects in each frame of the sequence and connecting these detections over time to create a continuous trajectory for each object. In the context of this dissertation, the detection step is accomplished through multi-object (instance) segmentation, as described above.

Tracking can be classified into several different methods. Some of the most common ones are:

- Feature-based tracking: This method involves identifying and tracking the movement of objects based on the distinctive features of the objects in images, such as shape, texture, or intensity. Feature-based tracking requires a manual selection of features and their extraction. Finding the correspondence between objects' feature vectors is crucial for accurately tracking objects across frames. One common approach to achieving this is by minimizing the distance between feature vectors in different frames. The distance metric is used to measure the similarity or dissimilarity between feature vectors, and the goal is to find the best-matching feature pairs with the smallest distance.

  By focusing on distinctive features, feature-based tracking can be computationally efficient and robust against noise and variations in lighting conditions but may struggle when dealing with repetitive patterns or objects with limited or ambiguous features [64, 65].

- Correlation-based tracking: This method involves using cross-correlation between successive frames in a video to determine the displacement of objects. In the first frame, each object is detected. The detected regions serve as templates for subsequent tracking. In the following frames, a search region around the previous location of each object is defined. The size of these search regions is determined based on the expected movement of the objects between frames.

  Within each search region, the cross-correlation between the corresponding template and the pixels inside the window is computed at each position. This cross-correlation measures the similarity between the template and the region being examined. The position with the highest correlation value indicates the best match between the template and the corresponding region in the current frame, determining the new location of each object.

Correlation-based tracking is fast and efficient, but it can be sensitive to changes in lighting, object shape, or orientation [66].

- Optical flow-based tracking: Optical flow-based tracking involves analyzing the movement of image pixels between successive frames in a video to determine the movement of objects.

  The process begins by computing the motion vectors for each pixel or small image patch in the current frame relative to the previous frame. These motion vectors represent the displacement of each pixel or patch over time and provide valuable information about the object's movement.

  The objective function for optical flow estimation aims to minimize the difference between the observed intensities of pixels in successive frames and their predicted locations based on the estimated motion vectors. One common formulation is the sum of squared differences (SSD) or the sum of absolute differences (SAD) between pixel intensities.

  For example, let $I(x, y, t)$ represent the intensity of a pixel at position (x, y) in frame t. The goal is to find the displacement vector (u, v) for each pixel such that the objective function

  $$E(u, v) = \sum_{x,y} (I(x, y, t) - I(x + u, y + v, t + 1))^2 \tag{2.7}$$

  is minimized by finding the displacement vectors u and v that result in the smallest possible difference between the intensities of corresponding pixels in successive frames.

  Optical flow provides a dense representation of motion, enabling precise tracking of pixel-level movement across frames. By integrating object detection or segmentation with optical flow-based tracking, the the motion information obtained from optical flow can be associated with specific objects detected or segmented in the scene.

  Despite its effectiveness, optical flow-based tracking can be sensitive to scene changes, occlusions, and large displacements, which may lead to ambiguous motion vectors and potential tracking errors [67, 68].

- Estimation methods formulate the tracking problem as an estimation problem in which a state vector represents each object. The state vector describes a system's dynamic behavior, such as an object's position and velocity. Estimation-based object tracking often uses filtering techniques to predict and update the state vector at each time step. Common filtering methods include the Kalman filter[69] and Particle filters [70].

  The Kalman filter is a recursive estimator that predicts an object's next state based on its current state and a motion model and then updates this prediction using actual measurements. It effectively balances the predicted and measured

states, considering the uncertainties associated with both the motion model and the measurements.

On the other hand, particle filters use a probabilistic representation of the object's state through a set of sampled particles. These particles represent various hypotheses about the object's state and are propagated through time based on the motion model. Particle filters are particularly useful for handling non-linearities and non-Gaussian noise in the tracking process.

These filters are designed to handle uncertainties and noise in the measurements and provide a mechanism to fuse the information from the measurement and motion models to produce an improved state vector estimate. However, applying these filters in the initial frames presents challenges. Accurately initializing the state estimation process can be difficult without sufficient information from previous frames.

By incorporating motion models and filtering techniques, estimation-based tracking can handle noisy measurements and uncertainties in the data, leading to more robust tracking results, but, over time, tracking errors and inaccuracies in the estimation process can accumulate, leading to drift in the object's tracked position and potential loss of tracking [71].

- Deep learning-based tracking: This method involves using convolutional neural networks (CNNs) to learn objects' appearance and movement patterns from a set of labeled images or videos [72, 73].

  Once trained, the CNN can process each frame in a video sequence to detect and track objects. The network can handle changes in lighting, occlusions, and other challenges that traditional methods might struggle with due to its ability to learn complex features and patterns from the data.

  However, formulating the object-tracking problem and choosing the appropriate network structure can be challenging. It requires defining the exact objectives, constraints, and annotation data format. Additionally, a significant amount of labeled data is necessary for effective training, which can be time-consuming and resource-intensive to obtain.

  Deep learning-based tracking offers high accuracy and adaptability but demands substantial computational resources and expertise in designing and training deep learning models.

- Hybrid tracking: This method combines two or more tracking methods to improve the accuracy and robustness of the tracking [74, 75].

2.4. Cell tracking in live-cell imaging

In our research, we are particularly interested in cell tracking. As mentioned in the introduction, live-cell (time-lapse) microscopy combined with microfluidic lab-on-chip

technology enables observing cells over time and, therefore, tracking [3]. Unlike standard tracking tasks (i.e., high-frame rate people or car tracking), tracking living microorganisms has specific, biology-related challenges. The main difference between cell tracking (Figure 2.6) and regular object tracking (Figure 2.7) tasks lies in the fact that cells can undergo division, resulting in the need to determine mother-daughter relations. In cell tracking, it is essential to monitor the movement of individual cells and establish the lineage of cells that appear from division events. One positive aspect is that cell tracking often does not involve occlusions, unlike in typical tracking scenarios, simplifying the tracking process.

Tracking cell division events and determining mother-daughter relationships require specialized algorithms that can handle the complexity of cell dynamics and ensure accurate tracking throughout the cell's life cycle. This involves identifying the division events, associating the newly formed daughter cells with their respective mother cells, and tracking their movements [76, 77] and will be discussed in detail in Chapter 7.

Generally, at the beginning of a live-cell image sequence, only a few cells are present, and those are sparsely distributed. In this situation, tracking, i.e., linking cells over frames and detecting division events, is relatively straightforward. As the experiment progresses, accurately detecting associations becomes increasingly difficult, especially during exponential growth phases. In densely populated cell colonies, where numerous divisions occur and result in significant and unpredictable cell movements between consecutive frames, correctly identifying cell associations becomes challenging even for experts [78]. Compared to standard object tracking tasks with a high frame rate, where the object positions can be predicted quite accurately because of relatively small frame-to-frame object displacements [79], in microbial live-cell imaging, the researchers have to deal with a low frame rate, relative to the cell division rates.

The low frame rate, that is used in such time series generally cannot be increased since illumination from live-cell imaging, when coupled with fluorescence, leads to phototoxic effects that may change the cells' growth behavior, thereby distorting the interpretation of the results. Consequently, the inevitable low frame rates, together with the cell movements, make the division events hard to predict and classify by time, leaving us dealing with many daughter cells that need to be linked to their mothers.

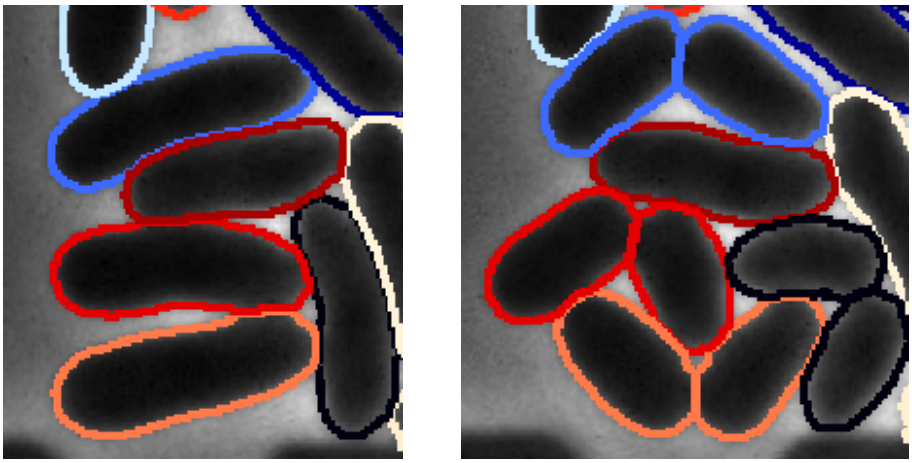


Figure 2.6.: The example of the cell tracking task illustrating multiple cell division events in two consecutive frames. The corresponding links are visualized with the outline colors.

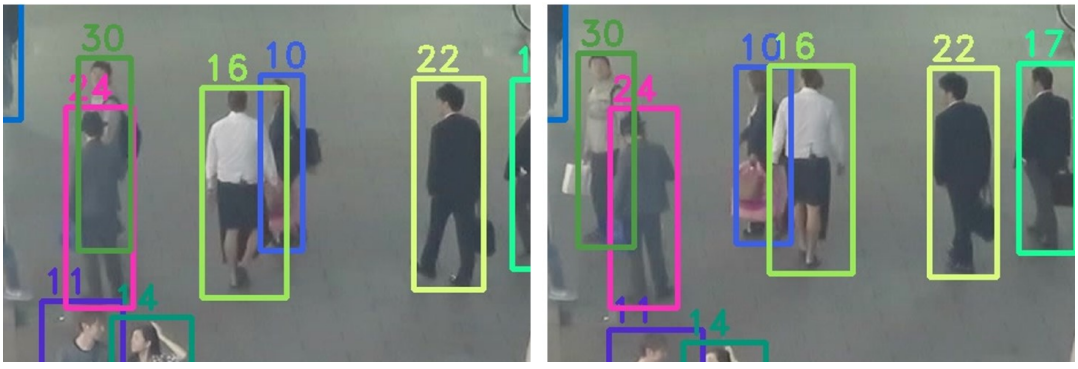


Figure 2.7.: Multi-person tracking in video surveillance. The figure is taken from [80].

Part II.

Ground truth generation

How bacteria maintain their shape and size is one of the big open questions of life [81]. As introduced in Section 1.2, microfluidic single-cell analysis coupled with live-cell imaging microscopy is a versatile tool to study the variety of cell morphologies and sizes together with their behavior [82, 13]. Rod-shaped bacteria or bacilli are found in many different taxonomic groups of bacteria. The group includes the highly industrially important bacteria, such as *C. glutamicum* and *Bacillus subtilis* with shapes ranging from spheric cocci to stretched, round-ended cylinders [83].

As mentioned in Chapter 2, multi-object segmentation is used to characterize every cell captured in microfluidic single-cell experiments within each image frame.

The prediction accuracy of supervised methods highly correlates with training data quality. Despite the availability of synthetic training data generation software, which produces natural-looking images with given bacteria type and microscopy settings, the presence of at least a fraction of real training data, nevertheless, improves the network's performance. Due to the uniqueness of each microfluidic experiment, the online available training and benchmark datasets may not match the target data. Besides, the real GT data is essential during the validation step to estimate the performance of the segmentation approach. Generating a ground truth (GT) for cell segmentation involves creating a reference that accurately identifies the location and boundaries of each cell of an image or image sequence. The GT can be used as a benchmark for evaluating the accuracy of automated segmentation methods and/or as training data for the data-driven approaches.

There are several methods for generating a GT for cell segmentation:

- **Manual annotation:** Manual annotation involves manually tracing the boundaries of each cell in the image using specialized software tools. Manual annotation by domain experts can be highly accurate but can be time-consuming and labor-intensive, especially for large datasets.

- **Semi-automatic annotation:** This involves a preliminary algorithmic approach to assist in annotation, such as ML-based algorithms or interactive segmentation tools and manual check and (or) correction of the algorithmic output, i.e., "silver" GT [84]. Semi-automatic annotation can be faster than manual annotation, but it may require effort to ensure accuracy.

- **Crowdsourcing:** This involves outsourcing the annotation task to a group of people, such as domain scientists or trained annotators who annotate images or image sequences independently. Crowdsourcing is a strategy to upscale the manual or semi-automatic annotation processes, which are mentioned above, making it more efficient to generate a GT for large datasets and/or a "gold" GT generation, obtained as a consensual or majority opinion of several human experts [84, 85].

- **Simulation of artificial images**, which involves generating synthetic images that mimic the characteristics of real cell colonies, along with their annotations, which can be used to train and validate computer vision algorithms for image segmentation, tracking, and analysis [86] and will be discussed in detail in Chapter 4.

For the first three methods mentioned above, the quality of the GT depends on the accuracy of the annotations, which can be influenced by several factors, including image quality, cell type, and the skill and effort of the annotator. Clear annotation guidelines and quality control measures, such as inter-annotator agreement checks, are essential to ensure annotation accuracy and consistency.

As mentioned above, to create a gold standard GT, multiple expert annotators independently annotate the same set of images or image sequences. This is used as a benchmark to evaluate the accuracy and reliability of other automated or semi-automated segmentation methods. The accuracy of these methods is typically measured by comparing their segmentation results to the gold standard using various performance metrics.

To produce gold standard GT, resulting annotations provided by the experts are then compared and reconciled to create a consensus annotation that represents the "truth" or "gold standard" for cell segmentation. Ensuring the accuracy and reliability of the gold standard requires a method to calculate an accurate average of the annotations, as there are multiple annotations that need to be combined into a consensus. This process will be discussed in detail in Chapter 3.

Creating a gold standard GT for cell segmentation requires significant resources and effort but is essential for validating and improving automated segmentation methods. It can also provide valuable insights into the variability and complexity of cell segmentation tasks and help to identify areas for further research and development.

Annotation variability

The variability in the annotation results has two major origins:

- **Technical:** The difference in the manually drawn cell outlines can be explained by different input devices (i.e., graphic tablet or mouse (Figure 2.8), which experts use to draw the cell contours. Indeed, although leading to more coarse segmentation results, using a mouse as an input tool for training data creation is the predominant approach, arguably due to being the cheaper and more abundant alternative. Another technical explanation of the deviating annotation results is the different brightness, contrast, and resolution of the output devices (i.e., monitors or tablet displays).

- **Methodological:** Since there are no commonly established protocols for cell border annotation, experts' opinions about the localization of cell borders may differ. In this case, the segmentation of one expert consistently tends to catch the inner rather than outer cells' border and vice versa.

- Moreover, the silver GT creation [84], where the GT is created algorithmically and manually refined by an expert, may add the third "**psychological**" item to the list of origins. Compared to annotating from scratch, the expert will more likely agree with the offered (automatic) segmentation to minimize the effort to refine the contour.

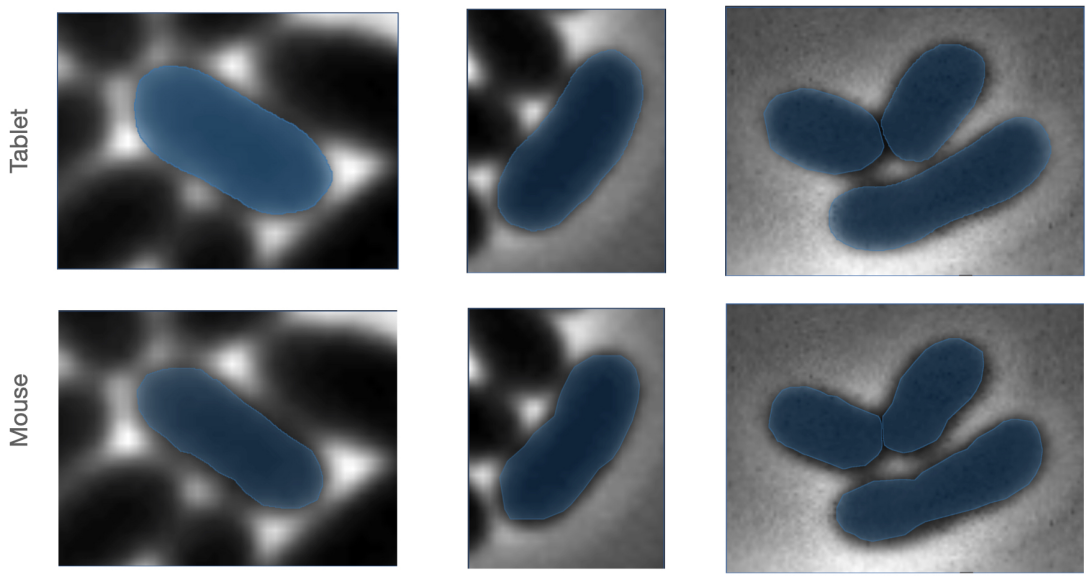


Figure 2.8.: Snapshots from the annotation software, illustrating the comparison of the annotations made by the experts using a graphical tablet (top) vs. a mouse (bottom).

3. Polar space-based shape averaging for star-shaped biological objects

This chapter is based upon the publication "Polar space-based shape averaging for star-shaped biological objects" by **K. Ruzaeva**, K. Nöh and B. Berkels. The author of this thesis developed the algorithm described therein, evaluated the data, and prepared the manuscript.

3.1. Traditional approaches to 2-D shape averaging

As mentioned in Part II, in order to generate the pixel-accurate GT, the images representing the target microorganisms must be labeled by the domain experts. However, this process is time-intensive and laborious [87]. Besides, despite their simplistic shape, in the prevalent case of low-resolution and low-signal-to-noise ratio data, image annotation (i.e., drawing cell outlines) becomes difficult, even for domain experts. In this case, repeated drawing of the desired segmentation by several raters (experts) and the "averaging" of the obtained contours is the only acceptable approach to come to a consensus GT.

Traditional approaches for 2-D shape averaging have several disadvantages. Simple landmarks methods [88, 89] are based on a distance averaging between corresponding landmarks (i.e., important geometrical features) and require manual laborious landmarks' specification or an automatic derivation of the landmarks as an additional preprocessing, which can be an additional source of errors, which can cause undesired results.

Alternatively, methods based on the Fourier approximation of a closed contour and averaging of the corresponding Fourier coefficients for a given number of harmonics [90] find their application in [91] and provide a smooth average outline of the average object, but highly depend on the number of the harmonics one approximates the contour with.

Another option to find a shape average is variational-based averaging methods [92] dealing with sophisticated non-aligned objects, and may even provide statistical information in addition to the average shape [93]. However, these methods are computationally expensive and require careful initialization to avoid local minima in the optimization due to the usually non-convex objective functionals involved. For this reason, the variational methods are more complicated than necessary in our case because we are dealing with relatively simple and aligned objects (the experts are labeling the same image, as illustrated in Figure 3.3).

A widely used algorithm, which is specifically aimed at a segmentation mask fusion given by different raters and has a user-friendly implementation [94], is the Simultaneous

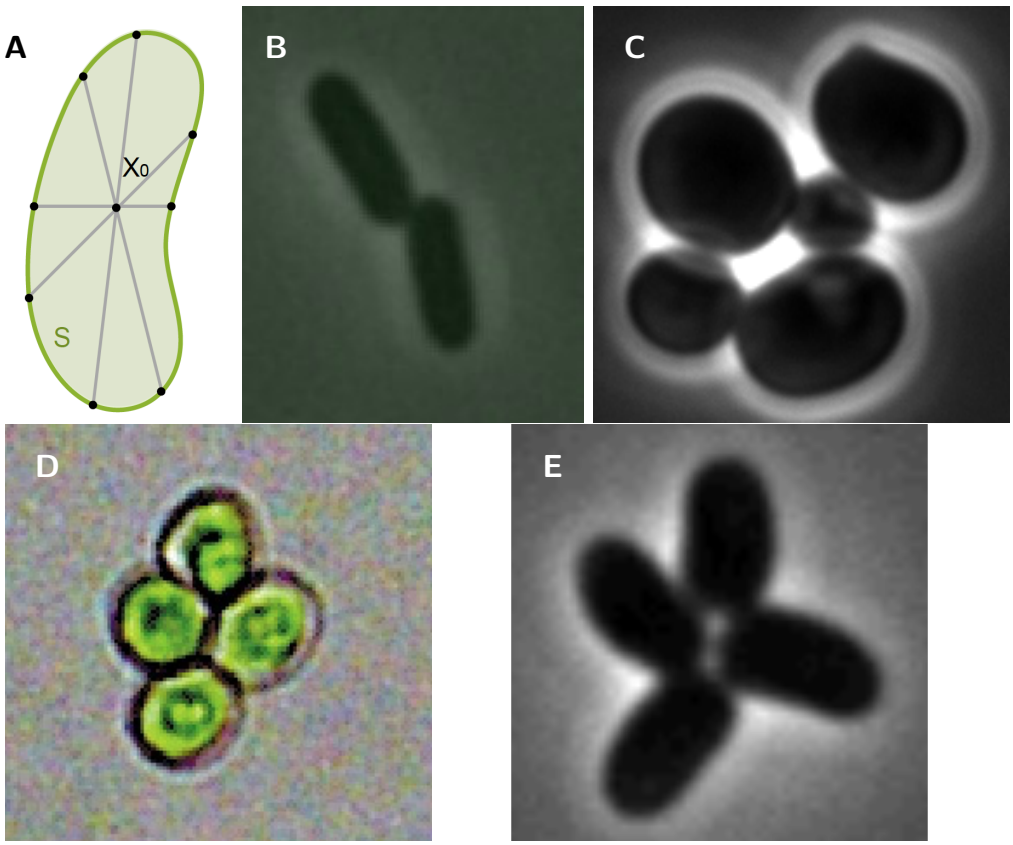


Figure 3.1.: Definition of a star-shaped object (**A**) and examples of star-shaped microorganisms: *E. coli*, *Saccharomyces cerevisiae* (*S. cerevisiae*), *Chlorella vulgaris*, *C. glutamicum*. Note that in the images, the term "star-shaped" refers to the shape of an individual cell, not to groups of cells.

Truth And Performance Level Estimation (STAPLE) [95]. STAPLE, however, does not produce uncertainty estimates, which are crucial for understanding the variability and reliability of segmentation results. This algorithm will be used as a baseline to compare our method in terms of the similarity of the resulting segmentation to the raters' opinions and the contour smoothness.

To obtain a consensus or ground truth (GT) contour from the annotation proposals of experts, we need an approach that respects and preserves the uncertainty caused by inter-rater variation while providing a smooth, natural-looking cell outline. The approach must respect and preserve the uncertainty induced by inter-rater variation while providing a naturally smooth cell outline.

3.2. Establishing ground truth for cell segmentation

As a target microorganism in this research, we choose *C. glutamicum*, which is introduced in Section 1.3.1. In the literature, *C. glutamicum* cells are described as slightly bent

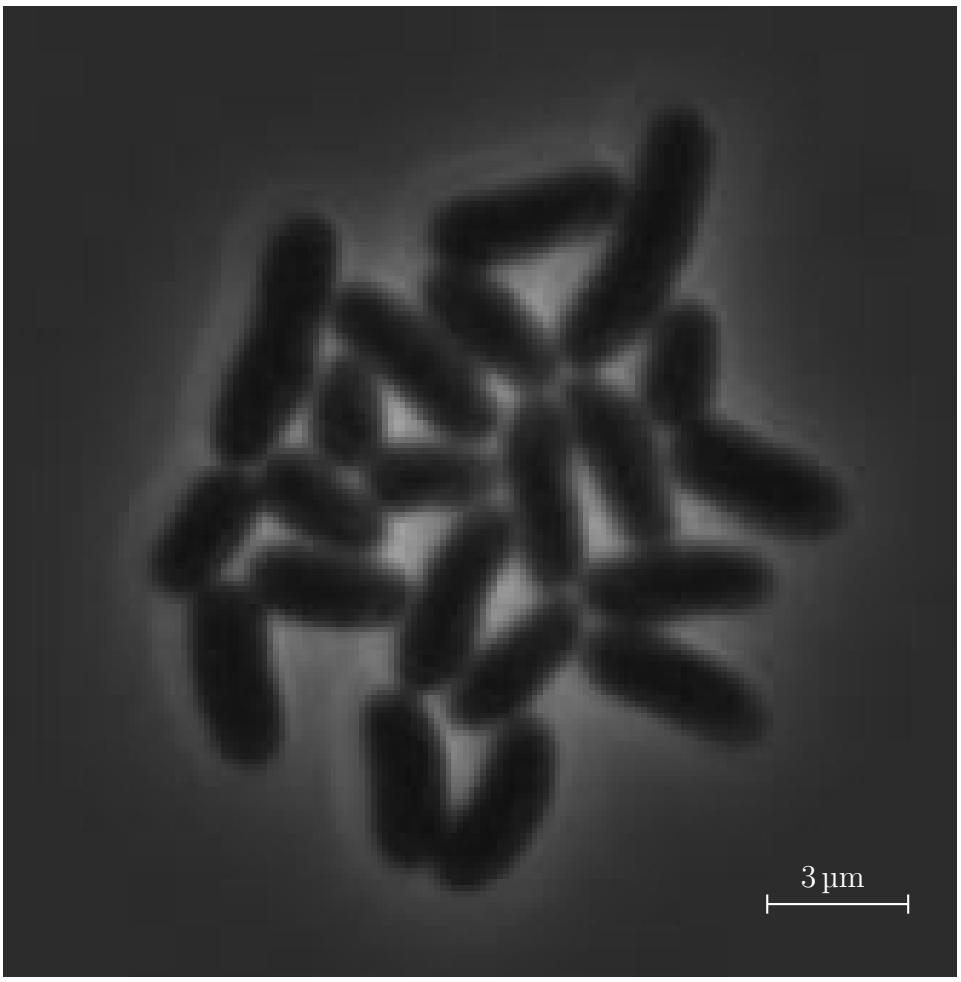


Figure 3.2.: Phase-contrast microscopy image of a *C. glutamicum* cell colony, highlighting the distinct star-shaped and rod-shaped morphology of each cell.

rods (Fig. 3.2). A significant aspect of *C. glutamicum*, is its distinctive cell shape, often described in the literature as slightly bent rods. One crucial geometric characteristic of *C. glutamicum* cells is that they are star-shaped. A set S in the Euclidean space $\mathbb{R}^n$ is called star-shaped if there exists $X_0 \in S$ such that for all $X \in S$ the line segment from X_0 to X is in S [96]. Such a point X_0 is called center of S.

C. glutamicum is not the only biologically relevant star-shaped object. There are other important star-shaped organisms, i.e., *E. coli* and *Saccharomyces cerevisiae* (*S. cerevisiae*) [97] (mother cell and buds have to be treated separately). Moreover, recent image processing methods explicitly exploit the star-shapedness [98, 99]. Motivated by this, we consider star-shaped objects in this research and use *C. glutamicum* to illustrate our proposed method. The method is not limited to this organism but applies to a wide range of rod- and spherical-shaped bacteria.

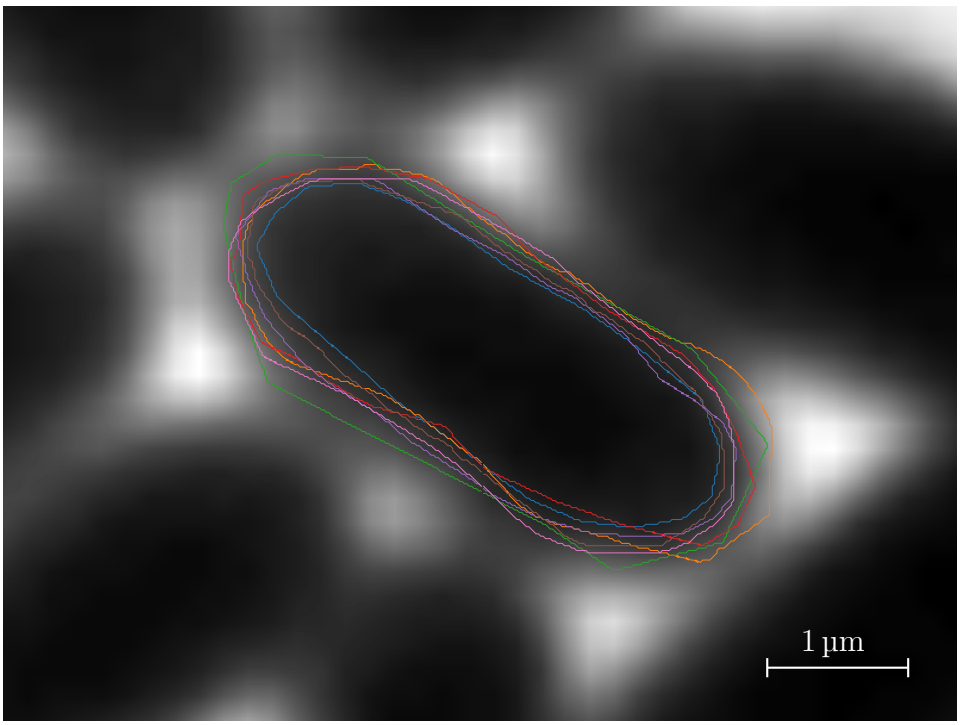


Figure 3.3.: Cutout from a phase-contrast microscopy image. Dark objects are *C. glutamicum* cells. Annotations from seven experts are shown for the central cell.

