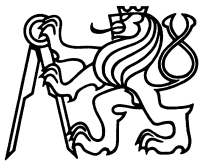
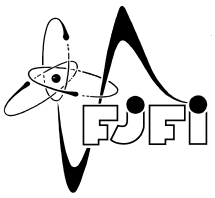


Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering



Tracking of cell processes in microscopic time-lapse images

Sledování buněčných procesů na časosběrných snímcích z mikroskopu

Master's Thesis

Author: **Bc. Adam Janich**

Supervisor: **Doc. Ing. Filip Šroubek, Ph.D., DSc.**

Consultant: **Ing. Zuzana Kadlecová, Ph.D.**

Academic year: 2022/2023

Katedra: matematiky Akademický rok: 2022/2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student:	Bc. Adam Janich
Studijní program:	Matematické inženýrství
Název práce (česky):	Sledování buněčných procesů na časosběrných snímcích z mikroskopu
Název práce (anglicky):	Tracking of cell processes in microscopic time-lapse images

Pokyny pro vypracování:

1) Seznamte se s metodami automatického sledování (trekování) objektů v biologických časosběrných datech (multi-target tracking). Nastudujte různá kritéria spojování detekcí v sousedních snímcích. Prostudujte grafové metody s lokální a globální optimalizací.
2) Použijte, v případě nutnosti upravte, Vámi vyvinuté detekční metody na bázi hlubokého učení.
3) Navrhněte metodu vhodnou pro trekování endocytických váčků na časosběrných snímcích pořízených TIRF-SIM (total internal reflection fluorescence - structured illumination microscopy).
4) Navrhněte metriku pro vyhodnocení přesnosti trekování a vyhodnoťte Vámi navrženou metodu na syntetických datech.
5) Proveďte anotaci reálných dat a statisticky vyhodnoťte navrženou metodu na anotovaných datech.

Doporučená literatura:

1) C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

2) I. Goodfellow et al., Deep Learning. MIT Press, 2016.

3) K. Jaqaman, D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S. L. Schmid, G. Danuser, Robust single-particle tracking in live-cell time-lapse sequences. Nature Methods 5(8), 2008, 695-702.

4) C. Haubold, Scalable Inference for Multi-Target Tracking of Proliferating Cells, disertační práce. Univerzita v Heidelbergu, 2017.

5) D. Ershov, M. S. Phan, J. W. Pylvänäinen et al., TrackMate 7: integrating state-of-the-art segmentation algorithms into tracking pipelines. Nature Methods 19, 2022, 829–832.

Jméno a pracoviště vedoucího diplomové práce:

Doc. Ing. Filip Šroubek, Ph.D., DSc.
Ústav teorie informace a automatizace, Pod Vodárenskou věží 4, 182 00, Praha 8

Jméno a pracoviště konzultanta:

Dr. Zuzana Kadlecová
Cambridge Institute for Medical Research, University of Cambridge, UK

Datum zadání diplomové práce: 31.10.2022

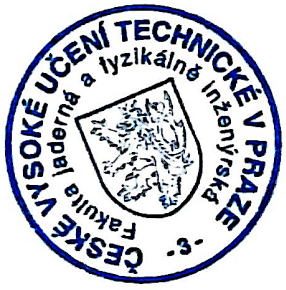
Datum odevzdání diplomové práce: 3.5.2023

Doba platnosti zadání je dva roky od data zadání.

V Praze dne 31.10.2022

.....
garant oboru

.....
vedoucí katedry



.....
děkan

Acknowledgment:
This thesis would not have been possible without the expertise and support of Doc. Ing. Filip Šroubek, Ph.D., DSc. I am thankful for his guidance and patience.

Author's declaration:
I declare that this Master's Thesis is entirely my own work and I have listed all the used sources in the bibliography.

Prague, January 8, 2024 Bc. Adam Janich

Název práce:

Sledování buněčných procesů na časosběrných snímcích z mikroskopu

Autor: Bc. Adam Janich

Program: Matematické inženýrství

Druh práce: Diplomová práce

Vedoucí práce: Doc. Ing. Filip Šroubek, Ph.D., DSc.,
Ústav teorie informace a automatizace AV ČR, Pod Vodárenskou věží 4, 182 00 Praha

Konzultant: Ing. Zuzana Kadlecová, Ph.D.,
Cambridge Institute for Medical Research, University of Cambridge, UK

Abstrakt: Tato práce se zabývá buněčnou endocytózou, se zaměřením na clathrinové váčky (CCPs) za použití pokročilé mikroskopie Total Internal Reflection Fluorescence - Structured Illumination Microscopy (TIRF-SIM). Představuje detekční algoritmus založený na architektuře U-Net a adaptuje algoritmus Multiple Hypothesis Tracking (MHT) pro identifikaci a sledování CCPs v časosběrných snímcích TIRF-SIM. Přesnost těchto výpočetních nástrojů je hodnocena proti anotacím odborníků, čímž potvrzuje jejich účinnost a identifikuje oblasti pro zlepšení. Tato práce přispívá k pochopení chování CCPs, kombinuje vysoce rozlišovací zobrazování s výpočetní analýzou a má širší dopad na výzkum buněčné biologie.