3.2.1. The polar average of star-shaped sets

An important property of star-shaped objects is that their boundary or contour can be expressed as a graph in polar coordinates. Let $R \subset \mathbb{R}^2$ be star-shaped with center $(x_0, y_0) \in R$ and boundary $C = \partial R$. The corresponding transformation to polar coordinates is

$$\begin{aligned} r &: \mathbb{R}^2 \to [0, \infty), \ (x, y) \mapsto r(x, y) = \sqrt{(x - x_0)^2 + (y - y_0)^2}, \\ \theta &: \mathbb{R}^2 \to (-\pi, \pi], \ (x, y) \mapsto \theta(x, y) = \operatorname{arctan2}(y - y_0, x - x_0). \end{aligned} \tag{3.1}$$

The $\operatorname{arctan2}(y, x)$ calculates the angle θ between the positive x-axis and the point (x, y), considering the correct quadrant of the point, returning a value from $-\pi$ to π. Since R is star-shaped, the polar transformed boundary

$$\mathcal{P} = \{(\theta(x, y), r(x, y)) : (x, y) \in C\} \tag{3.2}$$

is a graph over $(-\pi, \pi]$, i.e. there is a function $f : (-\pi, \pi] \to [0, \infty)$ such that $\mathcal{P} = \{(\theta, f(\theta)) : \theta \in (-\pi, \pi]\}$. This function f represents the radial distance as a function of the polar angle θ. This polar representation of star-shaped objects is essential for our contour averaging approach.

Let $R_1, \ldots, R_N \subset \mathbb{R}^2$ be N star-shaped sets, each representing an expert's proposal for the segmentation of a given object (cell), and all with a common center (x_0, y_0). Assuming no consistent bias exists in the rater proposals, a suitable average of the proposals should

be a good estimate of the GT segmentation. For $i = 1, \ldots, N$, let f_i be the graph function from the polar representation of ∂R_i. Then, the average of these f_i, i.e., $f := \frac{1}{N} \sum_{i=1}^{N} f_i$, is the graph function of the polar representation $\mathcal{P}$ of the contour of the averaged experts' proposals that minimizes the sum of squared distances between the proposals in the polar representation. Using the inverse transform of (3.1), i.e.

$$[0, \infty) \times (-\pi, \pi], \ (r, \theta) \mapsto (r\cos(\theta) + x_0, r\sin(\theta) + y_0), \tag{3.3}$$

on $\mathcal{P}$, we get the contour of the rater proposal average, and thus our GT estimate, i.e., the resulting contour set in Cartesian coordinates $\mathcal{C}_\mathrm{a}$ is

$$\mathcal{C}_\mathrm{a} = \{(f(\theta)\cos(\theta) + x_0, f(\theta)\sin(\theta) + y_0) \mid \theta \in (-\pi, \pi]\}. \tag{3.4}$$

3.2.2. Experimental setup

The test dataset was created to evaluate the proposed method's performance. The test dataset consists of six images of *C. glutamicum*, where each image illustrates up to six cells (Table 3.1) and parts of neighboring cells (exemplified in Figure 3.3, which shows the image corresponding to Row 3 of Table 3.1). Nine domain experts were asked to annotate (manually segment) fully present cells with a mouse or stylus pen, where each cell is supposed to be segmented separately. The annotations were performed using `Hasty.ai`, an easy-to-use online annotation platform, but choosing another annotation platform is possible.

To get an estimate of the inter-rater variability, we computed basic statistics of the segmentation proposals for some objects' morphological features. Specifically, the resulting average length and width were calculated according to [100]. The length is measured along the longest middle line of the bounding (rotated) rectangle with the minimum area and is $2.3037 \pm 0.1057\,\mu\mathrm{m}$. The width is derived as the average of eleven equidistant width segments, parallel to the short middle line of the rectangle and is $0.8966 \pm 0.0505\,\mu\mathrm{m}$. Finally, the area was computed as a sum of labeled pixels for each cell and determined to be $19.6530 \pm 2.6454\,\mu\mathrm{m}^2$.

3.2.3. Preprocessing and contour averaging

For the proposed method, the input consists of N binary masks $I_1, \ldots, I_N$, where each mask represents an expert's segmentation proposal. The top row of Figure 3.4 shows the original binary masks. Each binary mask is a matrix with entries that take values of either zero (background) or one (cell). The expected output is a binary mask that represents the average shape based on the expert annotations.

To process these inputs, the first step is to extract the contours from each binary mask. This is done by applying a discrete Laplace operator to each binary image I_i, to detect the edges of the segmented regions. Then, we convert the obtained binary images with the contours (shown in the bottom row of Figure 3.4) of the proposals to a list of (x, y) coordinates for each contour. The top row of Figure 3.4 shows the original binary segmentation masks.

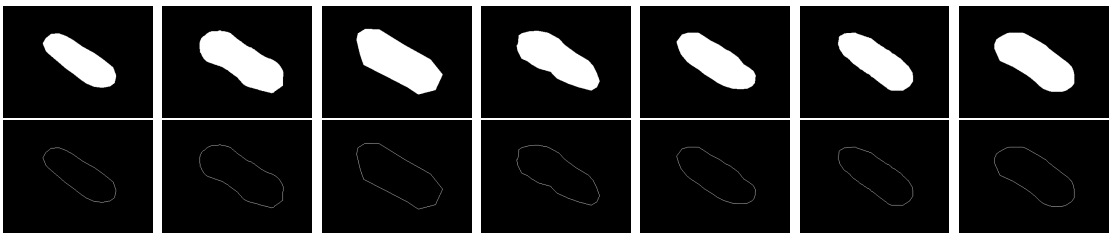


Figure 3.4.: Segmentation proposals (top), their contours (bottom).

After extracting the contours, the next step is to convert the contour coordinates to polar coordinates using the transformation defined in (3.1), where the center of mass of the sum of the masks is used as an estimate of the common center, (x_0, y_0) (note that, in a sensitivity analysis, we found that the influence of changing (x_0, y_0) was insignificant for our experiments). With this, for each binary mask I_i, we obtain a list of polar contour coordinates $(\theta_{i,1}, r_{i,1}), \ldots, (\theta_{i,K_i}, r_{i,K_i})$ (Figure 3.5(left)), where K_i is the number of points on the i-th contour. These coordinates are sorted by θ in ascending order. Here, the number of entries $K_i \in \mathbb{N}$ depends on the mask, and the spacing of the $\theta_{i,j}$ is non-uniform. Using linear interpolation, as a trade-off between simplicity and computational complexity, of the polar contour coordinate vector, we get a continuous function $R_i : (-\pi, \pi] \to [0, \infty)$ such that $R_i(\theta_{i,j}) = r_{i,j}$ for $j = 1, \ldots, K_i$. Now that we have a continuous representation, we compute the average contour in polar space. This involves calculating the average of the radial functions corresponding to each expert's proposal:

$$R_{a,j} = \frac{1}{N} \sum_{i=1}^{N} R_i(\theta_j) \tag{3.5}$$

For practical computation, we select θ as a vector of uniformly distributed points in $(-\pi, \pi]$, which should not be shorter than the longest vector of proposed object boundaries, i.e., $K \geq \max_{j=1,\ldots,N} K_j$. For simplicity, we used $K = 1000$ for all experiments to exceed the length of the longest proposed contour in our test dataset.

Finally, after averaging in polar space, the polar space coordinate vector is transformed back to real space using the inverse transformation (Equation (3.3)), rounding the results to the nearest pixel position. The resulting outline is filled to obtain the average binary mask I_a. To fill the outline, we use the `fillPoly` function from [101], which fills an area bounded by a polygonal (composed by the outline coordinates) contour. The resulting average binary mask is illustrated in Figure 3.5 (right).

3.2.4. Uncertainty-aware segmentation quality metric

One byproduct of the polar average Equation (3.5) is that each $R_{a,j}$ is associated with a standard deviation

$$\sigma_{a,j} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(R_i(\theta_j) - R_{a,j}\right)^2} \tag{3.6}$$

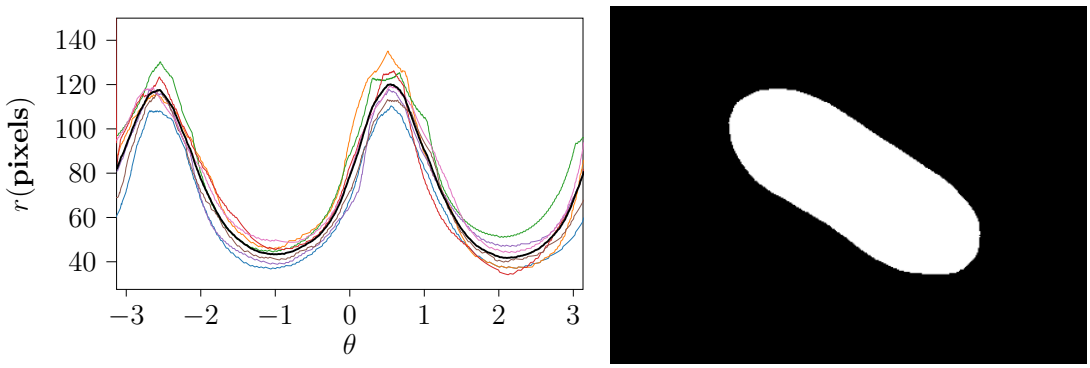


Figure 3.5.: Contours of the segmentation proposals in polar space and their average (black) (left). The resulting average of segmentation proposals (right).

encoding the uncertainty of the average contour R_a, i.e., the inter-rater variability).

Inter-rater segmentation deviation is usually inevitable. Moreover, unevenly distributed deviation may indicate varying difficulty in locating the boundary of an object in different regions (e.g., it is easier to segment cells in non-crowded colonies with no other cells around than in regions with densely packed cells), in particular when the cell touches the colony partially.

Most of the available metrics for the segmentation quality estimation are based on spatial overlap (e.g., Dice or Jacquard scores) and do not consider the region-specific type of uncertainty. However, to avoid bias, estimating the segmentation quality of a given algorithm using a "fuzzy" GT segmentation should take into account this uncertainty. In other words, an uncertainty-aware metric to estimate the segmentation performance, where the GT is composed of several experts' opinions, is desired.

We propose to use a Weighted Root-Mean-Square Error (WRMSE), where the weights are inversely proportional to the standard deviation of the object boundary, to emphasize the influence of the contour points with no (or low) variation. The suggested metric takes into account every boundary pixel while respecting the uncertainty. The Root-Mean-Square Error (RMSE) is a standard distance-based metric to evaluate the performance of an algorithm [102]. It illustrates the average distance between each of the N predicted objects' boundary points in polar space and the corresponding average boundary points. In our case, the RMSE is:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(R_{\text{pred}}(\theta_i) - R_a(\theta_i))^2} \tag{3.7}$$

where R_{pred} is the predicted contour in polar space and R_a is the GT contour. Based on this, we define the WRMSE as follows:

$$WRMSE = \sqrt{\frac{1}{\sum_{i=1}^{N} w_i}\sum_{i=1}^{N} w_j(R_{\text{pred}}(\theta_i) - R_a(\theta_i))^2} \tag{3.8}$$

where w is a vector of positive weights. To avoid division by zero and the extreme influence of low deviation regions, we use an exponential weighting of the form $w = e^{-\sigma}$, i.e.

$$w_i = e^{-\sigma_{a,i}} \tag{3.9}$$

3.3. Results

3.3.1. Ground truth averaging results

Here, we examine the results of the averaging method in terms of similarity to the segmentation proposals, using Dice score and visual contour smoothness, and compare them to baseline algorithms. As a baseline, we use the STAPLE algorithm with two different thresholds and the median of the segmentation proposals.

The Dice score between two finite sets A and B is defined as:

$$\mathrm{Dice}(A, B) = \frac{2|A \cap B|}{|A| + |B|} \tag{3.10}$$

where $|\cdot|$ denotes the cardinality (size) of the set. For the purpose of our evaluation, the average Dice score between the average mask I_a and each expert's segmentation proposal I_i is used:

$$\mathrm{Dice}(I_a, (I_i)_{i=1}^N) = \frac{1}{N} \sum_{i=1}^{N} \frac{2|I_a \cap I_i|}{|I_a| + |I_i|} \tag{3.11}$$

where I_i is the i-th expert's segmentation proposal for the given image, and I_a is the average mask.

The median of the segmentation proposals is defined as the pixel-wise median across the experts' segmentation proposals and is thus equivalent to the image average with respect to the L^1-norm. To create a binary mask out of an even number of experts' opinions (proposals), for the pixels with an equal amount of votes, the value 1 is chosen (instead of 0.5). The Dice score in Table 3.1 was calculated as the average of the Dice score for every binary instance (cell), not for the entire binary mask. Thus, for pictures 4-6, every cell was processed separately.

While all examined methods show almost indistinguishable average Dice scores (as shown in Table 3.1), the proposed method provides smoother and more naturally looking average results (Figure 3.6). The result of the proposed method does not inherit the coarse edges proposed by experts that originated due to the labeling with a mouse. The obtained smoothness is an important property, for instance, for training data generation, since ML-based algorithms could learn the artifacts that are present in the training data.

3.3.2. Uncertainty-aware metric

Since the deviation of the expert proposals used to create the GT above is mostly equally distributed over the cell boundary, we test the proposed uncertainty-aware metric with

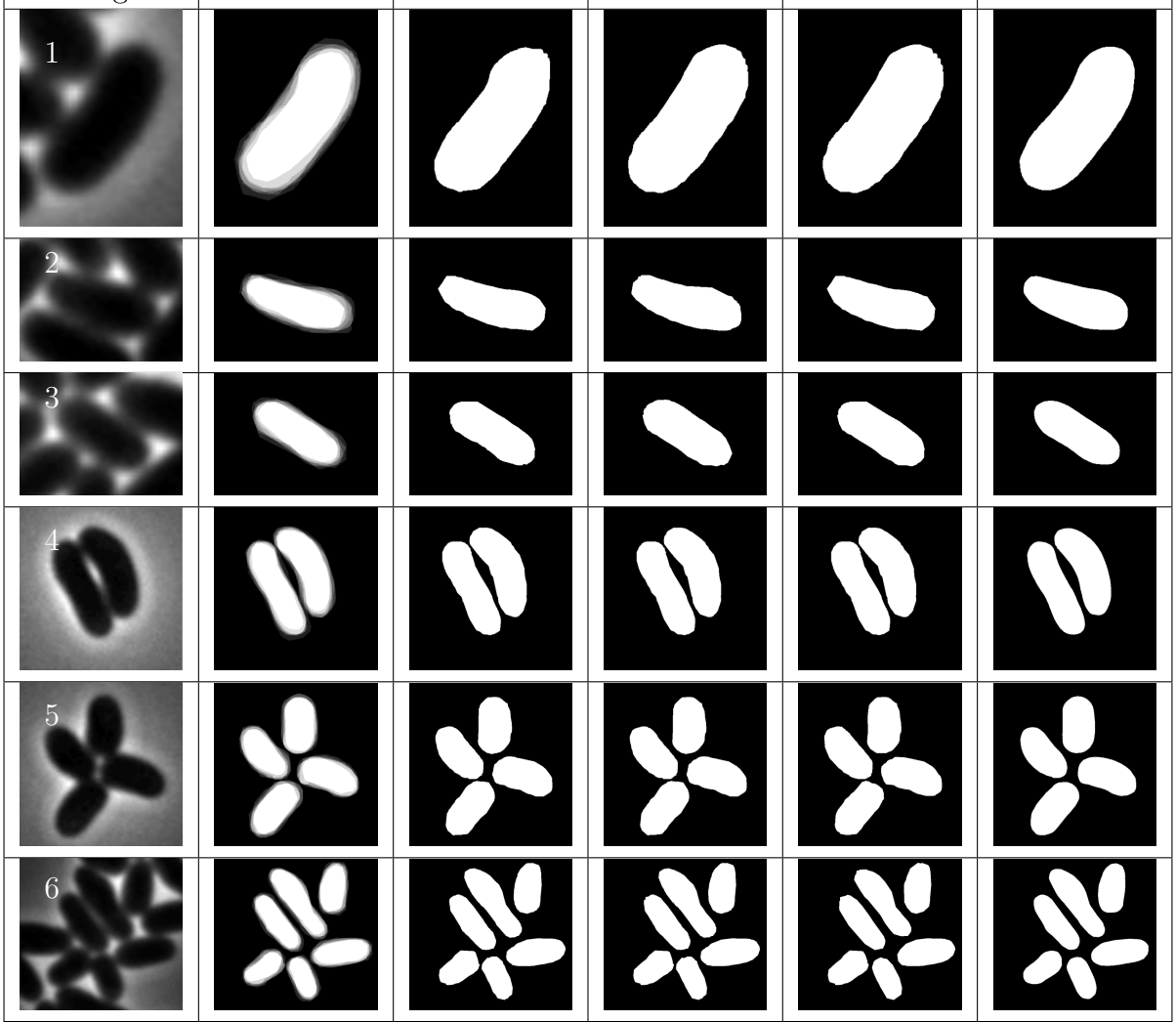
Image	Mean	Median	S 0.85	S 0.95	PA
					
Dice Score		0.9357	0.9348	0.9342	0.9331

Table 3.1.: The result of different averaging methods: Median, STAPLE with different thresholds (S 0.85 and S 0.95), and the proposed method based on the contour averaging in polar space (PA).

synthetic segmentation proposals. Five binary rod-shaped masks (segmentation proposals) were generated, where the width of the rod is well-defined, and the length deviates over the proposals (Figure 3.7 (top)).

Considering two contours (blue and red), where one matches the length of the GT, and the other matches the width, as the result of the segmentation algorithm, the WRMSE and the Dice score are calculated, cf. Table 3.2. Even though the red contour shows a better Dice score (more significant overlap with the average mask), it does not match the regions with the highest confidence (low variation), which is reflected in the WRMSE. On the other hand, the blue contour perfectly fits low variation regions, explaining the

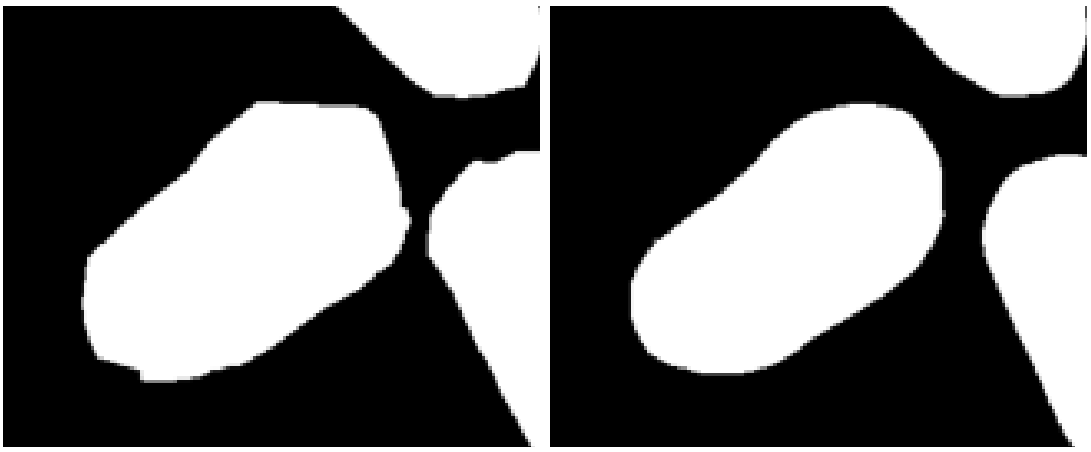


Figure 3.6.: Comparison of the results obtained by STAPLE (left) and the proposed method (right), which shows the more natural smoothness of the proposed method compared to STAPLE.

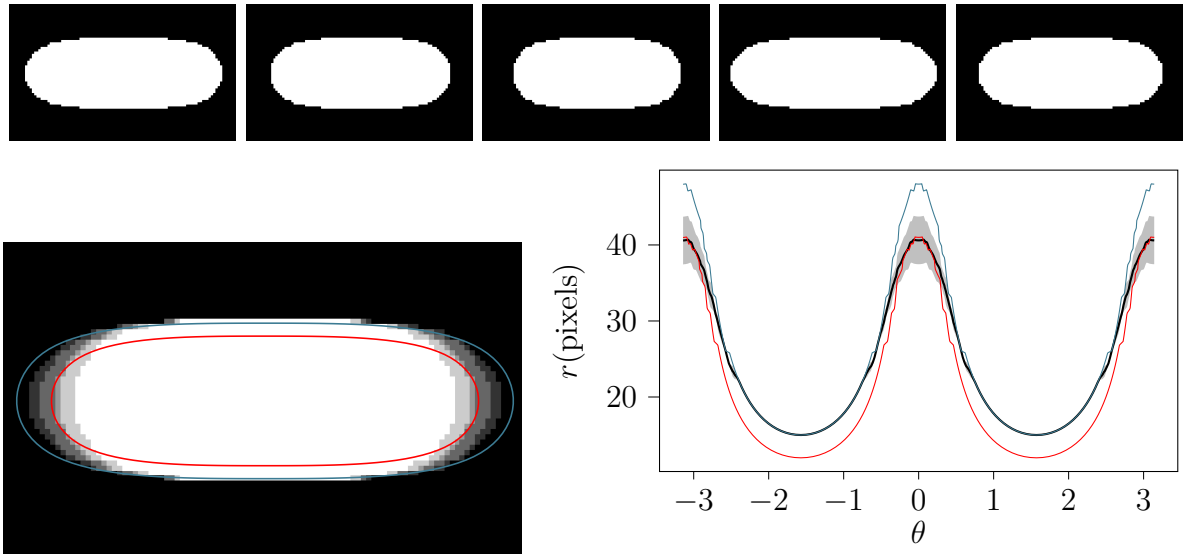


Figure 3.7.: Synthetic segmentation proposals (top). Contours to examine (red and blue) are plotted over a sum of the segmentation proposals(left). Average cell shape contour (black) with a standard deviation, and the contours of interest (right).

lower (better) WRMSE in this case.

The usability of the proposed metric is obviously limited due to the lack of several segmentation proposals per single training/validation image in most of the experiments. Consequently, we will not use this metric elsewhere in this thesis. Still, it can be an alternative option in high-precision experiments. The additional possible application of the proposed metric is its use as a loss function in the ML approaches, where the training data may be available from different sources (refined automatically, produced with other different methods, or manually created).

Table 3.2.: Dice score and WRMSE for two contours

	Blue	Red
Dice	0.9044	0.9172
WRMSE	1.0482	3.4881

3.4. Conclusions

In this chapter, we proposed an averaging method for expert segmentation proposals of microbial organisms, resulting in a smooth, naturally looking segmentation GT. The approach leverages the geometrical property of star-shapedness common to many microbial organisms and is based on contour averaging in polar space. It is robust, computationally efficient, and operates without introducing tunable parameters. The method utilizes the polar transform to find the average object outline for creating GT for training and validation purposes in segmentation tasks.

Moreover, the algorithm preserves the uncertainty (in terms of the standard deviation) of the experts' annotations, allowing for the introduction of an uncertainty-aware metric to estimate segmentation quality. This metric emphasizes regions with low variance in the GT, enhancing its reliability.

The performance of the proposed averaging method is demonstrated on time-lapse microscopy data of *C. glutamicum* and the uncertainty-aware metric on synthetic data.

4. Cell simulator for microcolony ground truth generation

This chapter is based upon the publication "CellSium: versatile cell simulator for microcolony ground truth generation" by C. C. Sachs, **K. Ruzaeva**, J. Seiffarth, W. Wiechert, B. Berkels and K. Nöh. The author of this thesis performed the YOLOv5 experiments, similarity comparison, and helped with the manuscript preparation.

4.1. Synthetic ground truth generation

Deep learning-based segmentation methods have revolutionized the field of bioimage analysis. These methods have been shown to outperform traditional approaches [103]. As introduced in Part II, generating large amounts of comprehensive GT data for training and validation requires considerable time and effort. However, synthetic data generation can effectively overcome this bottleneck.

As mentioned in Chapter 3, when it comes to cell segmentation, it is essential to create accurate masks that can distinguish individual cells from the background. Additionally, to better understand biological growth phenomena, cell simulators have been developed to model molecular aspects rather than to produce photorealistic images, e.g., [104]. While fluorescence image generators have been widely used for data augmentation in eukaryotic cells [105, 106], microbial image analysis, especially in phase-contrast microscopy, has yet to adopt synthetic image generation techniques.

This chapter introduces CellSium, a highly configurable simulator for bacterial microcolonies. Rather than just producing realistic images, CellSium models molecular aspects to better comprehend biological growth phenomena. We have demonstrated the efficiency of CellSium for object (cell) detection and semantic segmentation by training with synthetic data and validating the detection and instance segmentation results using real image data.

CellSium has multiple uses beyond image synthesis, including testing and optimizing image processing pipelines, evaluating image analysis algorithms, and developing new microscopy techniques. As a result of its flexible and modular design, CellSium has the potential to dramatically improve the efficiency and accuracy of microbial image analysis, which could lead to advancements in various fields of microbiology, biotechnology, and medicine.

4.2. Implementation and functional overview

CellSium is an agent-based simulator designed to model the growth and interactions of bacterial microcolonies. An agent-based model (ABM) refers to a computational framework in which individual cells are treated as autonomous agents. These agents operate independently, following specific rules that govern their growth, division, and interaction with neighboring cells. This approach allows for the simulation of complex colony dynamics, where the collective behavior of the system emerges from the interactions of individual cells.

4.2.1. Cell representation and growth models

Each cell in CellSium is represented as a Python object, encapsulating properties such as shape, size, and growth behavior. These properties are implemented using object-oriented programming paradigms, where mix-ins enable the flexible combination of different cell behaviors and characteristics. The cell geometries are modeled using closed polygonal chains, which allow for a wide range of bacterial shapes, including straight rods, bent rods, circular (coccoid), and ellipsoid forms. This flexibility is crucial for accurately simulating different bacterial species and their unique growth patterns.

To model the growth of cells, CellSium incorporates phenomenological cell size homeostasis models like the "timer" and "sizer" models, which describe how cells regulate their growth and division cycles [107]. During each simulated time step, the `grow()` method is invoked for each cell, updating its geometry based on the selected growth model. The physical placement of cells within the colony is then computed using a physics engine, ensuring that cells are positioned in a way that reflects realistic mechanical constraints and interactions.

4.2.2. Output capabilities

CellSium offers a range of output options tailored to different research needs. These include:

- Realistic phase contrast images: Simulated images that closely mimic the appearance of bacterial cells under a phase contrast microscope.
- Time-lapse videos: Visualizations that capture the dynamics of colony growth over time.
- TrackMate XML files: Data exports that facilitate cell tracking analyses in software like TrackMate [108].
- Computational fluid dynamics (CFD) Geometries: Simulation-ready input geometries for use in CFD simulations, supporting studies on fluid dynamics around growing colonies.

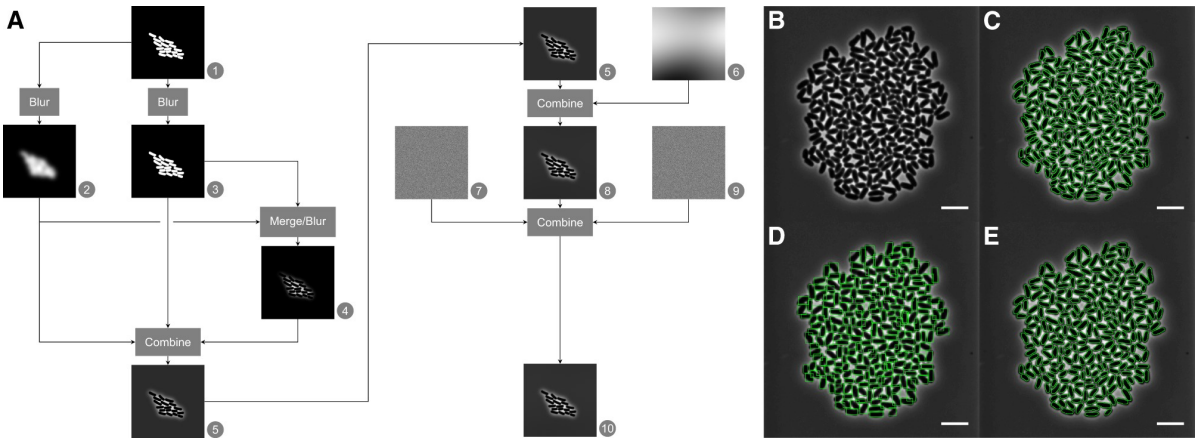


Figure 4.1.: (A) The configurable image generation pipeline for realistic phase contrast image generation. (B) Input image, a microcolony of *C. glutamicum*; (C) GT cell segmentation; (D) YOLOv5 detection results and (E) Mask R-CNN segmentation results, both networks trained on synthetic data generated with CellSium. The figure is taken from [86].

- Binary masks: Binary masks are generated to represent the shape and location of each cell within the simulated images, which serve as ground truth for training and validation of segmentation algorithms.
- Bounding boxes: Bounding boxes are used for object detection tasks, where the model predicts the location of objects within an image. Bounding boxes are generated around each cell, providing the training data for detection frameworks like YOLOv5.

4.2.3. Image generation process

The image generation process in CellSium is designed to produce realistic phase contrast images by simulating various imaging artifacts and noise characteristics. The process is illustrated in Fig. 4.1(A):

- First, "perfect" (noiseless) image is generated, which is augmented by adding different kinds of noise (1-5).
- The typical phase contrast "halo" around the cells is generated by Gaussian blurring of the image (2-4) to mimic a phase contrast image (5)
- Next, uneven illumination is added (6), along with additive/multiplicative Gaussian noise (7,9), which results in a realistic phase contrast bacterial cell image (10).

This sequence of steps results in a realistic phase contrast bacterial cell image. All noise models and simulation parameters are highly configurable, allowing for tailored and flexible image generation.

This allows for realistically looking image generation and makes *CellSium* readily transferable to other imaging modalities. Fluorescence can be optionally simulated using a Gaussian point spread function [86]. Further implementation details can be found in [15].

4.3. *CellSium* as GT generator

To assess the applicability of *CellSium* as a GT generator, a data set was generated using the `YOLOOutput` and `COCOOutput` modules (128 images, 512x512, 0.09 µm/pixel, 0 to 512 cells per frame). Synthesized images are verified to have a similar intensity distribution compared to real images with a similar cell/background ratio and illustrated in Figure 4.2. The synthesized outputs, the cells' binary masks (for instance segmentation task) or bounding boxes (for detection task) together with the simulated image of the cell colony were then used to train the object detection/segmentation frameworks YOLOv5 and the Mask R-CNN module of MMDetection [109].

YOLOv5 (You Only Look Once v5) [110] is an object detection model that identifies and localizes objects in images. The "You Only Look Once" approach means that the model processes an entire image in a single forward pass, predicting both the bounding boxes for detected objects and the corresponding class probabilities. It is implemented in PyTorch and is highly configurable, allowing users to adjust parameters such as the model size, the confidence threshold for predictions, and the non-maximum suppression (NMS) threshold, more details on the YOLOv5 usage will be introduced in Chapter 5

Mask R-CNN [63] is a popular instance segmentation model that extends Faster R-CNN, which is a model designed for object detection. In addition to detecting objects and their bounding boxes, Mask R-CNN also generates a binary mask for each detected object, making it particularly suitable for tasks where pixel-level segmentation is required. The model consists of two stages: Region Proposal Network (RPN), which proposes candidate object bounding boxes (regions of interest or RoIs); and Classification and Mask Prediction, where for each RoI, the model classifies the object within the bounding box and predicts a mask for it.

As test data, a microcolony image of *C. glutamicum* was used (Fig. 4.1B), which was interactively segmented using the Trainable Weka Segmentation [111] and then hand-corrected (Fig. 4.1C). The YOLOv5 (You Only Look Once v5) [110] net was trained in *yolo5l* configuration for 300 epochs, and the test data was predicted with an Intersection over Union (IoU) threshold for Non Maximum Suppression (NMS) of 0.6 and a confidence threshold of 0.001. A Mask R-CNN (Region-Convolutional Neural Network) was trained using the MMDetection implementation (*open-mmlab/mmdetection* v2.17.0) for 13 epochs. The test predictions for Mask R-CNN were configured with increased proposal counts to handle the number of cells: nms_pre, rpn.max_per_img, and rcnn.max_per_img were set to 12000, 4000, and 3000, respectively, with a score threshold of 0.5.

The mean average precision (mAP) results are given in Table 4.1.

The YOLOv5 network achieved a mean Average Precision (mAP50) of 0.994, while

the Mask R-CNN reached a mAP50 of 0.987 for both bounding box detections and segmentation.

4.4. Similarity of real and synthesized Images

A quantitative comparison of the similarity between real microscopic images and the synthetic images, generated with *CellSium*, is based on comparing their corresponding intensity histograms. To ensure a fair comparison, we select image crops containing similar amounts of cells and cell-to-background ratios. Specifically, we manually extract the crops from both the synthetic and real images, maintaining consistent cell and background proportions across them. Figure 4.2 shows the histograms for one pair of synthetic and real image crops, along with their correlation plots. For the histograms, we can obtain a 0.75 Pearson Correlation Coefficient (PCC), indicating high similarity. The Pearson Correlation Coefficient (PCC) is calculated as follows:

$$\mathrm{PCC}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{4.1}$$

where X_i and Y_i are the intensity counts of the pixels in the histograms of the real and synthetic images, respectively, $\bar{X}$ and $\bar{Y}$ are the mean values of these counts, $n = 50$ is the number of bins in the histograms.

The PCC measures the linear correlation between X and Y, with values ranging from -1 to 1. A PCC closer to 1 indicates a strong positive correlation, while a PCC closer to -1 indicates a strong negative correlation. A PCC of 0 indicates no linear correlation between X and Y.

The PCC of the histogram pairs can be used as a metric to optimize the parameters of *Cellsium* to support other microscopic modalities (e.g., bright-field imaging) or to develop tailored noise models.

Table 4.1.: Mean average precision (mAP) results of YOLOv5 and Mask R-CNN networks, trained with CellSium data.

	Bounding Box		**Segmentation**	
Network	**$\mathbf{mAP_{0.5:0.95}}$**	**$\mathbf{mAP_{0.5}}$**	**$\mathbf{mAP_{0.5:0.95}}$**	**$\mathbf{mAP_{0.5}}$**
YOLOv5	0.667	0.994	–	–
Mask R-CNN	0.465	0.987	0.470	0.987

4.5. Summary and outlook

CellSium is a microcolony simulator primarily designed to generate data sets for bacterial image analysis. The simulated training data has shown strong correlation with real data

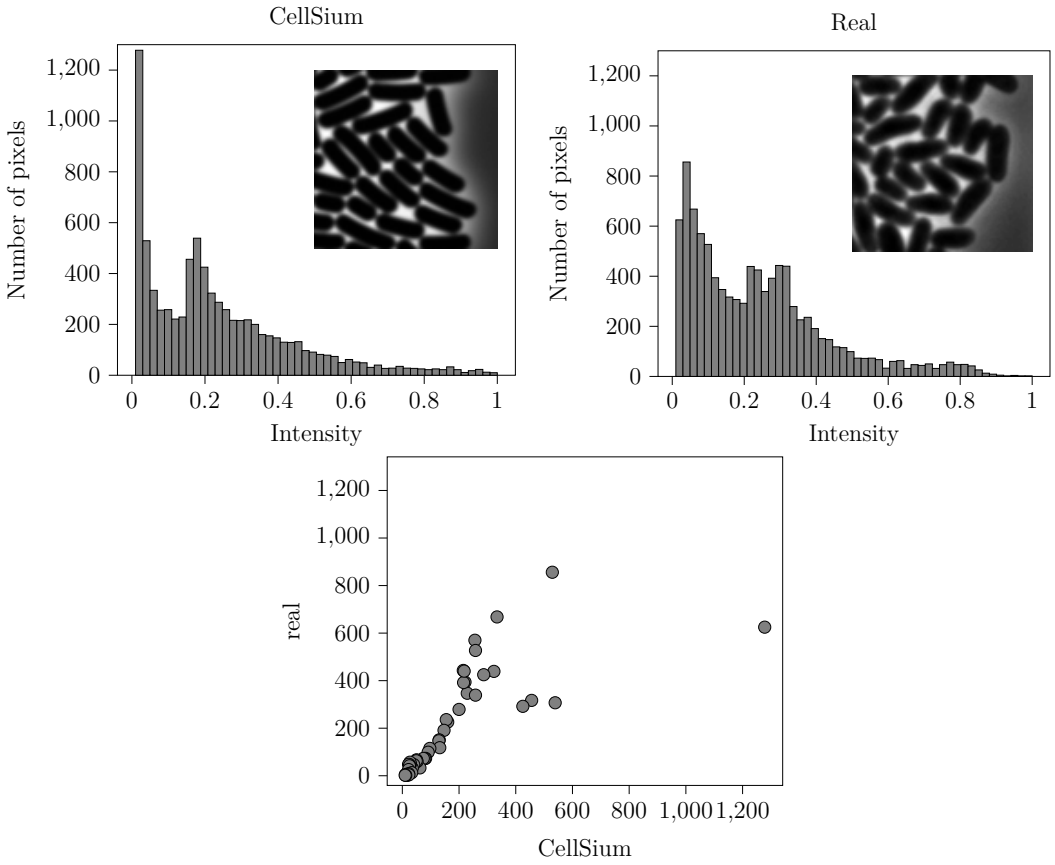


Figure 4.2.: The histograms (50 bins) of the normalized synthetic and real images (top) and their correlation plot (bottom). The correlation plot shows the relationship between the pixel intensities of the synthetic and real images, providing a quantitative measure of similarity.

in terms of intensity histograms. It has been successfully used to train state-of-the-art object detectors and segmentation frameworks like YOLOv5 and Mask R-CNN.

The main advantage of using *CellSium* lies in its ability to generate labeled data with desired characteristics in any desired quantity, which is much more efficient than manually annotating image sequences. This benefits both training and method verification in the field of bacterial image analysis. Additionally, it allows developers of bioimage analysis algorithms to benchmark and validate their segmentation and tracking methods in configurable settings.

The pluggable architecture of *CellSium* offers great flexibility, enabling users to implement other cell types and customize cell behavior within growing microcolonies. It can also be seamlessly integrated into deep learning training loops for continuous procedural training data generation.

While *CellSium* currently excels in training networks tailored to specific imaging modalities, future advancements may involve implementing generative adversarial networks (GAN) [112] or diffusion models [113] to achieve a higher level of realism.

Part III.

Segmentation

As mentioned in Chapter 1, microfluidic single-cell analysis with time-lapse microscopy is a versatile tool for studying cellular processes with spatial and temporal resolution. The analysis is used to reveal heterogeneity and cell dynamics in microbial populations with respect to growth, gene expression, cell interactions, production, or regulation [114, 115]. The ability to create large-scale datasets using automated time-lapse microscopy requires high-throughput extraction of individual cell features. To extract individual cell features, each cell must be segmented separately. Therefore, robust, efficient, and automatic multi-object segmentation for quantitative single-cell characterization has become crucial [87].

Most traditional semantic segmentation methods are general-purpose approaches and thus do not take into account prior knowledge of the object geometry. Consequently, this can lead to an object geometry of an unphysiological shape. Going unnoticed, this can lead to biased results, e.g., cell area over time for image analysis in biological applications. In the case of state-of-the-art ML-based instance segmentation methods such as Mask R-CNN [63], where the object geometry can be learned implicitly, a large amount of training data is required. Reference (benchmark) and training datasets specific to the target microorganism and the microscopy settings are often publicly online for microbial systems. As an alternative, the manual creation of the training or validation data for segmentation is time-consuming and labor-intensive, as discussed in the previous part. In particular, in the case of low resolution and low signal-to-noise ratio data, annotating images (drawing cell outlines) is difficult, even for the domain experts. Therefore, an approach that only needs an easy-to-create training dataset and encodes the available knowledge on cell morphology to preserve geometrical features for each segmentation instance (i.e., geometry-aware cell segmentation) with minimal supervision entails various benefits.

In this part, we present the development of a hybrid approach for single-cell segmentation in live-cell imaging in Lab-on-chip devices, discussed in Chapter 5, and the adaptation of this approach to robotic screening systems for segmenting both cell bodies and their inclusion bodies, which is detailed in Chapter 6.

5. A hybrid multi-object segmentation framework with model-based B-splines

This chapter is based upon the publication "A Hybrid Multi-Object Segmentation Framework with Model-Based B-Splines for Microbial Single Cell Analysis" by **K. Ruzaeva**, K. Nöh and B. Berkels. The author of this thesis developed and tested the method described therein and prepared the manuscript.

In this chapter, we introduce a hybrid approach for multi-object segmentation in single-cell analysis. The primary objective is to enhance the efficiency and accuracy of single-cell segmentation in large datasets generated by automated time-lapse microscopy. Here, we divide multi-object segmentation into two steps: detection and segmentation. As the first step, we use a machine learning-based detection approach. The main reason for splitting the detection step is to facilitate the use of a variational segmentation method. Variational segmentation methods, which are optimization-based techniques that segment objects by minimizing an energy function, and are introduced in Section 2.2.2, can be computationally infeasible when dealing with a large number of objects per image or region of interest [116]. These methods often require a time-consuming initialization and user interaction (e.g., manually drawing the object contours). By automating the initialization through the detection step, we can significantly streamline this process [117]. Another advantage of this two-step process is that it reduces computation time and helps prevent failed segmentation caused by the inclusion of parts of neighboring cells. Additionally, the detector serves as a fast cell counter, providing important technological metrics for microfluidic experiments [118].

Another advantage of splitting is that the segmentation can be done in parallel for the cells since each detected bounding box contains only one full cell, which is then processed independently.

5.1. Implementation details

5.1.1. Detection

As a detection framework, we use YOLOv5 [110], which is the fastest detection framework available at the time of developing this method. Furthermore, we substantiated its efficacy

with simulated data in the Chapter 4. Nevertheless, other choices for the detection method are surely possible. The training and validation data were generated by the synthetic cell renderer CellSium, which is described in detail in Chapter 4. In addition, a manually labeled real dataset was used. The overall training set size is 141 images, where 11 images are manually labeled. Figure 5.1 exemplifies the training data.

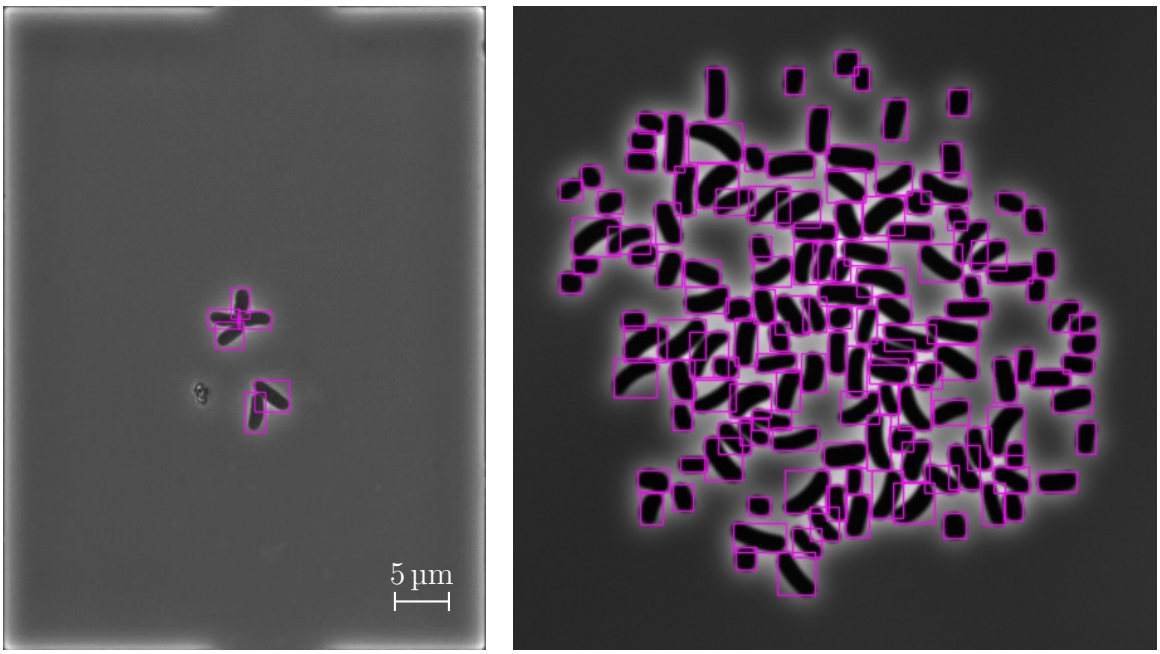


Figure 5.1.: Examples of labeled training images: Phase-contrast microscopy image of *C. glutamicum* (left) and synthetic image (right) generated by CellSium with bounding boxes.

5.1.2. Objective function

After detection, each detected cell is individually segmented using a variational method. The YOLO model provides n bounding boxes as output, from which we extract n corresponding image tiles, denoted as $I_1, \ldots, I_n$. Each tile I_i, for $i \in \{1, \ldots, n\}$, serves as the input for the segmentation process.

Variational methods formulate the given task, here, finding the cell border ($\mathcal{C}$, where $\mathcal{C} \subset \mathbb{R}^2$), as an optimization problem:

$$\mathcal{C} \in \arg\min_{\tilde{\mathcal{C}} \in \mathcal{A}} F(\tilde{\mathcal{C}}) \tag{5.1}$$

where $\mathcal{A}$ denotes the set of admissible curves. Since $\mathcal{C}$ is a continuous curve, solving this problem numerically requires a finite-dimensional representation. For this reason, we will later introduce a parametrization of $\mathcal{C}$, which transforms the optimization problem into one over a finite number of parameters.

This transformation is essential for practical computation, as it enables the use of efficient numerical optimization techniques.

In our case, the idea is to define an objective function F that measures how well our conditions on the segmentation are fulfilled. The construction of the objective involves formulating conditions combined with how to measure them. We then apply optimization procedures to obtain an optimal segmentation of the central cell from the background (and parts of neighboring cells if present (Figure 5.2) in the detection output).

Specifically, the input is one image tile, cropped according to the detected bounding box. Each image tile contains only one full cell, roughly centered in the tile and possibly parts of neighboring cells. Therefore, no additional cropping or centering is required, as it is handled by YOLO. Let $I : \Omega \to \mathbb{R}$ denote the current image tile and $\Omega \subset \mathbb{R}^2$ the corresponding image domain.

Our objective function F is the sum of three terms:

$$F(\mathcal{C}) = F_{\mathrm{CE}}(\mathcal{C}) + w_R \cdot F_{\mathrm{RE}}(\mathcal{C}) + w_D \cdot F_{\mathrm{GE}}(\mathcal{C}) \tag{5.2}$$

Here, the terms are responsible for edge detection (contour-based term, F_{CE}), for enclosing a region with given properties (region-based term, F_{RE}), and for selectively enclosing a single object of interest (geodesic distance-based term, F_{GE}).

The selection of the weights w_R and w_D is described in Section 5.2.2.

Contour-based term Object boundaries often coincide with edges in images, which are reflected by large image gradients [119, 117]. Hence, our objective F should maximize the average image gradient intensity along the contour. The corresponding term of the objective function, which encourages the contour to go through regions of high image gradient, is

$$F_{\mathrm{CE}}(\mathcal{C}) = \frac{1}{\mathrm{len}(\mathcal{C})} \int_{\mathcal{C}} \frac{1}{(|\nabla I(c)| + \varepsilon)^k} \mathrm{d}\mathcal{H}^1(c). \tag{5.3}$$

Here, $|\nabla I|$ is the norm of the image gradient (Figure 5.2 c), $\varepsilon = 0.001$ a regularization parameter to prevent division by zero, $k = 1.5$ an empirically chosen power, $\mathrm{len}(\mathcal{C})$ the length of $\mathcal{C}$ and $\mathcal{H}^1$ the one-dimensional Hausdorff measure. To ensure the integral is well-defined, we assume $I \in L^\infty(\Omega)$, a Lebesgue space that provides the necessary regularity.

Region-based term In the case of a crowded cell colony (cells touching each other), F_{CE} attracts the contour also to high-intensity gradient regions corresponding to the border of cells in the neighborhood of our target cell. We use the additional term F_{RE} to counter this undesired behavior. Such a region energy encodes our prior knowledge of the image intensities. Since the image was recorded in bright field mode, cells have lower intensities than the surrounding background, minimization of the average intensity inside the contour is beneficial. The corresponding term in the objective function is

$$F_{\mathrm{RE}}(\mathcal{C}) = \frac{1}{\mathrm{vol}(\mathcal{R}(\mathcal{C}))} \int_{\mathcal{R}(\mathcal{C})} I(x) \mathrm{d}x \tag{5.4}$$

Here, $\mathcal{R}(\mathcal{C})$ denotes the region enclosed by $\mathcal{C}$, and $\mathrm{vol}(\mathcal{R}(\mathcal{C}))$ is the volume of that region.

Geodesic distance term Neither F_{CE} nor F_{RE} can distinguish the target cell in our current single-cell image tile from neighboring cells that may also be visible in the tile. To address this issue, we add a second region-based term, which we call the geodesic distance term. This term aims to prevent the contour from enclosing any point that would have to be connected to a marker $M = (M_1, M_2)$ through a high gradient region. In this context, the marker is a reference point inside the target cell. To ensure the marker is within the target cell and to avoid user interaction, we choose the marker to be the center of the bounding box [120].

The geodesic distance map D (Figure 5.2) is obtained with the geodesic distance transform by solving the Eikonal equation. The Eikonal equation is a partial differential equation given by:

$$|\nabla D(x)| = \lambda \cdot (|\nabla I(x)| + \varepsilon)$$

where $D(x)$ is the geodesic distance at point x and $\lambda \times (|\nabla I(x)| + \varepsilon)$ is a weighting function that depends on the image gradient. In our implementation, λ is an empirically chosen image gradient weighting parameter. The distance map D is defined with the condition that $D(M) = 0$ at the marker position M, ensuring that the geodesic distance is measured relative to this marker. To implement this, we use a raster scan algorithm with the single-cell image tile as input and $\lambda = 0.8$, as described in the code from [121].

$$F_{\mathrm{GE}}(\mathcal{C}) = \frac{1}{\mathrm{vol}(\mathcal{R}(\mathcal{C}))} \int_{\mathcal{R}(\mathcal{C})} D(x)\mathrm{d}x \tag{5.5}$$

Even though F_{RE} and the geodesic distance term F_{GE} seem to be very similar in structure, they address different aspects of the image data. The term F_{RE} minimizes the average intensity within the contour, leveraging the fact that cells have lower intensities compared to the background. In contrast, F_{GE} uses the geodesic distance to prevent the contour from enclosing regions that are not connected to the marker via low-gradient paths. This distinction is crucial, as illustrated in Figure 5.2, where F_{GE} helps to differentiate the target cell from its neighbors. Their combination significantly improved the segmentation performance in our experiments.

Integrals over $\mathcal{R}(\mathcal{C})$ can be rephrased using the divergence theorem as line integrals over $\mathcal{C}$ [117]. Let $\mathcal{R} \subset \mathbb{R}^2$ be a compact set with a piece-wise smooth boundary and $\nu = (\nu_1, \nu_2)$ be the outer normal of $\mathcal{R}$. For $f : \mathbb{R}^2 \to \mathbb{R}$ Lebesgue integrable, let $f^x(x, y) = \int_0^x f(t, y)\,\mathrm{d}t$ and $f^y(x, y) = \int_0^y f(x, t)\,\mathrm{d}t$. Then, for $F := \frac{1}{2}(f^x, f^y)$, we get $\operatorname{div} F = f$ and thus

$$\int_{\mathcal{R}} f\,\mathrm{d}(x, y) = \frac{1}{2} \int_{\partial\mathcal{R}} (f^x \nu_1 + f^y \nu_2)\,\mathrm{d}\mathcal{H}^1(c). \tag{5.6}$$

By applying Equation (5.6), we can rephrase F_{RE} and F_{GE} as integrals over $\mathcal{C} = \partial\mathcal{R}(\mathcal{C})$. This transformation is advantageous because it allows us to use the parametrization of $\mathcal{C}$ for numerical computation. Consequently, we can compute the integrals over $\mathcal{C}$ more efficiently and accurately than directly integrating over $\mathcal{R}(\mathcal{C})$.

5.1.3. B-splines

A B-spline of order $k+1$ is a piece-wise polynomial function of degree k and is also continuously differentiable $k-1$ times. Closed cubic ($k=3$) B-spline curve-based segmentation demonstrated solid performance in medical data segmentation [117, 122, 123], as it provides both border smoothness (compared to Hermite splines), flexibility, and control over speed against accuracy by changing the number of spline control points. Thus, we propose to represent the cell border $\mathcal{C}$ as a closed cubic B-spline curve.

For a given number N of control points $P = (P_0, P_1, \ldots, P_{N-1})$, where $P_i \in \mathbb{R}^2$, and a knot vector u, where $u \in \mathbb{R}^N$ and the elements of which are in increasing order, we define a B-spline curve of degree k as the linear combination of B-splines weighted with the points. The B-spline curve is given by:

$$S : [0, N-1] \to \mathbb{R}^2, t \mapsto S(t) = \sum_{i=0}^{N-1} P_i B_i^k(t) \tag{5.7}$$

Here, B_i^k is the i-th B-spline basis function of degree k corresponding to the knot vector u and can be computed with the Cox-de Boor recursion formula [124], for all $t \in \mathbb{R}$:

$$B_i^k(t) = \frac{t - u_i}{u_{i+k-1} - u_i} B_i^{k-1}(t) + \frac{u_{i+k} - t}{u_{i+k} - u_{i+1}} B_{i+1}^{k-1}(t) \tag{5.8}$$

$$B_i^1(t) = \begin{cases} 1 & \text{for } u_i \leq t < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

where u_i - an element of an increasing set of real numbers also called the knot sequence (e.g. $u_1 = 1$ if $u = (1, 2, 3, 4, \ldots)$ in case of the uniform spline). t - parameter variable

The cell border corresponding to the curve is then $\mathcal{C} = \{S(t) : t \in [0, N-1]\}$. Since we are representing a closed curve, it is essential to ensure that $S(0) = S(N-1)$ to form a continuous loop. If $u_{i+1} - u_i$ is constant and does not depend on the distance between control points, the B-splines are called uniform.

Discrete representation of a closed B-spline curve

In this work, we use uniform cubic ($k=3$) B-Splines. Due to the uniformity, the B_i^3 are just shifted copies of each other. Combined with the fact that we are closing the curve (by putting the first k control points to the end of the control point vector) and a uniform knot vector, we get

$$B_i^3(t) = B^3_{(N+i-\lfloor t \rfloor) \bmod N}(t - \lfloor t \rfloor) \tag{5.10}$$

for $t \in [0, N]$. Here, $\lfloor t \rfloor$ is the Gauss bracket

The cubic B-splines for the knot vector $u = (0, 1, 2, 3)$ and $t \in [0, 1]$ are

$$\begin{aligned} B_0^3(t) &= \tfrac{1}{6}(1-t)^3 & B_1^3(t) &= \tfrac{1}{6}(3t^3 - 6t^2 + 4) \\ B_2^3(t) &= \tfrac{1}{6}(-3t^3 + 3t^2 + 3t + 1) & B_3^3(t) &= \tfrac{1}{6}t^3 \end{aligned} \tag{5.11}$$

Moreover, we have $B_i^3(t) = 0$ for $i > 3$ and $t \in [0, 1]$. This leads to

$$S(t) = \sum_{i=0}^{3} P_{(i+\lfloor t \rfloor) \mod N} B_i^3(t - \lfloor t \rfloor) \text{ for } t \in [0, N] \tag{5.12}$$

To evaluate the integrals over $\mathcal{C}$, given by B-Spline, an approximation of $\mathcal{C}$ with a discrete curve is needed. To obtain a discrete curve, Equation (5.12) is evaluated at a finite number of equidistant t in $[0, N-1]$, i.e. $t_i = i/n$ for $i = 0, \ldots, (N-1)(n-1)$, using Equations 5.11. Since $t - \lfloor t \rfloor \in [0, 1)$ and $t_{i+kn} = t_i + k$, $B_i^3(t_i)$ only needs to be evaluated for $i = 0, \ldots, n-1$.

A discrete representation of the closed uniform B-spline coordinates can be derived from the matrix:

$$B = \begin{pmatrix} B_0^3(t_0) & B_0^3(t_1) & \ldots & B_0^3(t_{n-1}) & 0 & 0 & \ldots & B_1^3(t_{n-1}) \\ B_1^3(t_0) & B_1^3(t_1) & \ldots & B_1^3(t_{n-1}) & B_0^3(t_0) & B_0^3(t_1) & \ldots & B_2^3(t_{n-1}) \\ B_2^3(t_0) & B_2^3(t_1) & \ldots & B_2^3(t_{n-1}) & B_1^3(t_0) & B_1^3(t_1) & \ldots & B_3^3(t_{n-1}) \\ B_3^3(t_0) & B_3^3(t_1) & \ldots & B_3^3(t_{n-1}) & B_2^3(t_0) & B_2^3(t_1) & \ldots & 0 \\ 0 & 0 & \ldots & 0 & B_3^3(t_0) & B_3^3(t_1) & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & B_0^3(t_{n-1}) \end{pmatrix} \tag{5.13}$$

Therefore, the coordinates of closed uniform B-spline for a given sequence of parameter t values can be calculated using the expression:

$$S(P) = P \cdot B \tag{5.14}$$

where

$$P = \begin{pmatrix} P_{x0} & P_{x1} & \ldots & P_{x(N-1)} \\ P_{y0} & P_{y1} & \ldots & P_{y(N-1)} \end{pmatrix} \tag{5.15}$$

is 2-D array of Spline control point coordinates and

$$S(P) = \begin{pmatrix} S_{x0} & S_{x1} & \ldots & S_{x(N-1)\cdot(n-1)} \\ S_{y0} & S_{y1} & \ldots & S_{y(N-1)\cdot(n-1)} \end{pmatrix} \tag{5.16}$$

is the discrete spline coordinates matrix. The described discretization for a given number of points per spline segment gives us the possibility to formulate Equations 5.3; and 5.4 and 5.5 using the (5.6) first, in discrete form, which will be given in Section 5.1.7.

5.1.4. Geometrical model

As mentioned above, *C. glutamicum* cells can be represented with a simple geometrical model, i.e., as slightly bent rods. We propose to exploit this prior knowledge of geometry by modeling this shape as a closed B-spline curve with six control points (cf. Figure 5.2), where the control points are parametrized using eight parameters: length (consisting of 2 segment lengths: (l_1 and l_2), width (w), two curvature parameters (d, e), center (c_x, c_y), and rotation angle α. The parameters are defined in the following sets: $l_1, l_2, w \in [0, \infty)$, $d, e \in \mathbb{R}$, $\alpha \in [0, 2\pi)$. Denoting the parameter vector by $\theta = (c_x, c_y, l_1, l_2, w, d, e, \alpha)$, the resulting coordinates of the six control points $P(\theta)$ are calculated using Equation (5.17). Note that broadcasting (the smaller vector/matrix is automatically expanded to match the size of the larger one) is assumed when adding vectors/matrices of different sizes:

$$P(\theta) = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & -\cos\alpha \end{pmatrix} \cdot \begin{pmatrix} \tilde{P}_x(\theta) - c_x \\ \tilde{P}_y(\theta) - c_y \end{pmatrix} + \begin{pmatrix} c_x \\ c_y \end{pmatrix} \tag{5.17}$$

where

$$\tilde{P}(\theta) = \begin{pmatrix} \tilde{P}_x(\theta) \\ \tilde{P}_y(\theta) \end{pmatrix} = \begin{pmatrix} \frac{w}{2} & \frac{w}{2} - d & -\frac{w}{2} - d & -\frac{w}{2} & -\frac{w}{2} - e & \frac{w}{2} - e \\ 0 & l_1 & l_1 & 0 & -l_2 & -l_2 \end{pmatrix} \tag{5.18}$$

It is important to mention that the idea of parameterizing the geometry is not limited to bent rods but can be applied to any kind of prior shape knowledge that can be expressed in terms of a parametrized spline, as shown in Section 5.3.

Unlike spline parameterization with a deformation matrix [125], which adjusts the spline shape by applying a deformation field to the control points, such a geometry model-based approach provides target object features (such as, in the case of the cells, width, and length) with no need for post-processing to derive the target parameters from the obtained contour. Moreover, having geometry parameters as variables of an objective function enables the application of shape constraints specific to the morphology and size of the target microorganism. In our case, these are biological constraints, such as both upper and lower limits on the width (w) and height ($l_1 + l_2$) of the cells. Considering our rod model, we only expect minor (see Figure 5.2) deviations of target object features from the geometry parameters. As a result, despite the fact that, unlike Hermite splines, where control points must lie on the curve, in the case of B-Splines, control points do not necessarily belong to the contour, the model parameters can be directly considered as target values without the need for postprocessing.

5.1.5. Initial Guess

The resulting minimization problem is non-convex, which can lead to challenges with local (non-global) minima. The initialization may heavily influence the computed solution and the number of iterations required to reach convergence. The initial guess should be chosen carefully to minimize the number of failed segmentations due to undesired local minima and the number of iterations. We suggest considering a straight symmetrical

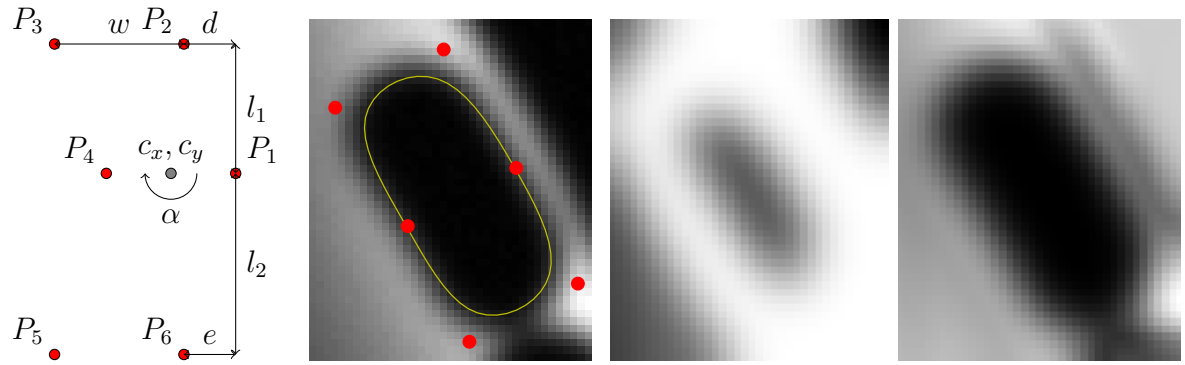


Figure 5.2.: Main components of the analysis, left to right: Geometrical model of *C. glutamicum* with $P_i = (P(\theta)_{i1}, P(\theta)_{i2})$; Input of the objective F: single-cell image tile I (output of the detection with the proposed model-based segmentation as overlay), gradient image ∇I and geodesic distance map D).

($l_1 = l_2$) rod with proper orientation as the initial guess. Orientation of the rod parameters was chosen considering the bounding box proportions: For a rectangular box, i.e., if the length of the box exceeds its with by more than 20%, we use $\alpha = 0°$ or $\alpha = 90°$, depending on which dimension of the bounding box is longer. Otherwise, we use 45° or $-45°$, depending on which angle results in a lower F_{GE}. This means F_{GE} is computed with the default parameters for both of these angles (i.e., no minimization), and the angle with the lower F_{GE} is chosen as the initial value. The rods parameters are: $l_1 = l_2 = 0.5 \cdot$ length of the greater dimension, $w = 17$ pixels, since we have a 0.065 µm pixel size and expect our cells to be around 1 µm wide. Initial guess examples are illustrated in Figure 5.3.

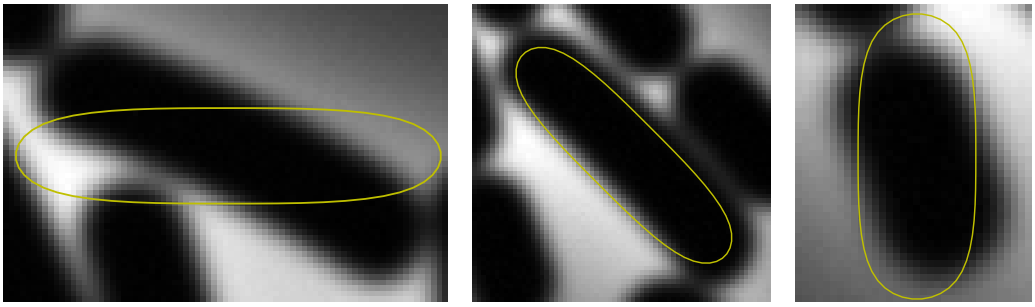


Figure 5.3.: Initial guess examples. Left to right: $\alpha = 0°$, $-45°$ and 90°.

5.1.6. Preprocessing

We applied a set of simple image-processing steps iteratively developed to address and fix specific problems observed during the segmentation method's development, ultimately providing improved segmentation results. To avoid undesired cropping of the parts of the cells, in case YOLO occasionally underestimates the bounding box size, 5 pixels of

padding were evenly applied for each bounding box. To unify the range of pixel intensity values of each tile, the intensity image I in Equation (5.4), the gradient image $|\nabla I|$ in Equation (5.3) and the geodesic distance map D in Equation (5.5), were normalized to the $[0, 1]$ range individually for each tile after postprocessing, i.e, to prevent erroneous convergence to local minima, we applied Gaussian smoothing ($\sigma = 3$ pixels) and clipping to $|\nabla I|$ ($[0, 0.45]$). This also reduces noise and the influence of undesired internal cell structures.

5.1.7. Discrete objective

To facilitate the reproducibility of our results, we provide the discrete representation of the terms of the objective function, where $C = S(P(\theta))$ is a discrete representation of the contour $\mathcal{C}$ for n points per segment. Here, $C \in \mathbb{R}^{2\times 6n}$ represents the coordinates of a closed B-spline curve, modeled by a parameter vector θ. Each point on the curve S_i is connected to its subsequent point S_{i+1}, forming the differential vector dS_i. The magnitude of this vector $|dS_i|$ is used in the calculation of the energy functional, which is then normalized by the length of the curve $|S|$.

Discretizing the continuous contour energy in Equation (5.3) results in:

$$F_{CE}(S) = \frac{1}{|S|} \sum_{i=1}^{6\cdot n} \frac{|dS_i|}{(|\nabla I(S_i)| + \varepsilon)^k} \tag{5.19}$$

where $|S|$ is the length of the discrete curve S, and ∇I is a bilinear interpolation of the image gradient, calculated using central differences and forward or backward differences at the boundaries.

For the region term $F_{\mathrm{RE}}(\theta)$, we use the divergence theorem to convert the area integral into a line integral over the boundary $\mathcal{C}$. The area $|\mathcal{R}(\mathcal{C})|$ of the region $\mathcal{R}(\mathcal{C})$ is computed as the integral of 1 over the region:

$$|\mathcal{R}(\mathcal{C})| = \int_{\mathcal{R}(\mathcal{C})} 1 \, \mathrm{d}x \tag{5.20}$$

Using the divergence theorem, this integral is converted into a line integral:

$$|\mathcal{R}(\mathcal{C})| = \oint_{\mathcal{C}} \mathbf{r} \cdot \mathbf{n} \, \mathrm{d}\mathcal{H}^1(c) \tag{5.21}$$

where $\mathbf{r}$ is the position vector, $\mathbf{n}$ is the outward-pointing unit normal vector, and $\mathrm{d}\mathcal{H}^1(c)$ is the differential arc length along the curve using the Hausdorff measure. The outward-pointing unit normal vector at each point S_i is given by:

$$\nu(S_i) = \frac{[dS_{yi}, -dS_{xi}]}{|dS_i|} \tag{5.22}$$

Thus, the line integral can be approximated by summing over the segments of the curve. The discrete representation of the region energy term is:

$$F_{\mathrm{RE}}(\theta) = \frac{\sum_{i=1}^{6\cdot n} \mathcal{I}_x(S_{xi}, S_{yi}) dS_{yi} - \mathcal{I}_y(S_{xi}, S_{yi}) dS_{xi}}{\sum_{i=1}^{6\cdot n} S_{xi} dS_{yi} - S_{yi} dS_{xi}} \tag{5.23}$$

where $\mathcal{I}_x(x,y) = \sum_{n=0}^{N} I(n,y)$ and $\mathcal{I}_y(x,y) = \sum_{m=0}^{M} I(x,m)$ are the cumulative sums of the image intensities along the horizontal and vertical directions, respectively.

Similarly, for the geodesic distance term $F_{\mathrm{GE}}(\theta)$, we have:

$$F_{\mathrm{GE}}(\theta) = \frac{\sum_{i=1}^{6\cdot n} \mathcal{D}_x(S_{xi}, S_{yi}) dS_{yi} - \mathcal{D}_y(S_{xi}, S_{yi}) dS_{xi}}{\sum_{i=1}^{6\cdot n} S_{xi} dS_{yi} - S_{yi} dS_{xi}} \tag{5.24}$$

where $\mathcal{D}_x(S_{xi}, S_{yi})$ and $\mathcal{D}_y(S_{xi}, S_{yi})$ are cumulative sums of the geodesic distance map D.

5.1.8. Minimization details

As discussed in Section 5.1.4, the organism morphology-dependent constraints can be directly applied to the variables of the objective function. According to [126], the segment lengths l_1, l_2 lie in the range of 0.4 µm to 2 µm, and the overall length $l = l_1 + l_2$ does not exceed the bounding box diagonal. The deviation from a straight rod in terms of d, e was limited to 0.5 µm, i.e., $d, e \in [-0.5\,\mu\mathrm{m}, 0.5\,\mu\mathrm{m}]$. The expected width w for the colony is 0.7 µm to 0.9 µm. Considering the different ranges of the geometrical parameters

$$g = \big(c_x, c_y, l_1, l_2, w, d, e\big) \tag{5.25}$$

and the rotation angle α, the objective function F is minimized alternatingly using the "constrained optimization by linear approximation" (COBYLA) algorithm [127], implemented in [128]. This means, for a given number of iterations, the function $F(\theta) = F(g, \alpha)$ is minimized with respect to the geometrical parameters g for a fixed angle α, and analogously, with respect to the angle for fixed geometrical parameters. The proposed optimization process is illustrated as pseudocode in Algorithm 1.

5.2. Results

We evaluated the results of the proposed method using a validation dataset containing 30 synthetic images, generated by CellSium (Chapter 4), where each image contains between 2 and 196 cells.

5.2.1. Detection results

The performance of the proposed detection step was evaluated using mean Average Precision (mAP) at IoU = 0.5 and the average mAP over IoU = 0.5 : 0.95. The mAP is a measure used to evaluate the accuracy of object detectors, defined as the mean of the average precision (AP) values for different IoU thresholds. Average Precision (AP) is the area under the Precision-Recall (PR) curve, which plots precision (the ratio of true positive detections to the sum of true positives and false positives) against recall (the ratio of true positive detections to the sum of true positives and false negatives) [129]. For the NMS = 0.45 (empirically chosen NMS parameter, which provides the highest mAP score), and Confidence = 0.6, we obtained: Average precision $P = 1$; Average recall

Algorithm 1: The proposed hybrid segmentation method

Data: Dataset, geometrical model, constraints
Result: Cell contours
for *image* $\in$ *dataset* **do**
  // YOLOv5 detection to obtain the bounding boxes with single cells
  $boxes = \text{YOLOv5\_detection}(image)$
  for *box* $\in$ *boxes* **do**
    // Calculation of the initial geometrical parameters (Section 5.1.8)
    $\theta = [g, \alpha] = \text{initial\_parameters}(box)$
    // Alternating minimization of F over g and α
    for $i \in N_{loops}$ **do**
      $\alpha^{(i+1)} = \arg\min_{\alpha} F(g^{(i)}, \alpha)$ using COBYLA
      $g^{(i+1)} = \arg\min_{g} F(g, \alpha^{(i+1)})$ using COBYLA
    end
    $\theta = [g, \alpha]$
    // Generate contour coordinates with θ
    $S(\theta) = P(\theta) \cdot B$
    // Store the result for the current box
    contours.append($S(\theta)$)
  end
end

$R = 0.98$; $\text{mAP}_{0.5} = 0.97$; $\text{mAP}_{0.5:0.95} = 0.76$. These results are on par with similar cell detection studies. For instance, Shakarami et al. reported a mAP0.5 of 0.8986 [130], and Waithe et al. reported mAP0.5 values ranging from 0.816 to 0.979 for different cellular datasets [131].

5.2.2. Segmentation results

We used two metrics to evaluate the segmentation accuracy based on the Dice score (Equation (3.10)) for binary segmentation:

$$\text{Dice}(I^{\text{gt}}, I^{\text{res}}) = \frac{2|I^{\text{gt}} \cap I^{\text{res}}|}{|I^{\text{gt}}| + |I^{\text{res}}|} \tag{5.26}$$

where I^{gt} is the ground trough segmentation and I^{res} is the result obtained by the given segmentation method. Overall, to evaluate the segmentation accuracy, we used two Dice-based scores:

- Foreground Dice (FD) score, i.e. the dice score of a foreground mask. The cell colony is assumed as the union of single cell masks. In this case, the overlaps of cell masks are treated as a region that belong to the foreground once, thereby neglected.

- Average Multi-object Dice (AMD), i.e., the average dice score for each single cell in comparison with a corresponding GT mask.

The accuracy of the proposed constrained geometry-aware method (GA+C) in the Table 5.1 is shown against the non-constrained geometry-aware method (GA) and a conventional (non-parametrized) spline fit (nGA), with the same number of control points. For the latter, the objective function, given by Equation (5.2), was minimized with respect to control points coordinates.

The weights w_R, w_D were chosen using `Hyperopt` [132] with $1 - \mathrm{AMD}(w_R, w_D)$ as objective function, assuming that w_R and w_D are uniformly distributed in $[0, 500]$ with the number of optimization attempts $N = 1000$. The objective function was calculated using frame 15 of the validation set sequence containing 15 cells.

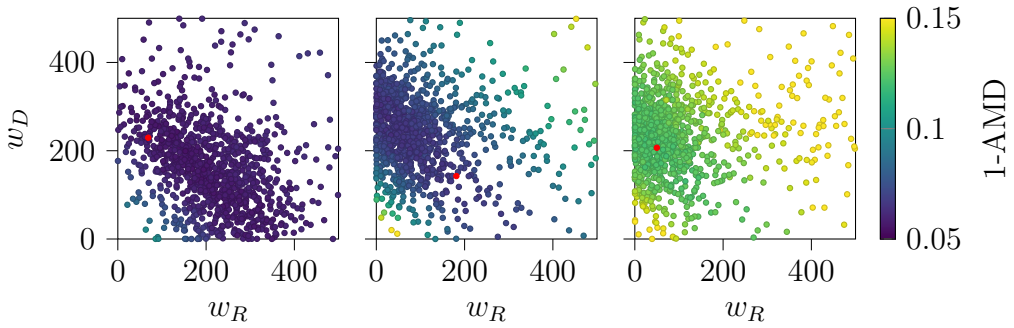


Figure 5.4.: Weight optimization results. Left to right: geometry-aware segmentation with constraints, without constraints, and unconstrained segmentation with control points as variables. The red point shows the best result for each method.

Figure 5.4 illustrates, that the introduction of the geometric model (GA) results in a substantial accuracy gain, compared to conventional (non-parametrized) spline fit (nGA). Despite a non-significant gain in terms of accuracy (cf. Table 5.1) of the constrained geometry-aware segmentation (GA+C) compared to the non-constrained geometry-aware method (GA) method shows visible improvement in terms of robustness, i.e. model insensitivity to the objective function's weights choice. The proposed method can compete with state-of-the-art instance (segmentation of each distinct object) ML-based segmentation methods (e.g. [133]) in terms of Dice score, applied to similar cell segmentation tasks in microscopy. Moreover, the proposed method significantly decreases the manual labor for training data creation (only the detection part has to be trained) since only bounding boxes need to be provided instead of pixel-precise segmentation masks.

Table 5.1.: Segmentation results. FD and AMD scores for all the test dataset

Image (number of cells)	FD			AMD		
	GA+C	GA	nGA	GA+C	GA	nGA
1(2)	0.9471	0.9041	0.8982	0.9471	0.9038	0.8978
5(3)	0.9356	0.9253	0.6719	0.9445	0.9317	0.7385
10(7)	0.9420	0.9391	0.8232	0.9447	0.9405	0.8491
15(15)	0.9451	0.9413	0.8835	0.9460	0.9421	0.8826
20(32)	0.9377	0.9373	0.7036	0.9366	0.9350	0.6375
25(71)	0.9408	0.9386	0.6647	0.9393	0.9355	0.6647
30(196)	0.9329	0.9297	0.8070	0.9297	0.9260	0.8134

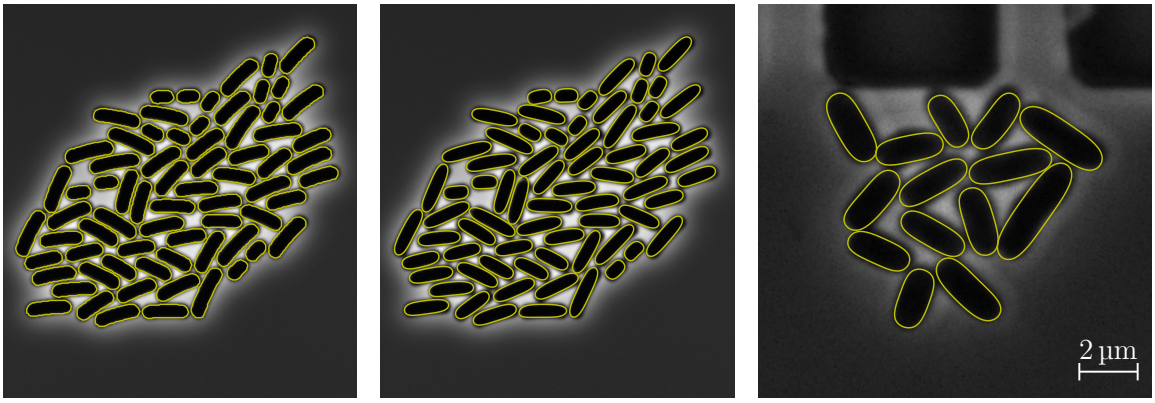


Figure 5.5.: Left to right: Simulated image with GT cell contours; The result of the proposed hybrid segmentation method applied to the synthetic image; The segmentation result of the real *C. glutamicum* data.

5.3. Application of the method to another microorganism with different morphology

As mentioned in Section 5.1.4, the approach is not only limited to rod-shaped cells (Figure 5.5c). There are many organisms with known geometries that can be modeled using simple geometrical models. Another example of such an organism is *S. cerevisiae*. *S. cerevisiae* cells are round to ovoid, 5 µm to 10 µm diameter [97] and can be represented similarly as a parametrized closed spline with 6 control points.

The ovoid-like shape here is proposed to be modeled as an "imperfect" circle of 6 control points, where each of the control points may deviate from the given radius (r) with deviation (Δ). Therefore, the control points are parametrized using eight parameters: radius (r), deviations from "circularity" for each control point (Δ_1, Δ_2, Δ_3, Δ_4, Δ_5, Δ_6), and, analogous to the rod model, center (c_x, c_y), and rotation angle α. Thus, the parameter vector for this ovoid-like model is denoted by:

$$\theta = (c_x, c_y, r, \Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5, \Delta_6, \alpha). \tag{5.27}$$

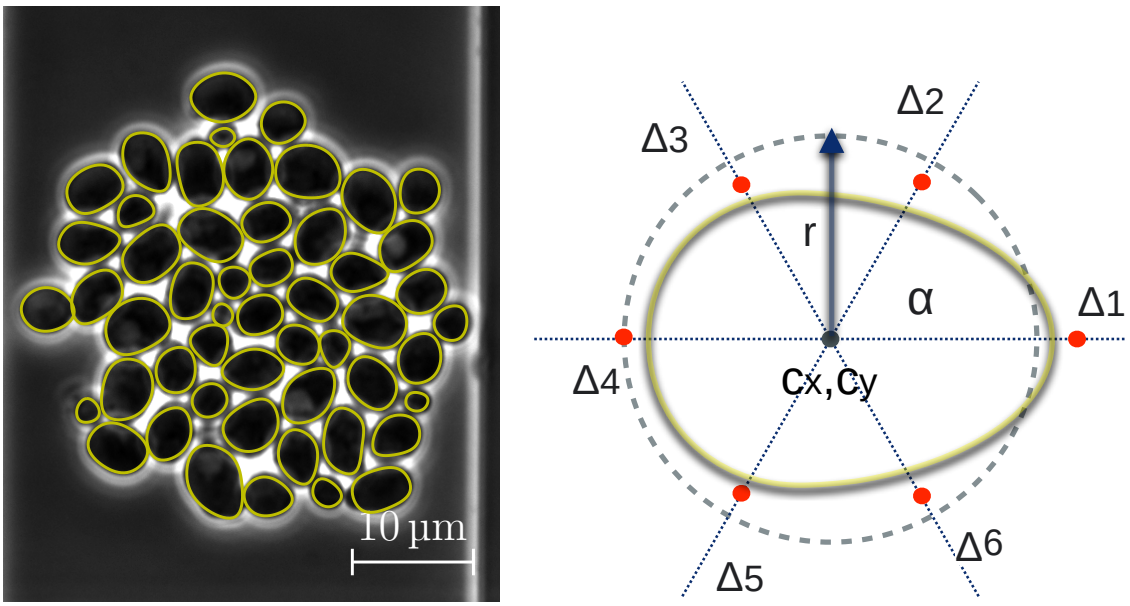


Figure 5.6.: *S. cerevisiae* segmentation (left) based on the ovoid-shape model (right).

To define $P(\theta)$ we use (5.17) and replace (5.18) with

$$\tilde{P}(\theta) = \begin{pmatrix} \tilde{P}_x(\theta) \\ \tilde{P}_y(\theta) \end{pmatrix} = \begin{pmatrix} r_1 \sin\frac{\pi}{3} & r_2 \sin\frac{2\pi}{3} & \dots & r_6 \sin\frac{6\pi}{3} \\ r_1 \cos\frac{\pi}{3} & r_2 \cos\frac{2\pi}{3} & \dots & r_6 \cos\frac{6\pi}{3} \end{pmatrix}, \tag{5.28}$$

where

$$r_i = r + r\Delta_i \tag{5.29}$$

The proposed model enables constraints for the radius value r and degree of "circularity" by constraining the absolute value of each Δ to be in a certain range.

The result of the geometry-aware segmentation of *S. cerevisiae* and the proposed model are illustrated in Figure 5.6. For this demonstration, the bounding boxes were provided manually. The constraints were set as $r \in [1\,\mu\text{m}, 7\,\mu\text{m}]$ and $\Delta_i \in [-0.3, 0.3]$.

5.4. Conclusions

In this chapter, we use the biotechnologically relevant microorganism *C. glutamicum*, described in Section 1.3.1, as a model organism to offer another solution to the notorious lack of training and validation data for microbial single-cell analysis. Specifically, our contribution to single-cell image analysis is developing a hybrid multi-object segmentation approach. Our approach is examined for the segmentation of *C. glutamicum* and summarized as follows:

First, the YOLO detection framework is used to detect the cells of interest in the image. Second, each detected cell is segmented individually using a variational-based segmentation approach that incorporates prior knowledge of the geometry of the cell

shape. The contour of the segmentation is modeled as a closed uniform cubic B-spline, whose control points are parametrized using the known cell geometry. This approach enables the segmentation to be performed on a per-cell basis, reducing the required training data compared to purely ML-based segmentation methods that require accurate segmentation maps.

The main advantages of the proposed approach are the reduced manual labor to create training datasets and the preservation of geometric features for each segmentation instance using the available knowledge of cell morphology. Here, the segmentation contour is modeled as a closed uniform cubic B-spline, whose control points are parametrized using the known cell geometry. Our method requires only bounding boxes, in contrast to purely ML-based segmentation approaches, which need accurate segmentation maps as training data that are very laborious to produce. Nevertheless, the proposed method performs on par with ML-based segmentation approaches usually used in this context.

6. Automated characterization of catalytically active inclusion body production in biotechnological screening systems

This chapter is based upon the publication "Automated characterization of catalytically active inclusion body production in biotechnological screening systems" by **K. Ruzaeva**, K. Küsters, W. Wiechert, B. Berkels, M. Oldiges, K. Nöh. The author of this thesis developed the segmentation method described therein, evaluated the data, and prepared the manuscript.

6.1. Automated screening and analysis of CatIB production

Inclusion bodies are misfolded proteins, typically regarded as cellular waste products, that occur during overproduction in bacteria such as *E. coli*. However, if these inclusion bodies are catalytically active (CatIBs), they become interesting targets for synthetic chemistry and industrial biotechnology, being easy to produce and harvest [134, 135]. Bioprocess conditions determine the efficiency of CatIB production. Although Catalytically active Inclusion Body (CatIB) production is compelling, little is known about efficient process conditions. Nowadays, microbioreactors, such as the so-called Biolector system, are used to screen for beneficial (cost-efficient) process conditions [136]. For automation, the Biolector is integrated with a liquid handling system, enabling the injection of samples into a flow chamber, which is fixed under a microscope. With an integrated camera, a series of images is taken of the samples.

CatIBs are observed on microscopy images as bright spots within the cells. Therefore, for condition screening, automated microscopy is used to characterize the production of CatIBs. The number of CatIBs per cell and the time point of the sample are thus detectable analysis parameters that are then correlated with the amount of protein produced. Being equipped with automated sampling and microscopy, this setup allows large-scale optical condition screening to characterize the production of CatIBs.

Due to the large amount of data generated per experiment (500 images per sample, and three samples per time point), manual analysis of CatIB counts is infeasible. The application of off-the-shelf machine learning (ML)-based segmentation approaches is

challenging because of the lack of training (GT) data, which is compounded by the difficulty in producing GT data. Since the height of the flow chamber is far larger than the cells, a large fraction of cells is out-of-focus (Figure 6.1), making manual annotation extremely difficult and risking the introduction of bias. To objectively analyze the formation of CatIBs in large-scale screens, we propose a Hybrid Image Processing Pipeline (HIPP), designed to segment in-focus cells and determine CatIB parameters. With this, analyzing CatIB production at cell-to-cell heterogeneity becomes possible, such as the number and the ratio of the cells carrying CatIBs, along with the average numbers, such as total CatIB and cell areas, to monitor their state in the cultivation process. We tested our HIPP with a real data set, showing biotechnologically comprehensible results.

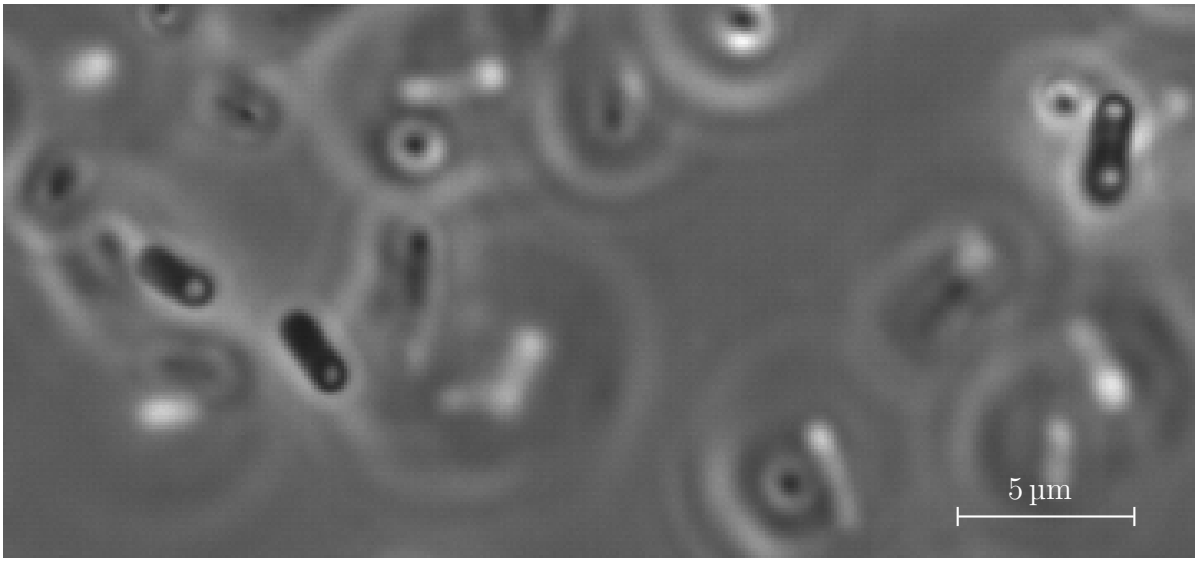


Figure 6.1.: Magnified part of the typical microscopy image showing three in-focus cells (dark "rods") with CatIBs (cells left to right: one CatIB, one CatIB, two CatIBs) and numerous out-of-focus cells.

6.2. Experimental set-up

E. coli DH5 was used as a cloning host for the generation of plasmid pET28a::*Ec*LDCc::SG::TDoT. DH5 is a strain of *E. coli* commonly used in molecular biology for cloning due to its high transformation efficiency and genetic stability. The plasmid pET28a is a widely used vector for protein expression in E. coli, and here it is fused with the gene *Ec*LDCc::SG::TDoT for an experimental purpose.

Heterologous expression of the gene fusions was performed using *E. coli* BL21(DE3). This strain of E. coli is frequently used for protein expression because it carries the DE3 lysogen, which contains the T7 RNA polymerase gene under the control of the lacUV5 promoter, allowing for high-level expression of target genes cloned into T7 promoter-based vectors such as pET28a. A lysogeny broth (LB) medium was employed to cultivate strains during the cloning procedure and for the expression strains' precultures. LB medium

is a nutrient-rich media commonly used for the growth of bacteria. For the expression, autoinduction (AI) medium consisting of 2.2 g/L KH_2PO_4, 9.4 g/L K_2HPO_4 and 5 g/L glycerol at pH 7.2) supplemented with 0.5 g/L glucose and 2 g/L lactose was used.

After precultivation, expression cultures were inoculated at OD600, and the cells were cultivated at 37°C for 3 hours at 1000 rpm. Then, the temperature was decreased to 15°C, and the expression ensued for 69 hours under the same shaking conditions. A filling volume of 800 µL was used for all wells. All cultures were supplemented with 50 µg/ml kanamycin for plasmid maintenance. CatIB strains were cultivated in a BioLector, which was integrated into a liquid handling system (Freedom Evo200, Tecan, Männedorf, Switzerland). Every 4.5 hours, cells were harvested from each well of the FlowerPlate (the last three wells were harvested at 72 hours) and stored in a deep well plate at 4°C on the robotic deck until automated microscopy was performed. For sample injection, a self-built injection station [21] was used. The station was connected via a tubing (VWR, Darmstadt, Germany) to a flow chamber (height: 20 µm, length: 58.5 mm, width: 800 µm, microfluidic ChipShop, Jena, Germany). The chamber was fixed on an inverted Nikon Eclipse Ti microscope (Nikon GmbH, Düsseldorf, Germany) equipped with a CFI Plan Apo Lambda 100x Oil objective (Nikon GmbH, Düsseldorf, Germany). 300 µL of each sample, followed by 400 µL H_2O were injected in the injection station. After the injection of 600 µL the flow was set to zero to allow for image acquisition. A 1 µL pulse with a velocity of 1 µL/switch after a 4 min delay was performed three times to flush in new cells. 500 images (one image per second) were taken of each sample in the flow chamber, resulting in 1500 images for each time point. Images were taken with a Thorlab camera DCC154M-GL (Thorlabs Inc., Newton, New Jersey, USA).

6.3. Hybrid image processing pipeline

The goal of the proposed HIPP is to segment the in-focus cells and their CatIBs (if present) separately and, afterward, classify the segmented cells based on the number of the CatIBs. To accomplish this goal, considering the large fraction of free-flowing out-of-focus cells, we split multi-object (in-focus cells) and multi-label segmentation (number of CatIBs in a cell) into two major steps: an ML-based single-class detection approach and a model-based segmentation step, similar to the approach discussed in the previous chapter. This approach includes variational-based and thresholding-based segmentation. The introduced separation has four reasons:

(i) Despite the availability of the multi-object segmentation methods, like [63], as already mentioned in the Chapter 5 their main limitation is the lack of training data, which is very laborious to produce. Here, unlike precise pixel-wise image annotation necessary for the training of a multi-object pipeline, we use an easy-to-create single-class training dataset since only bounding boxes are needed to train the detection framework.

(ii) Although the detection framework we use supports multi-class detection and could be used as a classifier for the different types of cells (none, 1, or 2 CatIBs per cell), it still requires a larger and more complex multi-class training dataset.

(iii) As stated in Chapter 5, another important benefit of the separation of detection and the segmentation step is simplifying the problem by reformulating multi-object segmentation as many single-object segmentation tasks. This unlocks single-object model-based segmentation methods, such as variational-based segmentation and thresholding, which are otherwise infeasible due to the huge number of object instances to be segmented per image or region of interest.

(iv) One more advantage of the splitting is that the segmentation can be done in parallel for the cells since each detected bounding box contains only one cell, which can be processed independently. This advantage was also highlighted in the previous chapter.

6.3.1. Detection

Analogously to Chapter 5, we use the YOLOv5 detection framework to detect the in-focus cells. The training data was generated by manual annotation of four images, where only in-focus cells were labeled, resulting in 151 objects (cells). To artificially increase the training dataset YOLOv5's built-in augmentation was used, which is image saturation, hue and value augmentation; rotation, translation and scaling; horizontal and vertical flips and image mosaic augmentation. Image mosaic augmentation involves combining four different training images into one, allowing the model to learn from varied contexts within a single image [137].

The detection was performed on 15 manually chosen images with a steady flow for each of the 16 time points. These images were selected based on criteria such as image quality, focus, and the presence of a steady flow of cells to ensure consistent analysis. The resulting output is discussed in Section 6.4.

Figure 6.2 shows detection examples for images from the beginning (4.5 hours, yellow), middle (36 hours, green), and the end (72 hours, red) of the dataset and the number of detected cells on 15 images for each time point. In the bar chart, each number of cell bar corresponds to the sum of the cells detected in 15 images from the corresponding time point. The images on top are one example of the 15 images used to generate the colored bar in the plot below. In the bar chart, the positive slope in the first part of the experiment (4.5-36 hours) coincides with the natural growth of the cell colonies. The decline of the number of detected cells during the last part (60.5-72 hours) is explained by the fact that during the last sampling points, *E. coli* cells were too crowded for exact automated image analysis. Especially in the last time point (Figure 6.2, top right), nearly all in-focus cells overlapped with out-of-focus cells, hampering the automated analysis. In the future, to improve the automated microscopy system, a dilution step of the cells should be added at a specific *E. coli* density to improve correct image analysis over the whole cultivation time. Moreover, the detection accuracy can likely be improved by using a bigger training dataset since we use an ML-based detection.

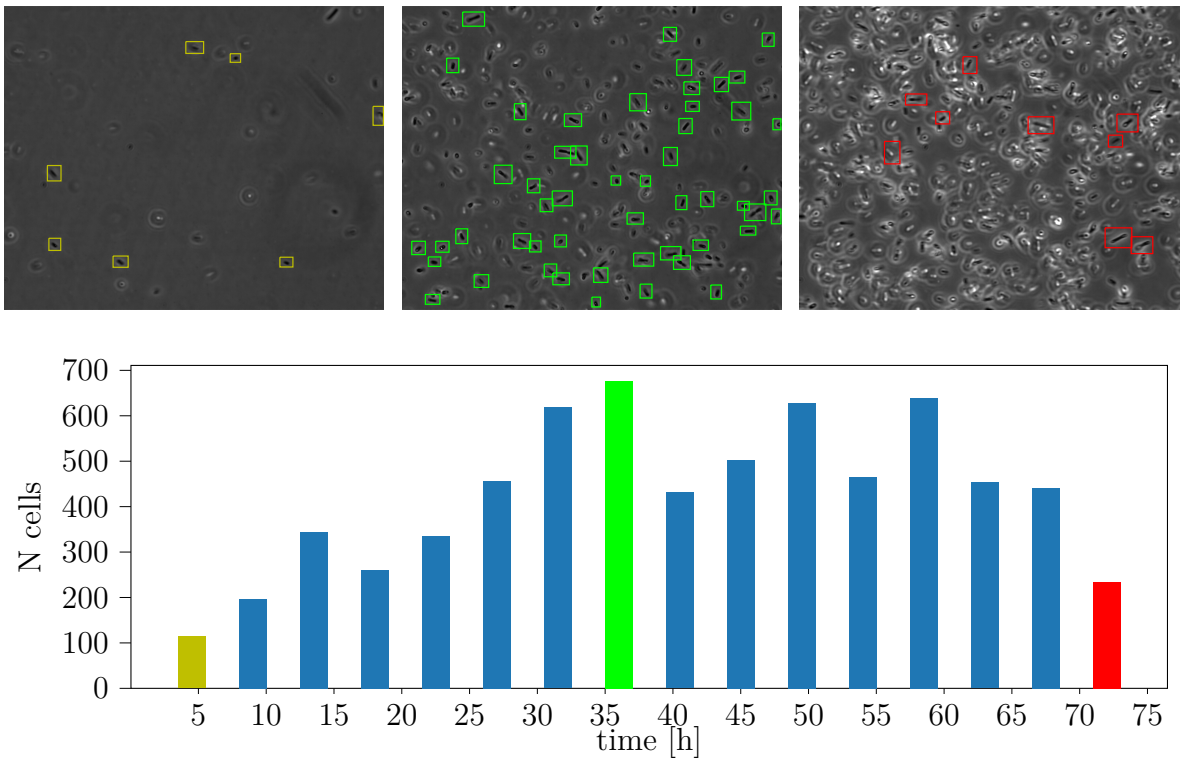


Figure 6.2.: Number of detected cells at each time point (bottom) and detection examples for images from different time points (top, marked with the corresponding colors). Each number of cell bar corresponds to the sum of the cells detected in 15 images from the corresponding time point. The three images on top each represent one example from the corresponding 15 images used to generate the colored bar below.

6.3.2. Segmentation

After the detection, each detected cell is segmented individually by considering the image tile inside the detected bounding box. The segmentation is divided into two steps: first, cell-body segmentation (no CatIBs are assumed to be present as we alter the data to remove inclusions), and next CatIB(s) segmentation in each cell. We use a model-based segmentation approach on each single-cell image tile. Direct cell body segmentation is complicated by the fact that CatIBs are as bright as the background, or even brighter, and in most cases are missegmented as background. To avoid this problem, we first apply several pre-processing steps.

Initially, we “fill” the CatIBs using a morphological closing operation, which combines the application of dilation (minimum filter, as the cell is darker than the background) and erosion (maximum filter) to the dilated image tile. We use a disk-shaped structuring element with a radius of 5 pixels for the morphological operations. Then, to smooth the artifacts resulting from the closing operation and to avoid local minima problems during the following variational segmentation step, Gaussian blur is applied to the obtained

"closed" (inclusion-less) image tile.

Next, the segmentation process is divided into two steps: cell-body segmentation and CatIB segmentation in each cell. For cell-body segmentation, we use a variational B-splines-based method where the cell shape is modeled as a straight rod by fixing the curvature parameters d and e to zero, as described in Chapter 5. The spline consists of six control points. As the shape of the cell is more flexible than the shape model used in the variational segmentation step, to obtain the fine cell contour, global manually set thresholding (the same threshold value was applied to the entire dataset) is applied to the pixels which belong to the interior of the spline. This results in a binary pixel mask of the cell body. The suggested spline fit is necessary if parts of other cells are in the bounding box but not necessary if sparse cell flow can be guaranteed (i.e., with dilution).

To acquire the mask for the CatIB(s) of the target cell, thresholding is also applied to the original image tile. The resulting CatIB mask is multiplied by the cell body mask obtained in the previous step.

Since every cell has a bright surrounding "halo", which can be missegmented as CatIB, every CatIB mask instance is checked for its area and its eccentricity. Eccentricity is defined as the ratio of the distance between the foci of the ellipse and its major axis length. The CatIB mask components with high eccentricity (more than 80%) and small area (2 or fewer pixels) are assumed to be outliers and removed from the mask.

The above-described pipeline for cell body and CatIB segmentation, which takes single-cell image tiles obtained at the detection step as input, is represented in Figure 6.3. The results of the segmentation are shown in Figure 6.4.

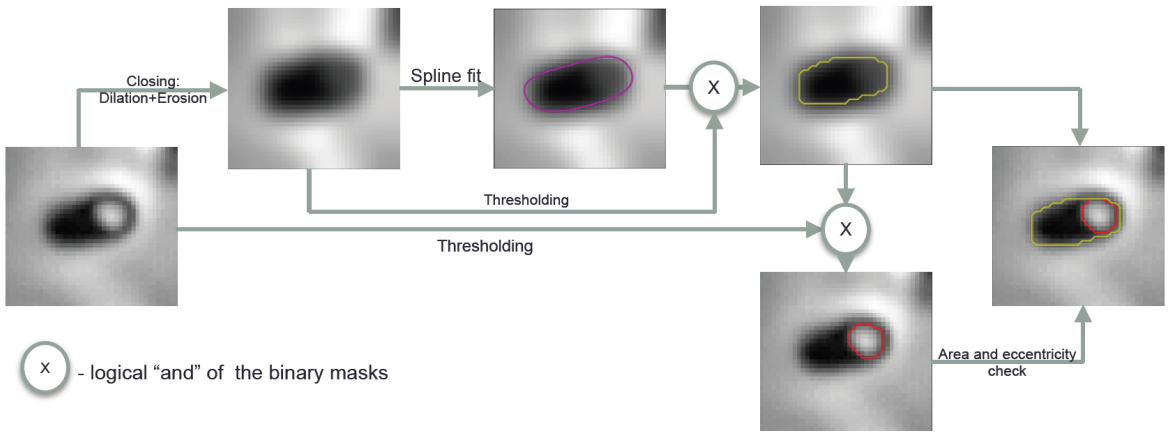


Figure 6.3.: The proposed segmentation pipeline.

6.4. Results

6.4.1. Accuracy quantification

Since no benchmark datasets are available for our application setting, we created GT data ourselves. Therefore, the validation dataset is rather small. As a validation dataset

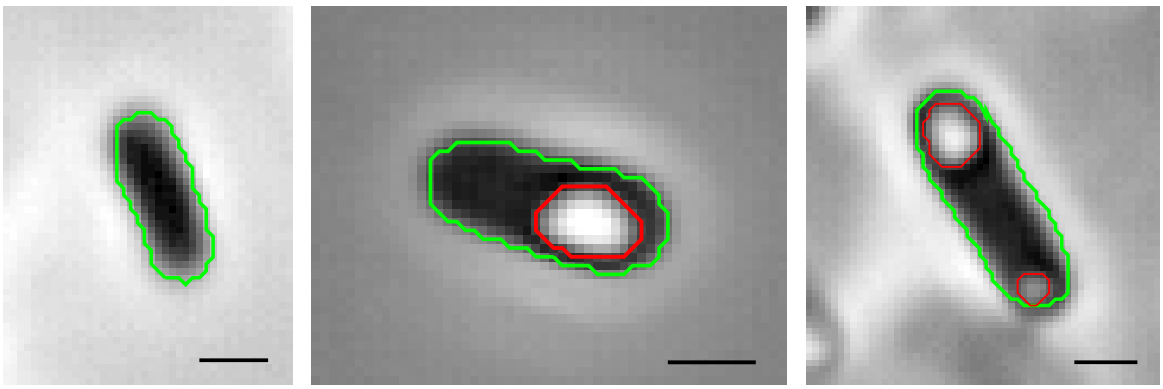


Figure 6.4.: Example segmentation results for the three single-cell image tiles, selected from the results, which illustrate different cell classes (zero, one, and two CatIBs) from the validation dataset. The green contour represents the cell body, and the red contour – the CatIBs (if present). The scale bar indicates a length of 1 µm.

for the segmentation, 30 single-cell image tiles obtained at the detection step were used, where an expert performed manual pixel-wise segmentation annotation. The dataset contains ten image tiles of each class from different time points, where for the "one CatIB" and the "two CatIBs" classes, the GT CatIB masks are provided as well.

As a quality metric for the proposed segmentation algorithm, we used the average Dice score (Equation (3.10)) for the cell and for the CatIBs:

$$\mathrm{Dice}((\hat{I}_i, I_i)_{i=1}^{N}) = \frac{1}{N}\sum_{i=1}^{N}\frac{2|\hat{I}_i \cap I_i|}{|\hat{I}_i| + |I_i|} \tag{6.1}$$

Here, I_i is the binary mask (either cell body or CatIB(s)), obtained with the proposed pipeline; $\hat{I}_i$ is the GT binary mask of the respective object, from the manual annotation by the expert and $N = 30$ is the number of images tiles in the validation dataset.

The resulting quality metrics for the empirically tuned parameter set, which provides the best scores, are $\mathrm{Dice}_{\mathrm{cell}} = 0.88$ and $\mathrm{Dice}_{\mathrm{CatIB}} = 0.78$. The parameter set includes the cell body and the CatIB thresholds; sizes of the masks for the dilation and erosion; and the weights for the variational spline-based segmentation step [138]. The specific parameter values used in the experiment are as follows: `threshold=0.2`, `dilation_mask=7x7`, `erosion_mask=5x5`, `sigma_of_gaussian=2` (for the spline fit); criteria for the detected IBs: `eccentricity=0.9`, `area=2`.

The relatively low $\mathrm{Dice}_{\mathrm{CatIB}}$ score can be explained by the fact that CatIBs generally have an area of around 3-10 pixels. Even the slightest deviation (1 pixel) of the result from the GT significantly affects the score. For example, if a CatIB has a GT area of 3 pixels and the segmentation result covers 2 of those pixels (with no false positives), the Dice score would be $2 * 2/(2 + 3) = 0.8$. The classification of the cells was performed based on the segmentation result, where cells with no CatIBs segmented belong to the

first class, cells with one solid CatIB (after the area and eccentricity checks) belong to the second class, and cells with two separate (not 8-connected) CatIBs detected belong to the last class. The confusion matrix is shown in Table 6.1. It is notable that the performance for detecting cells with two CatIBs is quite low, however, for practical purposes, distinguishing between cells with no CatIBs and those with one or more CatIBs is effective.

Table 6.1.: Confusion matrix for segmentation-based cells classification (3 classes: 0, 1, 2 CatIB(s) per cell)

		Predicted		
		No CatIB	1 CatIB	2 CatIBs
Actual	No CatIB	**10**	0	0
	1 CatIB	1	**9**	0
	2 CatIBs	1	4	**5**

6.4.2. CatIB characterization

To evaluate the automatic analysis, we selected and manually counted the cells with CatIB(s) and the total number of cells for each time point on five images. These five images were chosen from the 15 images selected earlier for automatic analysis. This manual annotation provided GT data, which was then compared against the results obtained with the proposed HIPP on all 15 images for each time point. This means that one-third of the images used in the automatic analysis were manually analyzed, ensuring a substantial validation set for comparison.

In Figure 6.5, in the second half of the experiment, an underestimation of cells with CatIBs is observed, which correlates with the classification results (Table 6.1). Nevertheless, the obtained automatic result qualitatively matches the manual result with PCC = 98 % (calculated using Equation (4.1)) and is useful for experts to adjust the experimental design, conditions control, etc.

The proposed automated HIPP, applied to the 15 images for each time point, was used to observe the CatIB formation process over the whole cultivation time along with the cell growth (Figure 6.6).

The significant growth of the cells with CatIBs is observed after 31.5 hours. The number of in-focus cells with one CatIB reached its peak at about 50%. By the end of the experiment, the formation of the second CatIB and the growth of the CatIB area were still ongoing. A linear trend in cell area growth was noted, with the cell area reaching its maximum (3 μm^2) by the end of the experiment. The growth of the CatIB area as a fraction of the cell area exhibited exponential behavior, reaching nearly 20%. A small peak observed at 18 hours is most likely an error and should be disregarded.

Overall, the resulting numbers provide the necessary information for CatIB production characterization, potentially unlocking control of fermentation processes based on

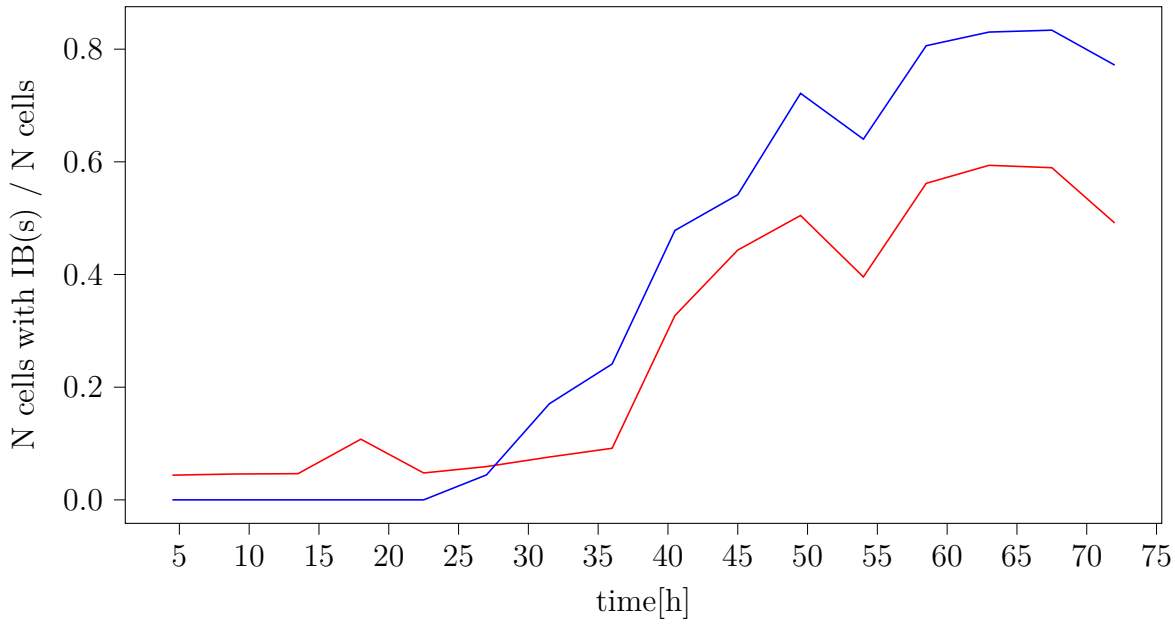


Figure 6.5.: The fraction of the CatIB-carrier cells over time during the cultivation experiment, obtained by manual counting (blue) and the proposed HIPP (red).

morphological features in real-time.

6.5. Summary and impact

This chapter presents an automated pipeline for the microscopy image-based characterization of CatIBs, incorporating a fully automatic high-throughput experimental workflow alongside a hybrid approach for multi-object microbial cell segmentation, leveraging many elements from the previous chapter.

To investigate the heterogeneity of CatIB development during cultivation and to monitor the size and quantity of CatIBs over time, a hybrid image processing pipeline was developed. This approach integrates an ML-based detection of in-focus cells with model-based segmentation.

The proposed method saves approximately 8 hours of manual microscopy per experiment, as highlighted in the results section. It was employed to explore the heterogeneity of CatIB development during cultivation and to track the size and quantity of CatIBs over time.

Due to the flexibility in selecting different segmentation parameters and the adequately accurate results despite low-quality images, we anticipate that a similar strategy could be effective for comparable image segmentation tasks in microbiology and medicine. This approach is particularly relevant in scenarios where there is insufficient training data to perform a fully data-driven segmentation approach, provided that the segmentation task

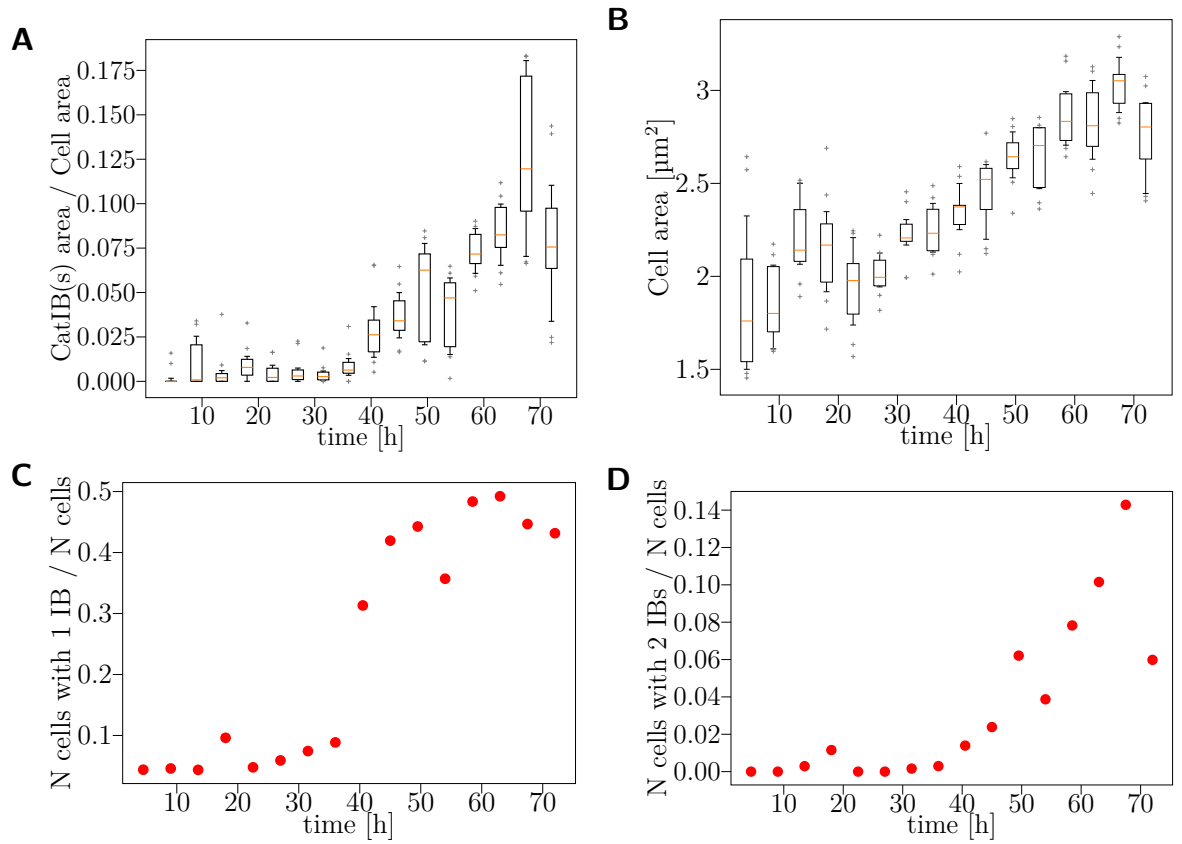


Figure 6.6.: CatIB development over time: (**A**) CatIB area fraction and (**B**) Average cell area, with the median (orange), the 25/75 % quartiles (black box) with 10 %/ 90 % whiskers and the outliers (gray crosses); (**C**) Fraction of the cells with one CatIB; (**D**) Fraction of cells with two CatIBs. The deviation of the values for the last time point from the overall trends is explained by artifacts due to high cell densities(which can be tackled with a dilution step before reaching the critical regime).

can be modeled appropriately. This strategy holds the potential to save significant time and resources.

Part IV.

Tracking

7. Cell tracking using an activity-prioritized assignment strategy

This chapter is based upon the publication "Cell tracking for live-cell microscopy using an activity-prioritized assignment strategy" by **K. Ruzaeva**, J.-C. Cohrs, K. Kasahara, D Kohlheyer, K. Nöh and B. Berkels. The author of this thesis developed the method described therein, evaluated the data, and prepared the manuscript.

Cell tracking is essential to determine single-cell features, such as elongation rates. Unlike the standard object tracking task, in live-cell experiments, cells grow, move, and divide over time toward cell colonies that are densely packed in confined microfluidic growth structures. With increasing cell numbers, following the cell-cell associations correctly over image sequences becomes increasingly challenging due to the increasing number of possible associations.

In the literature, cell tracking methods are split into unsupervised methods, such as tracking by model evolution and tracking by detection, and supervised approaches, usually ML-based.

In tracking by model evolution approaches, an initial segmentation is propagated over time, meaning that the result from the previous frame becomes an initialization for the next one, thereby performing simultaneous segmentation and tracking. These methods require a considerable overlap between the initialized contour and an object to be segmented, which may not be true in low-frame rate datasets. Besides, taking into account the variational nature of these methods, they may be computationally expensive for large cell colonies [139, 140, 141], making the algorithms slow compared to tracking by detection algorithms.

Tracking by detection methods splits the task into detection, often in the form of instance segmentation and tracking. The segmentation can be performed with a variety of available cell segmentation methods [142, 143, 138]. These tracking methods link the segmented cells between consecutive frames based on their similarities, i.e., by finding correspondences between cell features in successive frames. Here, cell association becomes complicated when the feature similarity of a cell to its within-frame neighbors is comparable to the similarity of the same cell in consecutive frames. Most traditional cell tracking association strategies are performed by comparing the feature vectors, where the dominant feature typically is the cell location, i.e., so-called nearest neighbor tracking

[144]. In addition to the cell location, other features, e.g., spectral features of single-cell image crops [145], are utilized.

Tracking by detection is performed by linking segmentation masks over time based on a loss, i.e., the "linking" measure. A simple loss is the Euclidean distance between the positions of the cell centroids. To prevent "impossible" long-distance associations, a maximal distance limit is often applied to the distances [146]. Other linking measures are based on manually engineered features, such as morphology and features of the cell's neighborhood [147, 148, 149]. The loss, consisting of several terms, requires the proper, manually tuned weighting of the components. That makes it hard for the algorithm to generalize across microorganisms featuring different growth behaviors. Besides, the cell daughters may lose the mother's features, such as the orientation angle, after the division, making the assignment problems less trivial.

As an alternative to the feature-vector comparison, some methods utilize cross-correlation [150]. The cross-correlation-based linking shows promising results for cells with more complex morphology, but the approach is not optimized for our case, where cells look very much alike.

The Uncertainty-Aware Tracking from [78], which relies on a Bayesian approach and keeps track of all possible lineages, ranking them by their probability, shows promising results for low-frame rate datasets. However, the approach is computationally expensive, extremely space demanding, and thus unsuitable for analyzing big cultivation datasets, like [151].

Unsupervised cell tracking approaches are still dominant in the cell tracking field. While supervised ML-based cell tracking approaches exist, e.g., [152, 153], these methods require training data, and considering every microfluidic experiment's uniqueness (different microorganisms, microscopy setup, etc.), training and benchmark datasets are often missing, and manual tracking annotations are especially laborious to produce.

To address the challenges mentioned before, this chapter introduces advanced cell tracking (Figure 7.1), which includes multi-object segmentation, activity map generation, and the two-step assignment strategy. The assignment of growing cells starts with the cell with the lowest activity. This strategy enhances the accuracy of tracking by prioritizing cell assignments based on their activity levels, thereby addressing the limitations of conventional tracking approaches.

7.1. Experimental Setup

To assess the method's performance, the tool is tested with an in-house generated test dataset of two prominent microbacteria *E. coli* and *C. glutamicum*.

E. coli In our numerical experiments, the wild-type strain *E. coli* MG1655 was used. The strain was cultured in a microfluidic device for microbial single-cell analysis, coupled with time-lapse microscopy. Experiments were conducted using an inverted time-lapse live cell microscope equipped with a 100×oil immersion objective and a temperature incubator. Phase contrast images of the growing microcolonies were captured every five

minutes. For further cultivation details, we refer to [16]. As noted in the Introduction (Section 1.3.2), *E. coli* divides symmetrically through binary fission, where each cell splits into two daughter cells. Although the daughter cells are generally similar, they are not always identical in every aspect due to potential variations in cell contents and slight asymmetries during division. Importantly, E. coli does not exhibit the snapping division behavior observed in some other bacteria, which simplifies the tracking task.

C. glutamicum As mentioned in the Introduction (Section 1.3), in contrast to the cell division of *E. coli*, *C. glutamicum* shows apical growth, i.e., a cell grows from its poles, and a division into two unequal daughters. In addition, the peculiar dynamic V-snapping of cells after division (Fig. 1.3.1) particularly complicates the tracking task for *C. glutamicum*, especially when it comes to finding correspondences between consecutive frames when the majority of cells in a population are dividing in a V-snapping manner at the same time, pushing the cells in their neighborhood. The wild-type strain *C. glutamicum* ATCC 13032 was used in our numerical experiments. The imaging parameters of the experiment are similar to the *E. coli* cultivation experiment. For further cultivation details, we refer to [154]. Unlike the *E. coli* experiment, phase contrast images here were captured every two minutes.

7.2. Activity-based tracking

7.2.1. Segmentation and preprocessing

As a segmentation framework, we use the recent U-net-based tool Omnipose [142], which was published shortly after our spline-based segmentation work (Chapter 5, [138]). The Omipose network was trained on similarly looking images of bacteria under phase contrast and addressed our segmentation problem well. We used the default weights and the `bact_omni` model type provided by the authors of [142]. We did not observe any false negative segmentation instances. The few false positive segmentation instances were filtered out by setting a minimal cell area limit known and specific to the microorganism ($0.3\,\mu m^2$ to $3\,\mu m^2$).

7.2.2. Activity map

We propose a tracking-free measure for cell activity that we call Activity Map (AM), highlighting the cells that are likely to experience translation, division, or tilting.

The activity of cells is typically derived from tracking results. However, the approximate estimation of the cell activity, as will be shown below, is possible without tracking information and may even enhance the tracking. The proposed measure is based on the intensity differences of consecutive frames combined with the cells' segmentation masks obtained in the segmentation step. The calculation of the intensity difference of two consecutive frames was utilized by [149] as a preliminary step for the cell tracking to

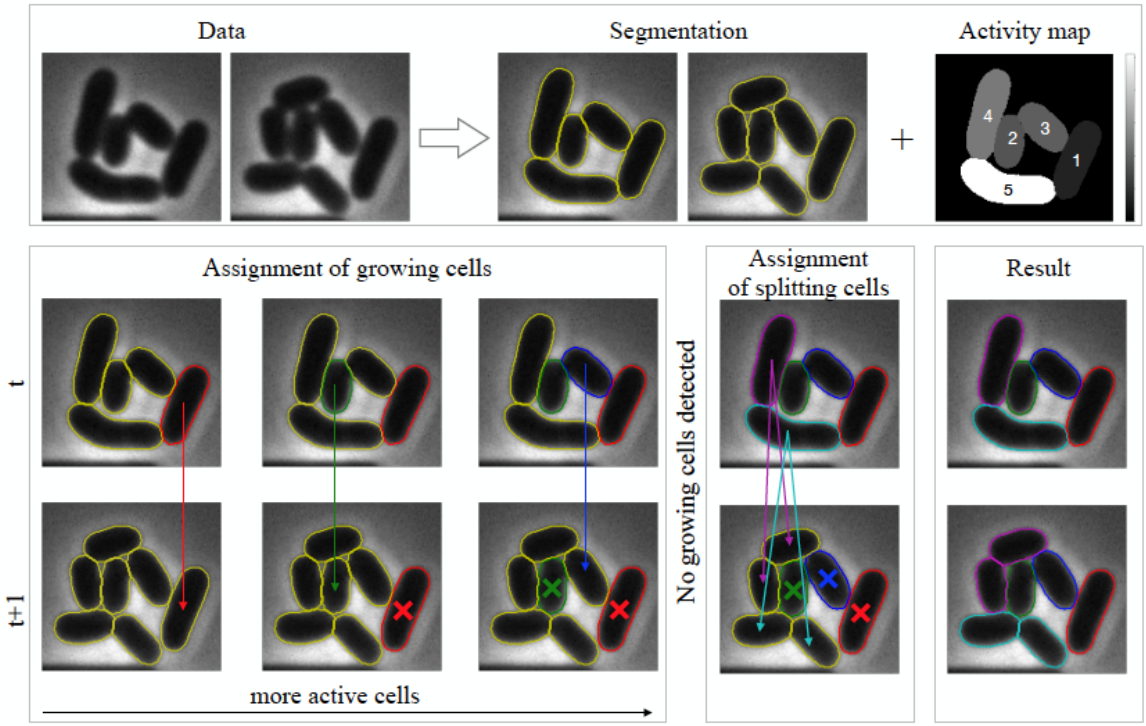


Figure 7.1.: The proposed tracking pipeline, which includes multi-object segmentation and activity map generation (top) and the two-step assignment strategy (bottom), for cells from two consecutive frames. The assignment of growing cells starts with the cell with the lowest activity. Once all growing cells are assigned, the remaining cells from time t and time $t+1$ participate in the combinatorial assignment.

estimate the cells' motion vectors, but limited to a high frame rate (more than one frame per five minutes) datasets.

To compute the proposed AM, we first calculate the moving pixel-wise standard deviation (S) over a stack of frames, defined as:

$$S_t(x,y) = \sqrt{\frac{1}{n_t} \sum_{s=\max(t-n_-,1)}^{\min(t+n_+,N)} \left(I_s(x,y) - \overline{I_{t-n_-}(x,y), \ldots, I_{t+n_+}(x,y)} \right)^2}, \tag{7.1}$$

where $t \in \{1, \ldots, N\}$ represents the frame number in the image stack, and $n_-, n_+ \in \mathbb{N}$ are user-defined parameters specifying the window size, $n_t = \min(t + n_+, N) - \max(t - n_-, 1) + 1$ the number of frames in the window. If $t < n_- + 1$, the summation is adjusted to start from the first frame, and if $t > N - n_+$, the summation ends at the last frame N. I_s for $s \in \{1, \ldots, N\}$ is the s-th image (frame) in the image stack, N is the number of frames, and $\overline{I_{t-n_-}, ..., I_{t+n_+}}$ denotes the pixel-wise average of the images in the specified window. For activity visualization purposes, we suggest taking the symmetric window ($n_- = n_+$).

To calculate $a_{t,i}$, the activity of i-th cell at time t, we integrate the obtained map S_t over the corresponding segmentation mask. To reduce the influence of the cell area, i.e., bigger cells would have more activity, the value was normalized by dividing by the area, i.e.,

$$a_{t,i} = \frac{\sum_{x=0,y=0}^{X-1,Y-1} M_{t,i}(x,y) S_t(x,y)}{\sum_{x=0,y=0}^{X-1,Y-1} M_{t,i}(x,y)} \tag{7.2}$$

where X and Y are the width and height of the image, respectively, $M_{t,i}$ is the binary segmentation mask for the i-th segmentation instance of the t-th frame ($t \in \{1, \ldots, N\}$). Note, that the index i refers to the specific cell within the frame t, and the number of cells, and hence the value of i, can vary from frame to frame. As a prioritization and activity metric for cell tracking, we consider only two ($n_- = 0, n_+ = 1$) consecutive frames. The proposed AM is illustrated in Fig. 7.2, where each cell's segmentation mask was "colored" with the respective activity value.

The proposed map may be beneficial during manual tracking annotation to attract the user's attention to the active (splitting, growing, or migrating) cells. Moreover, it allows automatic and straightforward "nearest neighbor" assignments for those cells that do not show any activity.

One additional use of the AM, which we do not explore here, is as an indicator for refining under-segmentation in "non-active" regions by incorporating the "missed" non-active cell mask from the previous or next frame.

Similarly, over-segmentation, such as mistakenly segmenting parts of the chip as cells, can be addressed by identifying objects that remain stationary throughout the entire cultivation period, as they are unlikely to be actual cells.

Additionally, the suggested mapping may help during cultivation in real time to control the environment and indicate starving cell sub-colonies (if no activity is present in the region) due to the lack of nutrition or other reasons.

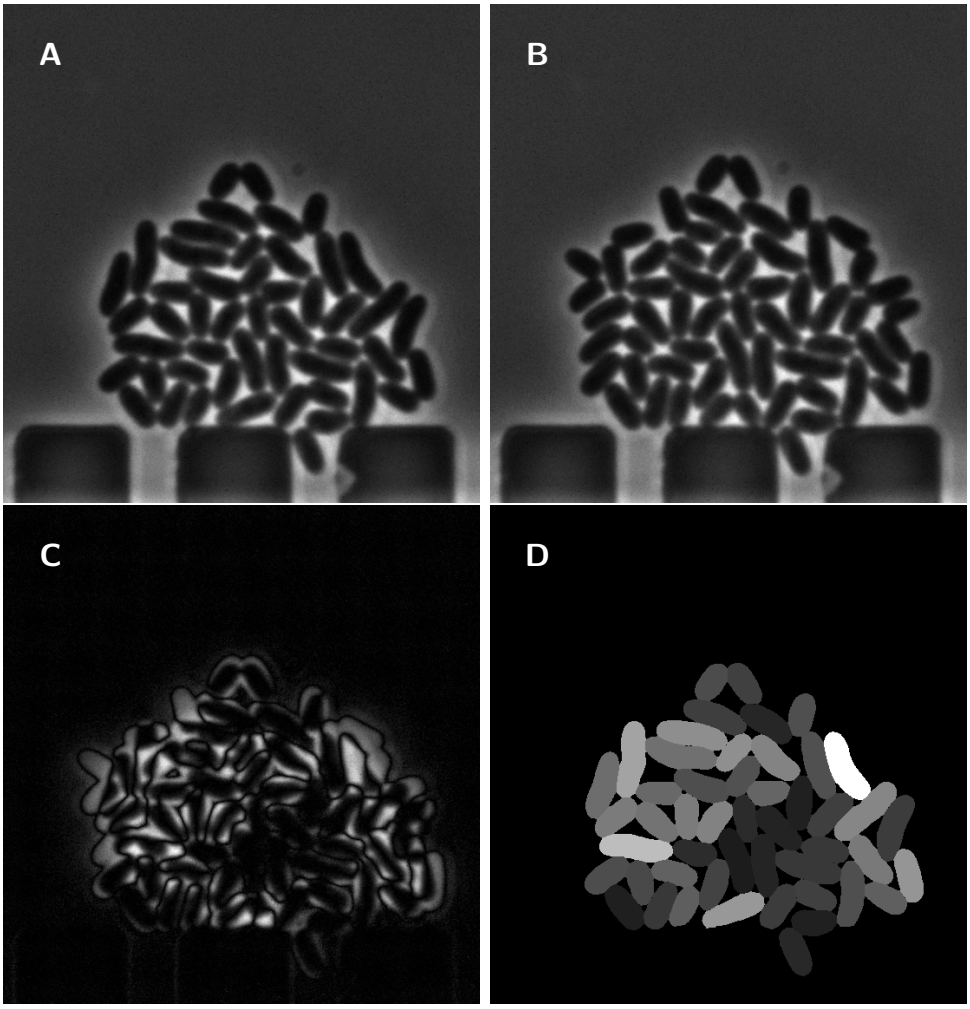


Figure 7.2.: Two consecutive phase contrast images I_t, I_{t+1} of *C. glutamicum* (**A** and **B**), the corresponding standard deviation S_t (**C**), and the AM (**D**).

7.2.3. Gaussian activity-based heatmap as a linking measure

One of our target microorganisms *C. glutamicum* is non-motile, i.e., the movement of a cell is expected only in the case of its division or division of the cells in its neighborhood. Moreover, the direction of the V-snap is hard or impossible to predict. Therefore, a linking measure to account for mentioned facts is desired.

As a linking measure, the traditionally used Euclidean distance between cells' centers [108] requires a globally set threshold to prevent "impossible" long-distance cell associations and does not consider that some cells are less likely to divide than others. To account for this, we propose a new linking measure, a new activity-based metric that restricts the daughter's search based on the mother's activity and thus offers fewer possible candidates compared to the Euclidean distance.

As an ingredient for the linking of the cells from t to $t+1$, for each cell from t, we use a 2D Gaussian function G centered at the center of mass of the i-th cell (c_i) and width $\sigma_i > 0$:

$$G_i^t(x,y) = \exp\left(-\tfrac{1}{2\sigma_i^2}((x-c_{x,i})^2 + (y-c_{y,i})^2)\right). \tag{7.3}$$

Here, σ_i is proportional to the activity of the i-th cell at time t (a_i), where a_i is a shorthand notation for $a_{t,i}$ when t is fixed, i.e $\sigma_i = a_i/k$, where $k > 0$ is a parameter. Since G is strictly positive but decaying exponentially, we reduce the number of possible candidates by treating G below 0.01 as zero. Unlike Euclidean distance thresholding, the proposed threshold is less sensitive to spatial image resolution because the Gaussian function scales relative distances rather than absolute pixel distances. However, the optimal value of k may still need adjustment depending on the pixel size, but the method remains more robust across varying resolutions compared to direct thresholding of the Euclidean distance. Fig. 7.3 illustrates the behavior of G with two cells similar in size but different in activity with $k = 2.5$. This value of k is used in all numerical experiments here.

7.2.4. Prioritization-based single assignment

We propose to split the assignment into two stages: A prioritization-based single assignment of growing cells, followed by a combinatorial linear assignment of splitting cells to their daughters. Before the assignment, we form two lists: the list of cells in frame t and the list of cells in frame $t+1$, where the cells in t are sorted in ascending order by activity (a_i). We formulate the linking loss between the i-th cell of frame t and the j-th cell of frame $t+1$ as follows:

$$L(i,j) = -G_i^t(c_{x,j}^{t+1}, c_{y,j}^{t+1}). \tag{7.4}$$

Here, G_i^t is the Gaussian map for the i-th cell of frame t, cf. Eq. 7.3, and $(c_{x,j}^{t+1}, c_{y,j}^{t+1}) \in \mathbb{R}^2$ is the center of mass of the j-th cell of frame $t+1$.

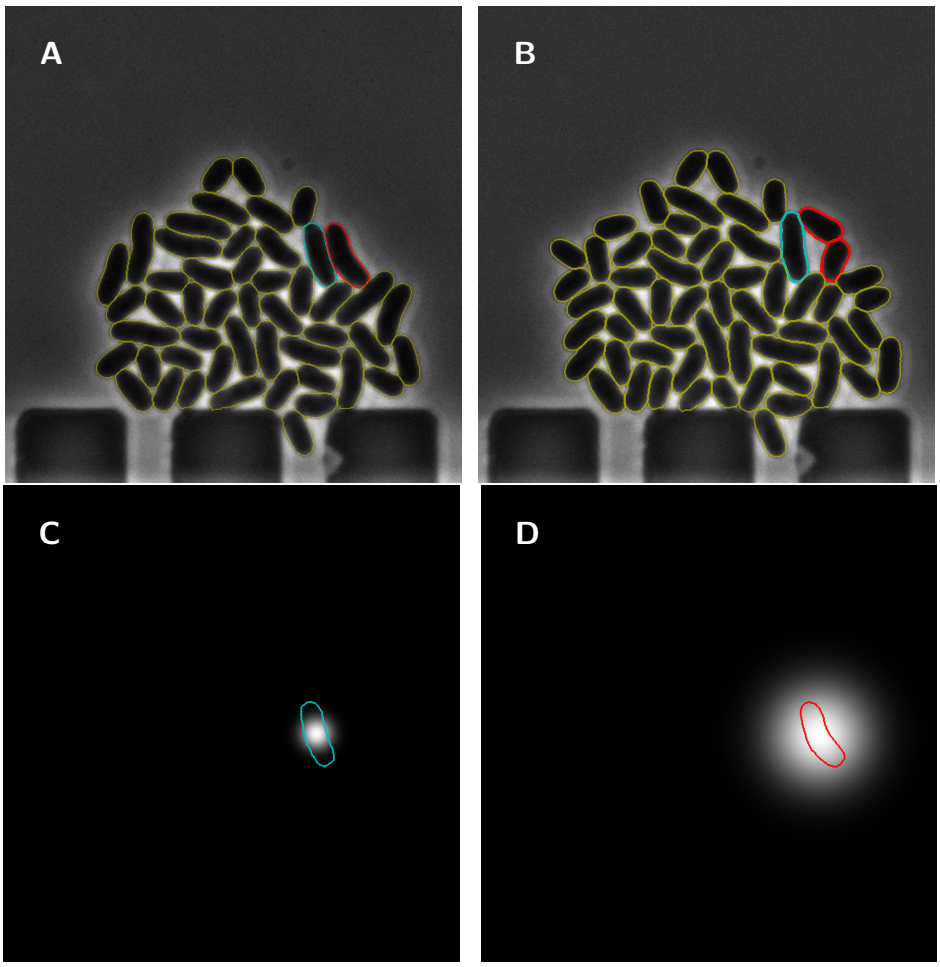


Figure 7.3.: Frames at times t (**A**) and $t+1$ (**B**) with highlighted cells: turquoise and red with relatively low and high activity in t (see Fig. 7.2) respectively. **C** The combination of **A** and **B** shows the correct t - $t+1$ assignment of the highlighted cells. **C** and **D** show the activity-based Gaussian maps G for the respective cells. Note that (**C**) and (**D**) were computed before the assignment and without using the tracking results shown in (**B**).

We pair the cells (over the cells in t in the sorted order) with minimal linking loss and check if a certain criterion is satisfied. As the criterion for a "valid" link, we assume that cells are growing or keep their size over time, i.e., a cell from frame $t+1$ has to have at least as large area as its link from frame t. If the cell shrinks by even 1 pixel, this criterion would not be satisfied, strictly enforcing the assumption that cells do not decrease in size. However, if the microorganism's behavior includes expected shrinking, this criterion can be tailored accordingly. If the criterion is satisfied, we remove the assigned cell from the list of possible candidates of frame t and the list of possible candidates frame $t+1$. After this initial assignment step, only the cells from frame t that either split (in most cases) or would have been erroneously assigned to smaller cells and their possible daughters/growing copies are without assignment. These unassigned cells next participate

in a double linear assignment step.

7.2.5. Linear assignment.

The so-called linear sum assignment problem is known as minimum weight matching in bipartite graphs. A problem instance is described by a matrix $C \in \mathbb{R}^{m \times d}$, where each matrix entry $C(i, j)$ is the cost (loss) of matching vertex i of the first set (a "worker", "mother" in our case) and vertex j of the second set (a "job" or "daughter"). The goal is to find a complete assignment of workers to jobs of minimal cost, i.e.,

$$\min_{X \in \mathcal{X}} \sum_i \sum_j C(i, j) X(i, j). \tag{7.5}$$

Here, $\mathcal{X}$ is the set of boolean matrices X with $\min(m, d)$ non-zero entries whose rows and columns sum to at most one and where, $X(i, j) = 1$ if cell i is assigned to cell j, and $X(i, j) = 0$ otherwise.

In the case that C is square, each row is assigned to exactly one column, and each column to exactly one row. Taking into account that one mother typically has two daughters, we extend the matrix C by doubling the number of rows (i.e., the number of mothers), stacking two identical rectangular matrices on top of each other. This results in an approximately square matrix, given that the number of daughters d is roughly twice the number of mothers m:

$$C_{2m \times d} = \begin{bmatrix} \hat{C}_{m \times d} \\ \hat{C}_{m \times d} \end{bmatrix} \tag{7.6}$$

where

$$\hat{C}_{m \times d} = \begin{bmatrix} L(1,1) & L(1,2) & \dots & L(1,d) \\ L(2,1) & L(2,2) & \dots & L(2,d) \\ \vdots & \vdots & \vdots & \vdots \\ L(m,1) & L(m,2) & \dots & L(m,d) \end{bmatrix}. \tag{7.7}$$

To solve the formulated problem, we use a so-called modified Jonker-Volgenant algorithm [155], implemented in [128]. This algorithm can also solve a generalization of the classic assignment problem, where the cost matrix is rectangular. If it has more rows than columns, then not every row needs to be assigned to a column, and vice versa. In case of an odd number of daughters when $d > 2m$, i.e., a new cell that was not present at frame t appears in frame $t + 1$, this cell will be left unassigned. On the contrary, when $d < 2m$, i.e., one of the daughters from frame $t + 1$ disappears, the mother will only be assigned one daughter.

The pseudocode of the proposed cell tracking strategy for two consecutive frames is shown as Algorithm 2.

Algorithm 2: The proposed two-stage assignment strategy

Data: Segmentation masks, Activity maps
Result: Cell pairs
for $t \leftarrow 0$ **to** $N-1$ **do**
  $Mothers \leftarrow$ [all *cells* at time t] ;
  sort ($Mothers$, ascending $Activity$);
  $Daughters \leftarrow$ [all *cells* at time $t+1$];
  $Pairs \leftarrow$ [];
  for *every* $cell_i(t)$ *in Mothers* **do**
    $\mathcal{L} \leftarrow$ [] ;
    for *every* $cell_j(t+1)$ *in Daughters* **do**
      $\mathcal{L}.append(L_{i,j})$;
    end
    if $area(cell_i(t)) \leq area(cell_{argmin(\mathcal{L})}(t+1))$ **then**
      remove $cell_{\mathrm{argmin}(\mathcal{L})}(t+1)$ from $Daughters$;
      remove $cell_i(t)$ from $Mothers$;
      $Pairs.append([cell_i(t), cell_{\mathrm{argmin}(\mathcal{L})}(t+1)])$;
    end
  end
  $C \leftarrow L(Mothers, Daughters)$;
  $Pairs.append(linear\_sum\_assignment(C))$;
end

7.3. Evaluation of the proposed cell tracking method

7.3.1. Ground truth generation for tracking

Given the lack of suitable GT data in the literature for our specific experimental setup, we generated our own GT for the datasets introduced in Section 7.1. We applied the proposed and baseline approach (see below) to the segmented highest time-resolution dataset and carefully manually checked and refined, if needed, the obtained lineage. In order to evaluate our algorithm for different frame rates, i.e., 2, 4, 6, 8, 10, and 12 minutes per frame for *C. glutamicum* and 5, 10, and 15 minutes per frame for *E. coli*, we down-sample our original datasets, respectively.

Additionally, the conducted frame rate sensitivity analysis of the algorithm for *C. glutamicum* and *E. coli* provides insights to the biotechnologists about the choice of the frame rate for their cell cultivation experiments, offering a trade-off between tracking accuracy and microscopy settings.

7.3.2. Comparison to a baseline approach

We use the TRA measure from the cell tracking challenge [144] to evaluate the performance of the proposed algorithm. The TRA score is based on the acyclic-oriented graph matching

(AOGM) measure [156], which assesses how accurately each given object (cell) has been identified and followed in successive frames by comparing the acyclic-oriented graph computed by an evaluated algorithm with the reference graph given by the GT. Numerically, TRA is defined as a normalized Acyclic Oriented Graph Matching (AOGM) measure:

$$\mathrm{TRA} = 1 - \frac{\min(\mathrm{AOGM}, \mathrm{AOGM}_0)}{\mathrm{AOGM}_0} \tag{7.8}$$

where AOGM_0 is the AOGM value required for creating the reference graph from scratch (i.e., it is the AOGM value for empty tracking results). The minimum operator in the numerator prevents the TRA value from being negative in the case when it is cheaper to create the reference graph from scratch than to transform the computed graph into the reference graph.

This measure accounts for various types of errors by penalizing the number of transformations needed to convert the predicted graph into the GT graph.

The types of errors penalized by the TRA measure include:

- Detection Errors: False positives (incorrectly identified cells) and false negatives (missed cells).
- Segmentation Errors: Errors where multiple cells are merged or a single cell is split.
- Tracking Errors: Missing links (failure to track a cell from one frame to the next), incorrect links (linking the wrong cells between frames), and semantic errors in the links (misidentifying the relationships between cells, such as parent-daughter relationships during cell division).

Since our goal is to evaluate the tracking accuracy alone, we provide the GT segmentation for both the baseline and the proposed approach. This ensures that any errors related to detection or segmentation do not influence the study, allowing us to focus exclusively on the correctness of the cell tracking.

As a baseline algorithm, we use a graph-based cell tracking algorithm [150], which was ranked in the top three at the time the method was developed for the majority of the datasets of the cell tracking challenge. The baseline algorithm offers segmentation refinement, such as untangling and false negative correction. Since, in our case, the tracking was performed on the GT segmentation with no need for refinement, we use two variants of the baseline approach: the algorithm with default parameter settings (`postprocessing_key=None`), and without the segmentation refinement step (`postprocessing_key='nd_ns+l'`). In the case of *E. coli* dataset evaluation, both settings lead to the same result. Thus, we only report the value at default settings (Fig. 7.5).

The high TRA values observed in the plots can be attributed to the way the TRA measure penalizes different types of errors. Specifically, the TRA measure applies higher penalties to detection errors, such as false negatives (penalty = 10) and false positives (penalty = 1), as well as segmentation errors like splitting operations (penalty = 5). In

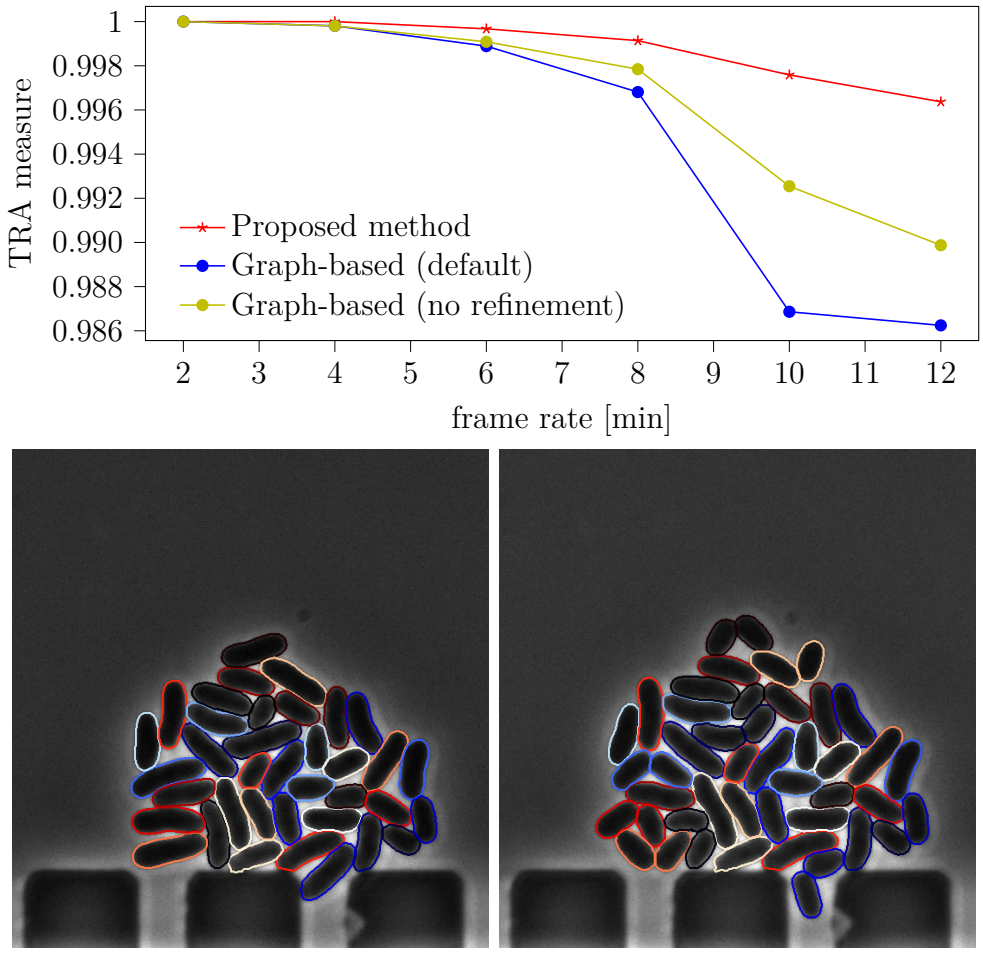


Figure 7.4.: Top: A quantitative comparison of the proposed method vs. the baseline approach for different frame rates *C. glutamicum*. The bottom row illustrates the tracking results of the proposed method for two consecutive frames of the image sequence of a *C. glutamicum* sequence with 12 min frame rate. All the illustrated assignments are correct.

contrast, tracking errors, such as redundant edges that need to be deleted (penalty = 1), edges that need to be added (penalty = 1.5), and edges with incorrect semantics (penalty = 1), are penalized less severely.

This means that even with some tracking errors, the overall TRA score remains high (higher than 0.98) because these errors carry lower penalties. Consequently, a TRA, which might seem slightly off (0.986) at first glance, is actually quite strong, given the relatively low penalties for tracking errors.

Nevertheless, we chose to use the TRA measure because it is widely recognized in the community, standardized, and provides an unbiased comparison across different methods. It allows us to maintain consistency with other studies and to benchmark our method fairly against existing approaches.

As can be observed in Fig. 7.4, the proposed method outperforms the baseline method

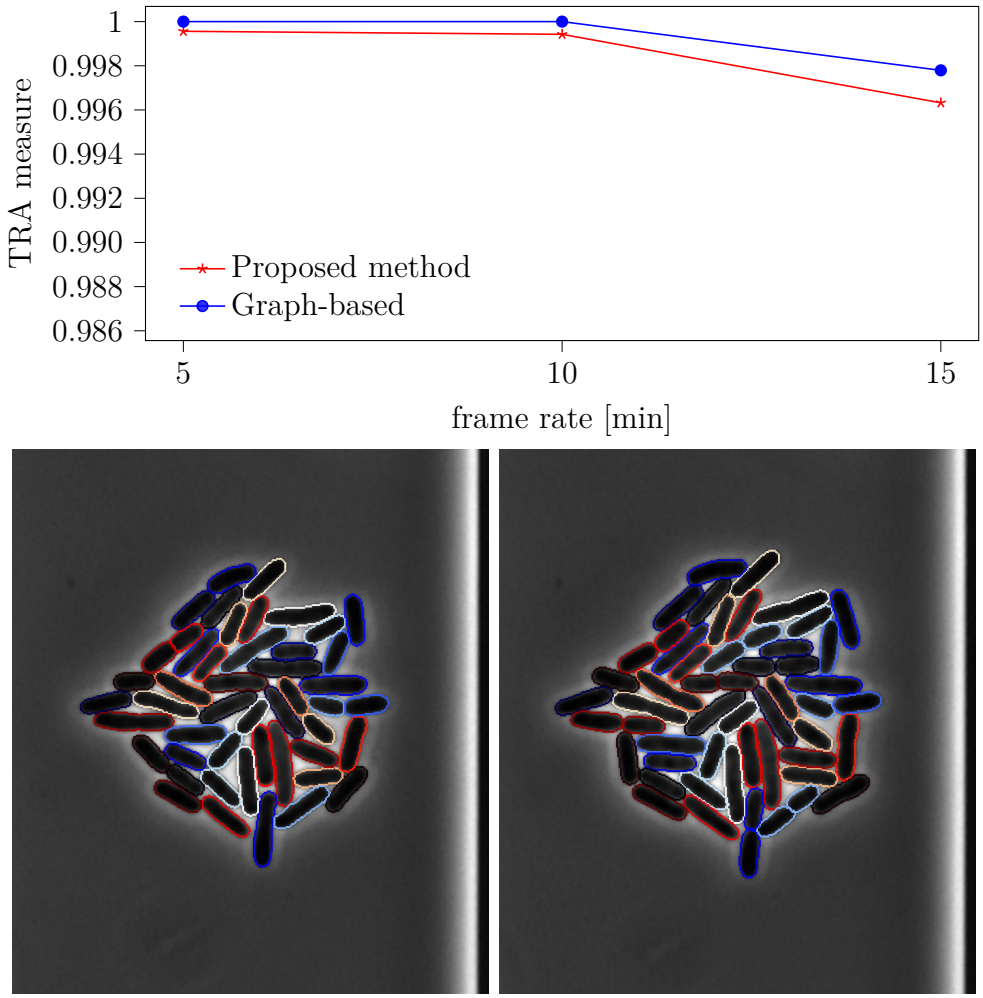


Figure 7.5.: Top row: A quantitative comparison of the proposed method vs. the baseline approach for different frame rates *E. coli* (right). The bottom row illustrates the tracking results of the proposed method for two consecutive frames of *E. coli* (15 min). All the illustrated assignments are correct.

when applied to a *C. glutamicum* dataset and is on par with the baseline approach applied to the *E. coli* dataset (Fig. 7.5). The cross-correlation nature of the baseline approach and relatively simple (no snapping, the angle of daughters is preserved) division behavior of *E. coli* explains the slightly better results of the method whereas the proposed method handles the "complex" V-snapping division behavior of *C. glutamicum* better. This is because the proposed activity map better identifies and prioritizes regions of the image where cell divisions and other significant movements occur, leading to more accurate tracking in scenarios with complex behaviors.

7.4. Application case: Fluorescence distribution tracking in oxygen-controlled microfluidic cultures

This section is based upon the publication "Enabling oxygen-controlled microfluidic cultures for spatiotemporal microbial single-cell analysis" by K. Kasahara, M. Leygeber, J. Seiffarth, **K. Ruzaeva**, T. Drepper, K. Nöh and D. Kohlheyer. The author of this thesis performed the tracking analysis to study the fluorescence distribution over the cell generation

Oxygen levels play a crucial role in influencing microbial growth, metabolism, and behavior. Traditional culture methods, however, do not allow for dynamic control and precise monitoring of oxygen levels. In this section, we apply the tracking approach developed earlier to a specific application: analyzing fluorescence distribution in microbial cells cultivated in a microfluidic device with controlled oxygen levels.

Microfluidic devices create controlled microenvironments, offering a potential solution for studying microorganisms under well-defined conditions. Despite their promise, maintaining and manipulating specific oxygen concentrations within these devices presents a significant challenge. Here, we explore how the developed tracking method can be utilized to study the spatiotemporal fluorescence distribution within microbial cells across different generations under varying oxygen conditions.

In this study, we aim to understand how oxygen availability influences cellular behavior and fluorescence expression over time by comparing the fluorescence distribution over cell generations under aerobic and anaerobic conditions. The investigation builds upon the microfluidic system mentioned earlier in the Introduction (Section 1.2.1), which enables the cultivation of cells under controlled oxygen conditions for spatiotemporal single-cell analysis.

The experiment involved cultivating cells under both aerobic (with oxygen) and anaerobic (without oxygen) conditions. During the second experiment, cells were initially grown under anaerobic conditions in a microfluidic device where the oxygen levels were tightly controlled. At a specific time point, which was set at $t = 08{:}00$ h, the environment within the microfluidic chambers was switched from anaerobic to aerobic by introducing oxygen into the medium. This setup allowed for a comparison of cell behavior under varying oxygen availability.

To analyze these differences, we employed the activity-prioritized cell tracking method introduced in the previous section (Chapter 7), which allows the exploration of single-cell fluorescence over successive cell generations.

By employing this method, the mean intensity of Green Fluorescent Protein (GFP) fluorescence (I_{cell}) from segmented cells was determined, and their corresponding generation numbers (representing the number of observed cell divisions) were tracked. Cells with high generation numbers indicated fast growth, while cells with lower generation numbers indicated slow growth. For analysis purposes, one exemplary micro-population was chosen from both the aerobic and anaerobic cultures. Subsequently, three consecutive

time points (t=08:00h, 08:30h, and 09:00 h) were selected to visualize the data analysis.

7.4.1. Observations

Under aerobic conditions, a growth rate of $\mu_{\text{aerobic}} = 0.59\ \text{h}^{-1}$ was derived for the selected population. As shown in the fluorescence time-lapse images of Fig. 7.6 A, a pronounced and relatively homogeneous intracellular GFP fluorescence was detected at all time points (8, 8.5, and 9 hours), the detected GFP fluorescence is visualized as bright regions within each cell, indicating intracellular GFP expression. In addition to single cell fluorescence levels, image analysis provided each corresponding cell generation, with the results visualized in (Fig. 7.6 B), where the color coding from blue to yellow indicates the generation. By plotting single-cell fluorescence I_{cell} over the generation (Fig. 7.6 C), correlations could be visualized. Here a slight tendency that cells with lower generation numbers (i.e., slow-growing cells) exhibited higher GFP fluorescence, while cells with higher generation numbers (i.e., fast-growing cells) exhibited lower GFP fluorescence for t = 08:00 h.

Under anaerobic conditions ($\mu_{\text{anaerobic}} = 0.50\ \text{h}^{-1}$) this effect was more prominent. As shown in the fluorescence images in Fig. 7.6D, the GFP fluorescence was negligible before the switch at t = 08:00 h. After the switch from anaerobic to aerobic conditions, the GFP fluorescence of the cells rapidly increased, but phenotypic population heterogeneity in terms of fluorescence intensity was more pronounced. The cells corresponding generation numbers are visualized in Fig. 7.6 E, also indicating a more pronounced heterogeneity in terms of single-cell growth rates. Moreover, the plot showing I_{cell} over generation number (Fig. 7.6 F) shows that intracellular GFP fluorescence decreases with the increasing division rate.

7.5. Summary of contributions

In this chapter, the tracking-free mapping of active cells — the AM was proposed. The AM can be a standalone metric of cell activity, a tool to be used in tracking annotation software to attract the user's attention to the "changing" cells, or a building block for cell tracking. Additionally, to tackle the challenges mentioned in Part IV, we introduced a feature-free tracking approach that utilizes the AM. The proposed tracking method consists of two steps: a "activity"-based prioritization single-cell assignment strategy of growing (non-splitting) cells and a combinatorial mother-daughter assignment of the dividing cells, using Jonker-Volgenant [155] algorithm to solve the linear assignment problem. We evaluated the proposed algorithm on datasets representing two important biotechnologically relevant rod-shaped microorganisms: *C. glutamicum*, with peculiar V-snapping division behavior, and *E. coli*, which divides uniformly by linear elongation. The numerical experiments show that the proposed tracking approach outperforms the baseline approach (a graph-based cell tracking algorithm [150]) in *C. glutamicum* tracking and is on par with the baseline approach in *E. coli* tracking.

Additionally, to evaluate the algorithm's performance on different frame-rate datasets,

we hope that the reported tracking quality metrics can provide insights to biotechnologists and help them choose the proper frame rate for their cultivation experiments.

Moreover, the proposed tracking method demonstrated its effectiveness in a study case, where it was successfully employed to investigate fluorescence distribution over cell generations for cells cultivated under aerobic and anaerobic conditions.

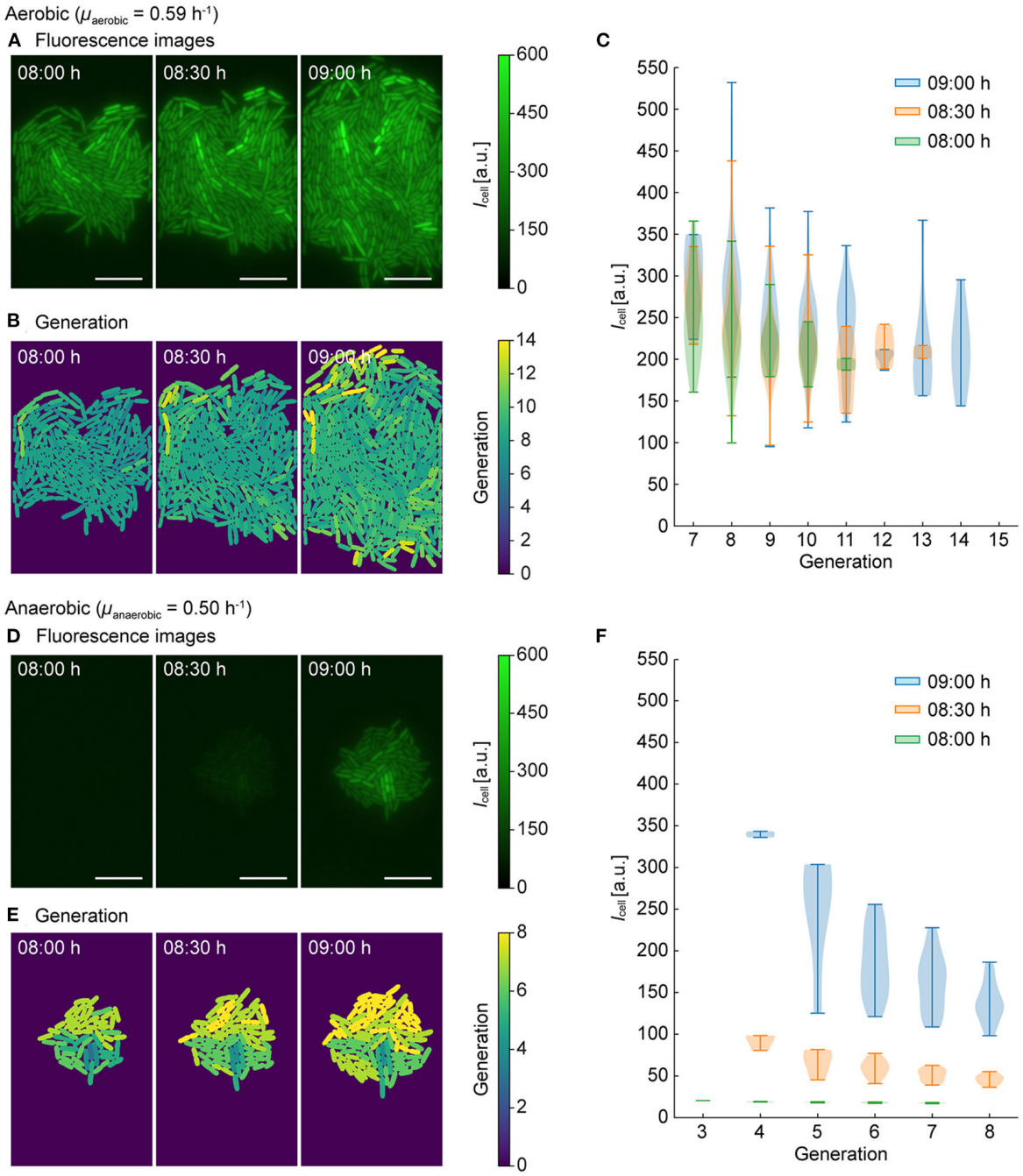


Figure 7.6.: Single-cell tracking and analysis of fluorescence intensities (I_{cell}) and generation numbers for aerobic (**A**, **B**, **C**) and anaerobic conditions with a switch to aerobic conditions (**D**, **E**, **F**). **A**, **D** Fluorescence intensity images at t = 08:00 h, 08:30 h, and 09:00 h (scale bars = 10 μm). **B**, **E** Generations number (i.e., number of division events detected) visualized by color coding. **C**, **F** Distribution of I_{cell} over cell generations at three time points. The figure is taken from [16].

Summary

This dissertation presents a comprehensive study of the development of advanced geometry-aware image analysis methods for microfluidic live-cell experimentation. It focuses on overcoming the challenges associated with analyzing the heterogeneous and dynamic behavior of single cells within microfluidic environments, particularly by considering the morphology of microorganisms and their behavior. The work is grounded in several published papers, which are thoroughly discussed throughout the thesis.

In the first part, two approaches for generating GT data for biological objects are proposed. The first method, polar space-based shape averaging, combines polar transformation and statistical methods to accurately represent the object's shape based on experts' annotation proposals. This results in a smooth, naturally looking segmentation GT. The method utilizes the geometric property of most organisms being star-shaped and relies on averaging contours in polar space. It is robust, computationally efficient, and works without introducing tuneable parameters. Moreover, the algorithm preserves the uncertainty (in terms of the standard deviation) of the experts' annotations, allowing the introduction of an uncertainty-aware metric to estimate the segmentation quality. This metric emphasizes the influence of GT regions with low variance. The performance of the proposed averaging method is studied on time-lapse microscopy data of *C. glutamicum* and the uncertainty-aware metric on synthetic data.

The second method, CellSium, is a versatile and user-friendly cell simulator designed to generate high-quality GT data across a wide range of microbial colony shapes and sizes. CellSium's flexibility allows it to simulate various experimental conditions and morphological variations of the cells, which makes it a valuable tool for creating customized datasets for training and evaluating image analysis algorithms. The simulator models microbial growth patterns and spatial arrangements in microcolonies, as well as various imaging artifacts and noise characteristics, enabling the generation of synthetic yet realistic datasets that reflect real-world experiments. These capabilities are particularly crucial for studies where real annotated datasets are scarce or labor-intensive to produce.

In the second part, a hybrid multi-object segmentation framework is presented, which combines machine learning-based detection with a model-based segmentation algorithm utilizing B-splines and allowing the incorporation of prior knowledge about cell geometries. This framework is effective for the robust and accurate segmentation of microbial cells in microfluidic experiments. The method is shown to be effective in segmenting cells using an easy-to-create training dataset.

Additionally, the segmentation method for automated characterization of CatIB production in biotechnological screening systems is proposed and presents an automated method for characterizing the quantity and sizes of CatIBs produced in biotechnological screening systems. The method was based on image analysis techniques, including ML-

based detection, B-spline-based segmentation, thresholding, and morphological operations. It was shown to provide accurate results for quantifying the size and shape of CatIBs.

In the third part, the thesis proposes an automated cell tracking method based on an activity-prioritized assignment strategy. The approach combines the assignment of growing cells using the AM and combinatorial mother-daughter assignment. The method is shown to be effective in tracking cells in a range of microfluidic experiments, including those with highly peculiar division behaviors. The proposed method outperforms the baseline tracking method in terms of TRA measure, making it a valuable tool for analyzing the dynamic behavior of single cells in microfluidic devices.

This dissertation presents a comprehensive and effective approach for image-based analysis of microfluidic live-cell experiments. By combining GT generation, segmentation, and tracking methods, the proposed approaches allow for accurate and detailed analysis of cell behaviors and interactions.

The proposed methods have been shown to provide high-quality results for analyzing the heterogeneous dynamic behavior of single cells in microfluidic devices.

Outlook and future work

The research conducted in this dissertation has addressed several issues in the field of geometry-aware image analysis for microfluidic live-cell experimentation. However, several areas remain where further exploration and development could enhance the applicability and effectiveness of the proposed methods.

The proposed segmentation methods have been effectively used for some bacterial cell forms, including ovoid and rod-shaped cells. Future work should focus on expanding these methods to accommodate more intricate morphologies, like those seen in eukaryotic cells or irregularly shaped bacteria. Developing models that can accurately segment and track these complex shapes will require new approaches to parameterization and may involve integrating machine learning techniques with traditional geometric models. Additionally, the methods developed here could potentially be adapted for use with other biological systems, such as plant cells, mammalian cells, or even tissues. Expanding the applicability of these methods to other systems would require adapting the models to account for the different geometries and behaviors of the microorganisms and new experimental conditions.

As biotechnological processes continue to scale up, the ability to analyze large datasets efficiently becomes more important. Integrating the methods developed in this work with high-throughput data processing pipelines would enable the analysis of thousands of images in real time, making it possible to monitor and optimize bioprocesses continuously and provide immediate feedback during live-cell experiments. However, achieving true real-time performance may require further optimizations, such as implementing parallel processing. Once optimized, it would be possible to dynamically adjust experimental conditions based on the ongoing analysis of cell behavior, leading to more efficient and responsive experiments.

While this dissertation has introduced robust methods for producing ground truth data, further improvements could be achieved by including more complex noise models or replicating more intricate experimental settings. Furthermore, the synthetic datasets might become even more realistic through the application of techniques like diffusion models [113] or generative adversarial networks (GANs) [112], which would make them even more valuable for training and validating machine learning models.

To increase the impact of the work presented here, it would be beneficial to develop user-friendly software tools that experimental scientists can easily use. These tools could be integrated into existing image analysis platforms or distributed as standalone applications.

Addressing these future directions could further refine and expand the methods developed in this dissertation, leading to even more significant contributions to the field of geometry-aware image analysis for microfluidic live-cell experimentation.

Bibliography

[1] Biotechnology Innovation Organization. *What is Industrial Biotechnology?* URL: https://archive.bio.org/articles/what-industrial-biotechnology (visited on 12/30/2022).

[2] R. R. Kumar and S. Prasad. "Metabolic Engineering of Bacteria." In: *Indian Journal of Microbiology* 51.3 (Mar. 2011), pp. 403–409.

[3] A. Grünberger. *Single cell analysis of microbial production strains in microfluidic bioreactors*. Vol. 114. Schriften des Forschungszentrums Jülich : Reihe Schlüsseltechnologien. Jülich: Forschungszentrum Jülich, Zentralbibliothek, 2015.

[4] S. D. Doig, F. Baganz, and G. J. Lye. "High-throughput screening and process optimisation." In: *Basic Biotechnology*. Cambridge University Press, May 2006, pp. 289–306.

[5] S. Helfrich. *High-throughput Live-cell Imaging for Investigations of Cellular Heterogeneity in Corynebacterium Glutamicum* -. Universitätsbibliothek der RWTH Aachen, 2016.

[6] T. S. Tikhomirova, M. S. Taraskevich, and O. V. Ponomarenko. "The role of laboratory-scale bioreactors at the semi-continuous and continuous microbiological and biotechnological processes." In: *Applied Microbiology and Biotechnology* 102.17 (July 2018), pp. 7293–7308.

[7] J. Büchs. "Introduction to advantages and problems of shaken cultures." In: *Biochemical Engineering Journal* 7.2 (Mar. 2001), pp. 91–98.

[8] J. I. Betts and F. Baganz. In: *Microbial Cell Factories* 5.1 (2006), p. 21.

[9] T. Linde, N. Hansen, M. Lübeck, and P. Lübeck. "Fermentation in 24-well plates is an efficient screening platform for filamentous fungi." In: *Letters in Applied Microbiology* 59.2 (Apr. 2014), pp. 224–230.

[10] E. H. Solutions. *It's Not Just About Size: Talking about Shake Flasks and Bioreactors*. URL: https://handling-solutions.eppendorf.com/cell-handling/bioprocess/processes-and-applications/detailview/news/its-not-just-about-size-talking-about-shake-flasks-and-bioreactors/ (visited on 12/30/2022).

[11] B. C. L. Sciences. *Advantages and Applications of the BioLector XT Microbioreactor*. URL: https://www.beckman.com/microbioreactor/biolector-xt/advantages-and-applications (visited on 12/30/2022).

[12] R. Jansen. "Advancing automated bioprocess development for industrial platform organisms." Dissertation. RWTH Aachen University, 2020.

[13] A. Grünberger, N. Paczia, C. Probst, G. Schendzielorz, L. Eggeling, S. Noack, W. Wiechert, and D. Kohlheyer. "A disposable picolitre bioreactor for cultivation and investigation of industrially relevant bacteria on the single cell level." In: *Lab on a Chip* 12.11 (2012), p. 2060.

[14] C. Probst. "Microfluidic tools for single cell analysis." Thesis. Universität Siegen, Institut für Werkstofftechnik, Jan. 2016.

[15] C. C. Sachs. "Online high throughput microfluidic single cell analysis for feed-back experimentation." Dissertation. RWTH Aachen University, 2018.

[16] K. Kasahara, M. Leygeber, J. Seiffarth, K. Ruzaeva, T. Drepper, K. Nöh, and D. Kohlheyer. "Enabling oxygen-controlled microfluidic cultures for spatiotemporal microbial single-cell analysis." In: *Frontiers in Microbiology* 14 (June 2023).

[17] A. Grünberger, W. Wiechert, and D. Kohlheyer. "Single-cell microfluidics: opportunity for bioprocess development." In: *Current Opinion in Biotechnology* 29 (Oct. 2014), pp. 15–23.

[18] A. Grünberger, J. van Ooyen, N. Paczia, P. Rohe, G. Schiendzielorz, L. Eggeling, W. Wiechert, D. Kohlheyer, and S. Noack. "Beyond growth rate 0.6: Corynebacterium glutamicum cultivated in highly diluted environments." In: *Biotechnology and Bioengineering* 110.1 (2013), pp. 220–228. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/bit.24616`.

[19] M. Carlquist, R. L. Fernandes, S. Helmark, A.-L. Heins, L. Lundin, S. J. Sørensen, K. V. Gernaey, and A. E. Lantz. "Physiological heterogeneities in microbial populations and implications for physical stress tolerance." In: *Microbial Cell Factories* 11.1 (July 2012).

[20] K. Küsters. "Accelerated production and screening of catalytically active inclusion body libraries via automated workflows." en. PhD thesis. 2022, p. 2022.

[21] R. Jansen, K. Küsters, H. Morschett, W. Wiechert, and M. Oldiges. "A fully automated pipeline for the dynamic at-line morphology analysis of microscale Aspergillus cultivation." In: *Fungal Biology and Biotechnology* 8.1 (Mar. 2021).

[22] D. Ray, U. Anand, N. K. Jha, E. Korzeniewska, E. Bontempi, J. Proćków, and A. Dey. "The soil bacterium, Corynebacterium glutamicum, from biosynthesis of value-added products to bioremediation: A master of many trades." In: *Environmental Research* 213 (Oct. 2022), p. 113622.

[23] L. Eggeling and M. Bott, eds. *Handbook of Corynebacterium glutamicum*. CRC Press, Mar. 2005.

[24] K.-A. Baritugo, H. T. Kim, Y. David, J.-i. Choi, S. H. Hong, K. J. Jeong, J. H. Choi, J. C. Joo, and S. J. Park. "Metabolic engineering of Corynebacterium glutamicum for fermentative production of chemicals in biorefinery." In: *Applied Microbiology and Biotechnology* 102.9 (2018), pp. 3915–3937.

[25] A. Zahoor, S. N. Lindner, and V. F. Wendisch. "Metabolic engineering of Corynebacterium glutamicum aimed at alternative carbon sources and new products." In: *Computational and Structural Biotechnology Journal* 3.4 (Oct. 2012), e201210004.

[26] A. Mhatre, S. Shinde, A. K. Jha, A. Rodriguez, Z. Wardak, A. Jansen, J. M. Gladden, A. George, R. W. Davis, and A. M. Varman. "Corrigendum: Corynebacterium glutamicum as an efficient omnivorous microbial host for the bioconversion of lignocellulosic biomass." In: *Frontiers in Bioengineering and Biotechnology* 10 (Dec. 2022).

[27] J. Becker and C. Wittmann. "Advanced Biotechnology: Metabolically Engineered Cells for the Bio-Based Production of Chemicals and Fuels, Materials, and Health-Care Products." In: *Angewandte Chemie International Edition* 54.11 (Feb. 2015), pp. 3328–3350.

[28] Y. Nakayama. "Corynebacterium glutamicum Mechanosensing: From Osmoregulation to L-Glutamate Secretion for the Avian Microbiota-Gut-Brain Axis." In: *Microorganisms* 9.1 (Jan. 2021), p. 201.

[29] S.-D. Park, J.-Y. Lee, S. Sim, Y. Kim, and H.-S. Lee. "Characteristics of methionine production by an engineered Corynebacterium glutamicum strain." In: *Metabolic engineering* 9 4 (2007), pp. 327–36.

[30] F. Käß, I. Hariskos, A. Michel, H. Brandt, R. Spann, S. Junne, W. Wiechert, P. Neubauer, and M. Oldiges. "Assessment of robustness against dissolved oxygen/substrate oscillations for C. glutamicum DM1933 in two-compartment bioreactor." In: *Bioprocess and Biosystems Engineering* 37 (2014), pp. 1151–1162.

[31] X. Shen, Y. Huang, and S. Liu. "Genomic Analysis and Identification of Catabolic Pathways for Aromatic Compounds in Corynebacterium glutamicum." In: *Microbes and Environments* 20 (2005), pp. 160–167.

[32] B. Alberts. *Molecular Biology of the Cell*. 4th ed. New York: Garland Science, 2017.

[33] K. J. Kieser and E. J. Rubin. "How sisters grow apart: mycobacterial growth and division." In: *Nature Reviews Microbiology* 12.8 (July 2014), pp. 550–562.

[34] X. Zhou, F. P. Rodriguez-Rivera, H. C. Lim, J. C. Bell, T. G. Bernhardt, C. R. Bertozzi, and J. A. Theriot. "Sequential assembly of the septal cell envelope prior to V snapping in Corynebacterium glutamicum." In: *Nature Chemical Biology* 15.3 (Jan. 2019), pp. 221–231.

[35] R. Morra, J. Shankar, C. Robinson, S. Halliwell, L. Butler, M. Upton, S. Hay, J. Micklefield, and N. Dixon. "Dual transcriptional-translational cascade permits cellular level tuneable expression control." In: *Nucleic Acids Research* 44 (2015), e21–e21.

[36] H. Mori. "From the sequence to cell modeling: comprehensive functional genomics in Escherichia coli." In: *Journal of biochemistry and molecular biology* 37 1 (2004), pp. 83–92.

[37] V. Gupta, G. Radhakrishnan, J. Harms, and G. Splitter. "Invasive Escherichia coli vaccines expressing Brucella melitensis outer membrane proteins 31 or 16 or periplasmic protein BP26 confer protection in mice challenged with B. melitensis." In: *Vaccine* 30 27 (2012), pp. 4017–22.

[38] J. M. Clomburg and R. Gonzalez. "Biofuel production in Escherichia coli: the role of metabolic engineering and synthetic biology." In: *Applied Microbiology and Biotechnology* 86.2 (Feb. 2010), pp. 419–434.

[39] R. Gonzalez, P. Campbell, and M. Wong. "Production of ethanol from thin stillage by metabolically engineered Escherichia coli." In: *Biotechnology Letters* 32.3 (Nov. 2009), pp. 405–411.

[40] Y. Sohn, H. T. Kim, K.-A. Baritugo, H. M. Song, M. Ryu, K. Kang, S. Jo, H. Kim, Y. J. Kim, J.-i. Choi, S. K. Park, J. Joo, and S. J. Park. "Biosynthesis of polyhydroxyalkanoates from sucrose by metabolically engineered Escherichia coli strains." In: *International journal of biological macromolecules* (2020).

[41] M. Valenzuela-Ortega, J. T. Suitor, M. F. M. White, T. Hinchcliffe, and S. Wallace. "Microbial Upcycling of Waste PET to Adipic Acid." In: *ACS Central Science* 9.11 (Nov. 2023), pp. 2057–2063.

[42] National Institute of Allergy and Infectious Diseases (NIAID). *E. coli*. URL: `https://www.niaid.nih.gov/diseases-conditions/e-coli` (visited on 08/07/2023).

[43] J. S. Grah, J. A. Harrington, S. B. Koh, J. A. Pike, A. Schreiner, M. Burger, C.-B. Schönlieb, and S. Reichelt. "Mathematical imaging methods for mitosis analysis in live-cell phase contrast microscopy." In: *Methods* 115 (2017), pp. 91–99.

[44] M. W. D. Douglas B. Murphy. *Fundamentals of Light Microscopy and Electronic Imaging*. BLACKWELL PUBL, Nov. 2012. 560 pp.

[45] H. Li, H. Zhang, Y. Xu, A. Tureckova, P. Zahradník, H. Chang, and P. Neuzil. "Versatile digital polymerase chain reaction chip design, fabrication, and image processing." In: *Sensors and Actuators B: Chemical* 283 (2019), pp. 677–684.

[46] M. Beck, S. Brockhuis, N. van der Velde, C. Breukers, J. Greve, and L. W. M. M. Terstappen. "On-chip sample preparation by controlled release of antibodies for simple CD4 counting." In: *Lab Chip* 12.1 (2012), pp. 167–173.

[47] Y. Zhang, Y. Shin, K. Sung, S. Yang, H. Chen, H. Wang, D. Teng, Y. Rivenson, R. P. Kulkarni, and A. Ozcan. "3D imaging of optically cleared tissue using a simplified CLARITY method and on-chip microscopy." In: *Science Advances* 3.8 (Aug. 2017).

[48] M. Elsayed, T. M. El-Sherry, and M. Abdelgawad. "Development of computer-assisted sperm analysis plugin for analyzing sperm motion in microfluidic environments using Image-J." In: *Theriogenology* 84.8 (Nov. 2015), pp. 1367–1377.

[49] Z. Zhang, L. Chen, Y. Wang, T. Zhang, Y.-C. Chen, and E. Yoon. "Label-Free Estimation of Therapeutic Efficacy on 3D Cancer Spheres Using Convolutional Neural Network Image Analysis." In: *Analytical Chemistry* 91.21 (Oct. 2019), pp. 14093–14100.

[50] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona. "Fiji: an open-source platform for biological-image analysis." In: *Nature Methods* 9.7 (June 2012), pp. 676–682.

[51] N. Otsu. "A Threshold Selection Method from Gray-Level Histograms." In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (1979), pp. 62–66.

[52] W. Niblack. *An Introduction to Digital Image Processing* -. New Jersey: Prentice-Hall International, 1986.

[53] J. Bernsen. "Dynamic Thresholding of Grey-Level Images,[in:] Proc. of the 8th Int." In: *Conf. on Pattern Recognition*. 1986.

[54] S. Beucher and C. Mathmatique. "The Watershed Transformation Applied To Image Segmentation." In: *Scanning Microscopy Supplement* 6 (July 1992), pp. 299–314.

[55] P. Bankhead. *Introduction to Bioimage Analysis*. 2022.

[56] L. A. Vese and C. L. Guyader. *Variational Methods in Image Processing*. Chapman and Hall/CRC, Nov. 2015.

[57] P. Getreuer. "Chan-Vese Segmentation." In: *Image Processing On Line* 2 (Aug. 2012), pp. 214–224.

[58] T. Chan and L. Vese. "An Active Contour Model without Edges." In: *Scale-Space Theories in Computer Vision*. Springer Berlin Heidelberg, 1999, pp. 141–151.

[59] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu. "scikit-image: image processing in Python." In: *PeerJ* 2 (2014), e453.

[60] S. Agatonovic-Kustrin and R. Beresford. "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research." In: *Journal of Pharmaceutical and Biomedical Analysis* 22.5 (2000), pp. 717–727.

[61] J. Unpingco. "Machine Learning." In: *Python for Probability, Statistics, and Machine Learning*. Cham: Springer International Publishing, 2022, pp. 359–502.

[62] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." In: *Lecture Notes in Computer Science*. Springer International Publishing, 2015, pp. 234–241.

[63] K. He, G. Gkioxari, P. Dollar, and R. Girshick. "Mask R-CNN." In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2017.

[64] N. Saunier and T. Sayed. "A feature-based tracking algorithm for vehicles in intersections." In: *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*. IEEE. 2006, pp. 59–59.

[65] A. Prioletti, A. Møgelmose, P. Grisleri, M. M. Trivedi, A. Broggi, and T. B. Moeslund. "Part-based pedestrian detection and feature-based tracking for driver assistance: real-time, robust algorithms, and evaluation." In: *IEEE Transactions on Intelligent Transportation Systems* 14.3 (2013), pp. 1346–1359.

[66] S. Liu, D. Liu, G. Srivastava, D. Połap, and M. Woźniak. "Overview and methods of correlation filter algorithms in object tracking." In: *Complex Intelligent Systems* (June 2020).

[67] B. K. Horn and B. G. Schunck. "Determining optical flow." In: *Artificial Intelligence* 17.1-3 (Aug. 1981), pp. 185–203.

[68] S. Aslani and H. Mahdavi-Nasab. "Optical flow based moving object detection and tracking for traffic surveillance." In: *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering* 7.9 (2013), pp. 1252–1256.

[69] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems." In: *Journal of Basic Engineering* 82.1 (Mar. 1960), pp. 35–45.

[70] J. S. Liu and R. Chen. "Sequential Monte Carlo Methods for Dynamic Systems." In: *Journal of the American Statistical Association* 93.443 (Sept. 1998), pp. 1032–1044.

[71] M. Marron, J. Garcia, M. Sotelo, M. Cabello, D. Pizarro, F. Huerta, and J. Cerro. "Comparing a Kalman Filter and a Particle Filter in a Multiple Objects Tracking Application." In: *2007 IEEE International Symposium on Intelligent Signal Processing*. IEEE, 2007.

[72] V. Katariya, M. Baharani, N. Morris, O. Shoghli, and H. Tabkhi. "DeepTrack: Lightweight Deep Learning for Vehicle Trajectory Prediction in Highways." In: *IEEE Transactions on Intelligent Transportation Systems* 23.10 (2022), pp. 18927–18936.

[73] Y. Xu, A. sep, Y. Ban, R. Horaud, L. Leal-Taixe, and X. Alameda-Pineda. "How to Train Your Deep Multi-Object Tracker." In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2020.

[74] X. Chen, C. Shen, W.-b. Zhang, M. Tomizuka, Y. Xu, and K. Chiu. "Novel hybrid of strong tracking Kalman filter and wavelet neural network for GPS/INS during GPS outages." In: *Measurement* 46.10 (2013), pp. 3847–3854.

[75] X. Chen, X. Xu, Y. Yang, Y. Huang, J. Chen, and Y. Yan. "Visual ship tracking via a hybrid kernelized correlation filter and anomaly cleansing framework." In: *Applied Ocean Research* 106 (2021), p. 102455.

[76] M. M. Usaj, C. H. L. Yeung, H. Friesen, C. Boone, and B. J. Andrews. "Single-cell image analysis to explore cell-to-cell heterogeneity in isogenic populations." In: *Cell Systems* 12.6 (June 2021), pp. 608–621.

[77] V. Ulman, M. Maška, K. E. G. Magnusson, O. Ronneberger, C. Haubold, N. Harder, P. Matula, P. Matula, D. Svoboda, M. Radojevic, I. Smal, K. Rohr, J. Jaldén, H. M. Blau, O. Dzyubachyk, B. Lelieveldt, P. Xiao, Y. Li, S.-Y. Cho, A. C. Dufour, J.-C. Olivo-Marin, C. C. Reyes-Aldasoro, J. A. Solis-Lemus, R. Bensch, T. Brox, J. Stegmaier, R. Mikut, S. Wolf, F. A. Hamprecht, T. Esteves, P. Quelhas, Ö. Demirel, L. Malmström, F. Jug, P. Tomancak, E. Meijering, A. Muñoz-Barrutia, M. Kozubek, and C. Ortiz-de-Solorzano. "An objective comparison of cell-tracking algorithms." In: *Nature Methods* 14.12 (Oct. 2017), pp. 1141–1152.

[78] A. Theorell, J. Seiffarth, A. Grünberger, and K. Nöh. "When a single lineage is not enough: Uncertainty-Aware Tracking for spatio-temporal live-cell image analysis." In: *Bioinformatics* 35.7 (Sept. 2018). Ed. by R. Murphy, pp. 1221–1228.

[79] M. T. Cazzolato, A. J. M. Traina, and K. Böhm. "Efficient and Reliable Estimation of Cell Positions." In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, Oct. 2018.

[80] D. Stadler and J. Beyerer. "Improving Multiple Pedestrian Tracking by Track Management and Occlusion Handling." In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 10953–10962.

[81] L. K. Harris and J. A. Theriot. "Surface Area to Volume Ratio: A Natural Variable for Bacterial Morphogenesis." In: *Trends in Microbiology* 26.10 (Oct. 2018), pp. 815–832.

[82] D. Lindemann, C. Westerwalbesloh, D. Kohlheyer, A. Grünberger, and E. von Lieres. "Microbial single-cell growth response at defined carbon limiting conditions." In: *RSC Advances* 9.25 (2019), pp. 14040–14050.

[83] W. Sattley and M. Madigan. "Microbiology." In: *Encyclopedia of Life Sciences (eLS)* (Aug. 2015).

[84] M. Maška, V. Ulman, P. Delgado-Rodriguez, E. Gómez-de-Mariscal, T. Nečasová, F. A. G. Peña, T. I. Ren, E. M. Meyerowitz, T. Scherr, K. Löffler, R. Mikut, T. Guo, Y. Wang, J. P. Allebach, R. Bao, N. M. Al-Shakarji, G. Rahmon, I. E. Toubal, K. Palaniappan, F. Lux, P. Matula, K. Sugawara, K. E. G. Magnusson, L. Aho, A. R. Cohen, A. Arbelle, T. Ben-Haim, T. R. Raviv, F. Isensee, P. F. Jäger, K. H. Maier-Hein, Y. Zhu, C. Ederra, A. Urbiola, E. Meijering, A. Cunha, A. Muñoz-Barrutia, M. Kozubek, and C. Ortiz-de-Solórzano. "The Cell Tracking Challenge: 10 years of objective benchmarking." In: *Nature Methods* 20.7 (May 2023), pp. 1010–1020.

[85] H. Irshad, L. Montaser-Kouhsari, G. Waltz, O. Bucur, J. Nowak, F. Dong, N. Knoblauch, and A. Beck. "Crowdsourcing image annotation for nucleus detection and segmentation in computational pathology: evaluating experts, automated methods, and the crowd." In: *Biocomputing 2015*. World Scientific, Nov. 2014.

[86] C. C. Sachs, K. Ruzaeva, J. Seiffarth, W. Wiechert, B. Berkels, and K. Nöh. "CellSium: versatile cell simulator for microcolony ground truth generation." In: *Bioinformatics Advances* 2.1 (Jan. 2022). Ed. by A. Bateman.

[87] M. Leygeber, D. Lindemann, C. C. Sachs, E. Kaganovitch, W. Wiechert, K. Nöh, and D. Kohlheyer. "Analyzing Microbial Population Heterogeneity–Expanding the Toolbox of Microfluidic Single-Cell Cultivations." In: *Journal of Molecular Biology* 431.23 (2019), pp. 4569–4588.

[88] T. S. Ray. "Landmark Eigenshape Analysis: Homologous Contours: Leaf Shape in Syngonium (Araceae)." In: *American Journal of Botany* 79.1 (Jan. 1992), p. 69.

[89] R. Davies, C. Twining, T. Cootes, J. Waterton, and C. Taylor. "A minimum description length approach to statistical shape modeling." In: *IEEE Transactions on Medical Imaging* 21.5 (May 2002), pp. 525–537.

[90] F. P. Kuhl and C. R. Giardina. "Elliptic Fourier features of a closed contour." In: *Computer Graphics and Image Processing* 18.3 (Mar. 1982), pp. 236–258.

[91] L. Sakamoto, H. Kajiya-Kanegae, K. Noshita, H. Takanashi, M. Kobayashi, T. Kudo, K. Yano, T. Tokunaga, N. Tsutsumi, and H. Iwata. "Comparison of shape quantification methods for genomic prediction, and genome-wide association study of sorghum seed morphology." In: *PLOS ONE* 14.11 (Nov. 2019). Ed. by S. Mamidi, e0224695.

[92] B. Berkels, G. Linkmann, and M. Rumpf. "An SL(2) Invariant Shape Median." In: *Journal of Mathematical Imaging and Vision* 37.2 (Feb. 2010), pp. 85–97.

[93] M. Rumpf and B. Wirth. "An Elasticity-Based Covariance Analysis of Shapes." In: *International Journal of Computer Vision* 92.3 (June 2010), pp. 281–295.

[94] Z. Yaniv, B. C. Lowekamp, H. J. Johnson, and R. Beare. "SimpleITK Image-Analysis Notebooks: a Collaborative Environment for Education and Reproducible Research." In: *Journal of Digital Imaging* 31.3 (Nov. 2017), pp. 290–303.

[95] S. Warfield, K. Zou, and W. Wells. "Simultaneous Truth and Performance Level Estimation (STAPLE): An Algorithm for the Validation of Image Segmentation." In: *IEEE Transactions on Medical Imaging* 23.7 (July 2004), pp. 903–921.

[96] G. Hansen, I. Herburt, H. Martini, and M. Moszyńska. "Starshaped sets." In: *Aequationes mathematicae* 94.6 (May 2020), pp. 1001–1092.

[97] H. Feldmann. *Yeast : molecular and cell biology*. Weinheim: Wiley-VCH, 2010.

[98] M. Weigert, U. Schmidt, R. Haase, K. Sugawara, and G. Myers. "Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy." In: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Mar. 2020.

[99] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers. "Cell Detection with Star-Convex Polygons." In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Springer International Publishing, 2018, pp. 265–273.

[100] G. Facchetti, B. Knapp, F. Chang, and M. Howard. "Reassessment of the Basis of Cell Size Control Based on Analysis of Cell-to-Cell Variability." In: *Biophysical Journal* 117.9 (Nov. 2019), pp. 1728–1738.

[101] G. Bradski. "The OpenCV Library." In: *Dr. Dobb's Journal of Software Tools* (2000).

[102] B. Johnston and P. de Chazal. "A review of image-based automatic facial landmark identification techniques." In: *EURASIP Journal on Image and Video Processing* 2018.1 (Sept. 2018).

[103] H. Jeckel and K. Drescher. "Advances and opportunities in image analysis of bacterial cells and communities." In: *FEMS Microbiology Reviews* 45.4 (Nov. 2020).

[104] M. Gutiérrez, P. Gregorio-Godoy, G. Pérez Del Pulgar, L. E. Muñoz, S. Sáez, and A. Rodríguez-Patón. "A new improved and extended version of the multicell bacterial simulator gro." In: *ACS Synthetic Biology* 6.8 (May 2017), pp. 1496–1508.

[105] A. Lehmussola, P. Ruusuvuori, J. Selinummi, H. Huttunen, and O. Yli-Harja. "Computational Framework for Simulating Fluorescence Microscope Images With Cell Populations." In: *IEEE Transactions on Medical Imaging* 26.7 (July 2007), pp. 1010–1016.

[106] D. Svoboda and V. Ulman. "MitoGen: A Framework for Generating 3D Synthetic Time-Lapse Sequences of Cell Populations in Fluorescence Microscopy." In: *IEEE Transactions on Medical Imaging* 36.1 (Jan. 2017), pp. 310–321.

[107] S. Taheri-Araghi, S. Bradde, J. T. Sauls, N. S. Hill, P. A. Levin, J. Paulsson, M. Vergassola, and S. Jun. "Cell-Size Control and Homeostasis in Bacteria." In: *Current Biology* 25.3 (Feb. 2015), pp. 385–391.

[108] J.-Y. Tinevez, N. Perry, J. Schindelin, G. M. Hoopes, G. D. Reynolds, E. Laplantine, S. Y. Bednarek, S. L. Shorte, and K. W. Eliceiri. "TrackMate: An open and extensible platform for single-particle tracking." In: *Methods* 115 (Feb. 2017), pp. 80–90.

[109] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. "MMDetection: Open MMLab Detection Toolbox and Benchmark." In: (June 2019). arXiv: `1906.07155 [cs.CV]`.

[110] G. Jocher, A. Stoken, J. Borovec, NanoCode012, Ayush Chaurasia, TaoXie, L. Changyu, Abhiram V, Laughing, Tkianai, YxNONG, A. Hogan, Lorenzomammana, AlexWang1900, J. Hajek, L. Diaconu, , Marc, Yonghye Kwon, , Oleg, Wanghaoyang0106, Y. Defretin, A. Lohia, Ml5ah, B. Milanko, B. Fineran, D. Khromov, D. Yiwei, , Doug, Durgesh, and F. Ingham. *ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations*. 2021.

[111] I. Arganda-Carreras, V. Kaynig, C. Rueden, K. W. Eliceiri, J. Schindelin, A. Cardona, and H. S. Seung. "Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification." In: *Bioinformatics* 33.15 (Mar. 2017). Ed. by R. Murphy, pp. 2424–2426.

[112] A. Radford, L. Metz, and S. Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks." In: *CoRR* abs/1511.06434 (2015).

[113] J. Singh, S. Gould, and L. Zheng. "High-Fidelity Guided Image Synthesis with Latent Diffusion Models." In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 5997–6006.

[114] C. Dusny and A. Grünberger. "Microfluidic single-cell analysis in biotechnology: from monitoring towards understanding." In: *Current Opinion in Biotechnology* 63 (June 2020), pp. 26–33.

[115] V. Ortseifen, M. Viefhues, L. Wobbe, and A. Grünberger. "Microfluidics for Biotechnology: Bridging Gaps to Foster Microfluidic Applications." In: *Frontiers in Bioengineering and Biotechnology* 8 (2020).

[116] Y. N. Law, H. K. Lee, C. Liu, and A. M. Yip. "A Variational Model for Segmentation of Overlapping Objects With Additive Intensity Value." In: *IEEE Transactions on Image Processing* 20.6 (June 2011), pp. 1495–1503.

[117] R. Delgado-Gonzalo and M. Unser. "Spline-based framework for interactive segmentation in biomedical imaging." In: *IRBM* 34.3 (June 2013), pp. 235–243.

[118] A. Grünberger, J. van Ooyen, N. Paczia, P. Rohe, G. Schiendzielorz, L. Eggeling, W. Wiechert, D. Kohlheyer, and S. Noack. "Beyond growth rate 0.6: Corynebacterium glutamicum cultivated in highly diluted environments." In: *Biotechnology and Bioengineering* 110.1 (Aug. 2012), pp. 220–228.

[119] R. Delgado-Gonzalo, P. Thevenaz, C. S. Seelamantula, and M. Unser. "Snakes With an Ellipse-Reproducing Property." In: *IEEE Transactions on Image Processing* 21.3 (Mar. 2012), pp. 1258–1271.

[120] M. Roberts, K. Chen, and K. L. Irion. "A Convex Geodesic Selective Model for Image Segmentation." In: *Journal of Mathematical Imaging and Vision* 61.4 (Nov. 2018), pp. 482–503.

[121] G. Wang, M. A. Zuluaga, W. Li, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin, and T. Vercauteren. "DeepIGeoS: A Deep Interactive Geodesic Framework for Medical Image Segmentation." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.7 (July 2019), pp. 1559–1572.

[122] P. Brigger, J. Hoeg, and M. Unser. "B-spline snakes: a flexible tool for parametric contour detection." In: *IEEE Transactions on Image Processing* 9.9 (2000), pp. 1484–1496.

[123] S. Mandal and V. Uhlmann. *A Learning-Based Formulation of Parametric Curve Fitting for Bioimage Analysis*. Tech. rep. bioRxiv, Jan. 2020.

[124] C. D. Carl de Boor. *A Practical Guide to Splines*. Springer-Verlag GmbH, Nov. 29, 2001.

[125] J. H. Tan and U. R. Acharya. "Active spline model: A shape based model—interactive segmentation." In: *Digital Signal Processing* 35 (Dec. 2014), pp. 64–74.

[126] J. Messelink, F. Meyer, M. Bramkamp, and C. P. Broedersz. "Single-cell growth inference of Corynebacterium glutamicum reveals asymptotically linear growth." In: *bioRxiv* (May 2020).

[127] M. J. D. Powell. "Direct search algorithms for optimization calculations." In: *Acta Numerica* 7 (Jan. 1998), pp. 287–336.

[128] P. Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." In: *Nature Methods* 17 (2020), pp. 261–272.

[129] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick. "Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016. arXiv: `1512.04143 [cs.CV]`.

[130] A. Shakarami, M. B. Menhaj, A. Mahdavi-Hormat, and H. Tarrah. "A fast and yet efficient YOLOv3 for blood cell detection." In: *Biomedical Signal Processing and Control* 66 (Apr. 2021), p. 102495.

[131] D. Waithe, J. M. Brown, K. Reglinski, I. Diez-Sevilla, D. Roberts, and C. Eggeling. "Object detection networks and augmented reality for cellular detection in fluorescence microscopy." In: *Journal of Cell Biology* 219.10 (Aug. 2020).

[132] J. Bergstra, D. Yamins, and D. D. Cox. "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures." In: *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*. Vol. 28. JMLR Workshop and Conference Proceedings. JMLR.org, 2013, pp. 115–123.

[133] S. Fujita and X.-H. Han. "Cell Detection and Segmentation in Microscopy Images with Improved Mask R-CNN." In: *Computer Vision – ACCV 2020 Workshops*. Springer International Publishing, 2021, pp. 58–70.

[134] R. Kloss et al. "Tailor-made catalytically active inclusion bodies for different applications in biocatalysis." In: *Catalysis Science & Technology* 8.22 (2018), pp. 5816–5826.

[135] V. D. Jäger et al. "Catalytically-active inclusion bodies for biotechnology—general concepts, optimization, and application." In: *Applied Microbiology and Biotechnology* 104.17 (July 2020), pp. 7313–7329.

[136] K. Küsters et al. "Construction and comprehensive characterization of an EcLDCc-CatIB set-varying linkers and aggregation inducing tags." In: *Microbial Cell Factories* 20.1 (Feb. 2021).

[137] A. Bochkovskiy, C. Wang, and H. M. Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection." In: *CoRR* abs/2004.10934 (2020). arXiv: `2004.10934`.

[138] K. Ruzaeva, K. Nöh, and B. Berkels. "A hybrid multi-object segmentation framework with model-based B-splines for microbial single cell analysis." In: *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*. (in press). IEEE, Mar. 2022.

[139] M. Maska, O. Danek, S. Garasa, A. Rouzaut, A. Munoz-Barrutia, and C. Ortiz-de-Solorzano. "Segmentation and Shape Tracking of Whole Fluorescent Cells Based on the Chan–Vese Model." In: *IEEE Transactions on Medical Imaging* 32.6 (June 2013), pp. 995–1006.

[140] O. Dzyubachyk, W. van Cappellen, J. Essers, W. Niessen, and E. Meijering. "Advanced Level-Set-Based Cell Tracking in Time-Lapse Fluorescence Microscopy." In: *IEEE Transactions on Medical Imaging* 29.3 (Mar. 2010), pp. 852–867.

[141] A. Arbelle, J. Reyes, J.-Y. Chen, G. Lahav, and T. R. Raviv. "A probabilistic approach to joint cell tracking and segmentation in high-throughput microscopy videos." In: *Medical Image Analysis* 47 (July 2018), pp. 140–152.

[142] K. J. Cutler, C. Stringer, T. W. Lo, L. Rappez, N. Stroustrup, S. B. Peterson, P. A. Wiggins, and J. D. Mougous. "Omnipose: a high-precision morphology-independent solution for bacterial cell segmentation." In: *Nature Methods* 19.11 (Oct. 2022), pp. 1438–1448.

[143] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu. "Cellpose: a generalist algorithm for cellular segmentation." In: *Nature Methods* 18.1 (Dec. 2020), pp. 100–106.

[144] V. Ulman et al. "An objective comparison of cell-tracking algorithms." In: *Nature Methods* 14.12 (Oct. 2017), pp. 1141–1152.

[145] A. P. Cuny, A. Ponti, T. Kündig, F. Rudolf, and J. Stelling. "Cell region fingerprints enable highly precise single-cell tracking and lineage reconstruction." In: *bioRxiv* (Oct. 2021).

[146] J.-Y. Tinevez et al. "TrackMate: An open and extensible platform for single-particle tracking." In: *Methods* 115 (Feb. 2017), pp. 80–90.

[147] D. H. Theriault, M. L. Walker, J. Y. Wong, and M. Betke. "Cell morphology classification and clutter mitigation in phase-contrast microscopy images using machine learning." In: *Machine Vision and Applications* 23.4 (June 2011), pp. 659–673.

[148] A. D. Balomenos, P. Tsakanikas, and E. S. Manolakos. "Tracking single-cells in overcrowded bacterial colonies." In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, Aug. 2015.

[149] S. Sarmadi, J. J. Winkle, R. N. Alnahhas, M. R. Bennett, K. Josić, A. Mang, and R. Azencott. "Stochastic Neural Networks for Automatic Cell Tracking in Microscopy Image Sequences of Bacterial Colonies." In: *Mathematical and Computational Applications* 27.2 (Mar. 2022), p. 22.

[150] K. Löffler, T. Scherr, and R. Mikut. "A graph-based cell tracking algorithm with few manually tunable parameters and automated segmentation error correction." In: *PLOS ONE* 16.9 (Sept. 2021). Ed. by K. Metze, e0249257.

[151] S. Schito et al. "Communities of Niche-optimized Strains (CoNoS) – Design and creation of stable, genome-reduced co-cultures." In: *Metabolic Engineering* 73 (Sept. 2022), pp. 91–103.

[152] K. Löffler and R. Mikut. *EmbedTrack – Simultaneous Cell Segmentation and Tracking Through Learning Offsets and Clustering Bandwidths*. 2022.

[153] J.-B. Lugagne, H. Lin, and M. J. Dunlop. "DeLTA: Automated cell segmentation, tracking, and lineage reconstruction using deep learning." In: *PLOS Computational Biology* 16.4 (Apr. 2020). Ed. by A. R. Asthagiri, e1007673.

[154] N. Mosheiff et al. "Inheritance of Cell-Cycle Duration in the Presence of Periodic Forcing." In: *Phys. Rev. X* 8 (2 May 2018), p. 021035.

[155] D. F. Crouse. "On implementing 2D rectangular assignment algorithms." In: *IEEE Transactions on Aerospace and Electronic Systems* 52.4 (Aug. 2016), pp. 1679–1696.

[156] P. Matula, M. Maška, D. V. Sorokin, P. Matula, C. Ortiz-de-Solórzano, and M. Kozubek. "Cell Tracking Accuracy Measurement Based on Comparison of Acyclic Oriented Graphs." In: *PLOS ONE* 10.12 (Dec. 2015). Ed. by T. Abraham, e0144959.