Klíčová slova: Buněčná biologie, Endocytoza, Hluboké učení, Sledování více objektů, TIRF-SIM, Zpracování obrazu

Title:

Tracking of cell processes in microscopic time-lapse images

Author: Bc. Adam Janich

Abstract: This thesis delves into cellular endocytosis, focusing on clathrin-coated pits (CCPs) using advanced Total Internal Reflection Fluorescence - Structured Illumination Microscopy (TIRF-SIM). It presents a U-Net-based detection algorithm and a customized Multiple Hypothesis Tracking (MHT) algorithm for identifying and tracking CCPs in TIRF-SIM images. The performance of these computational tools is evaluated against human expert annotations, validating their effectiveness and identifying areas for improvement. This work contributes to the understanding of CCP behavior, blending high-resolution imaging with computational analysis, and has broader implications for cellular biology research.

Key words: Cellular Biology, Deep Learning, Endocytosis, Image Processing, Multiple Object Tracking, TIRF-SIM

Contents

Introduction **11**

1 Cellular Endocytosis and Imaging of Clathrin-Coated Pits **13**
1.1 Chapter Overview . . . 13
1.2 Cellular Endocytosis . . . 14
1.3 Receptor-Mediated Endocytosis . . . 15
1.4 Imaging Clathring-Coated Pits . . . 16
1.4.1 TIRF Technology . . . 16
1.4.2 TIRF-SIM Technology . . . 18
1.5 Chapter Conclusion . . . 19

2 Multiple Object Tracking **21**
2.1 Introduction to Multiple Object Tracking . . . 21
2.2 Problem Formulation . . . 22
2.3 Network Flow-based Algorithms . . . 23
2.3.1 Description of the Algorithm . . . 23
2.3.2 Discussion of The Network Flow-based Algorithm . . . 26
2.4 Multiple Hypothesis Tracking . . . 26
2.4.1 Introduction to Multiple Hypothesis tracking . . . 26
2.4.2 Description of the MHT algorithm . . . 28
2.5 Tracking Performance Metrics . . . 36
2.5.1 Introduction to Tracking Performance Metrics . . . 36
2.5.2 Multiple Object Tracking Precision . . . 36
2.5.3 Root Mean Square Error . . . 37
2.5.4 Multiple Object Tracking Accuracy . . . 38

3 Data Acquisition and Preparation **39**
3.1 Chapter Overview . . . 39
3.2 Cell Type, Lineage and Cultivation . . . 40
3.3 Cell Imaging Techniques and TIRF-SIM Microscopy Setup . . . 42
3.4 Data Format . . . 43
3.5 Data Normalization . . . 43
3.6 Data Annotation . . . 45

4 Clathrin-Coated Pits Detection Algorithm **47**
4.1 Detection Algorithm Overview 47
4.2 Target Data Preparation 48
4.3 Neural Network Architecture 50
4.4 Data Loading and Augmentation 54
4.5 Training the Detector 56
4.6 Detector Evaluation and Parameter Finetuning 58

5 Clathrin-Coated Pits Tracking Algorithm **63**
5.1 Chapter Overview 63
5.2 Measuring Characteristics of CCP Motion 64
5.3 Selecting Tracking Evaluation Metrics 65
5.4 Algorithm Description 68
5.5 Runtime Analysis of the Tracking Algorithm 71
5.6 Results 72

Conclusion **77**

Introduction

This thesis explores the process of endocytosis in cells, with a special focus on structures called clathrin-coated pits (CCPs). Endocytosis is how cells take in molecules from outside their membrane, and CCPs play a crucial role in this process by selecting specific molecules to absorb. Understanding CCPs is important for cell biology and can have implications for medical research. To study CCPs in detail, this thesis uses a high-tech imaging method called Total Internal Reflection Fluorescence - Structured Illumination Microscopy (TIRF-SIM). TIRF-SIM allows for clearer and more detailed images than traditional microscopy methods, helping us to better understand how CCPs function in cells.

In the realm of TIRF-SIM, high-quality imaging is crucial for observing CCPs at the cellular membrane. However, the challenge arises in the analysis of the complex data generated by this method. To address this, the thesis proposes computational tools for data interpretation. A detection algorithm, utilizing U-Net deep learning, has been developed to pinpoint CCP locations in the images. Furthermore, the Multiple Hypothesis Tracking (MHT) method has been adapted to track CCP movements and interactions.

The effectiveness of the tracking algorithm is measured using a newly introduced evaluation metric. This metric is compared with the accuracy level of human experts, offering a comprehensive assessment of the algorithm's performance. The analysis not only validates the methods but also identifies areas for potential enhancement.

Overall, the research enhances the study of CCPs, merging sophisticated imaging with in-depth computer analysis.

Subsequent chapters will further develop these themes, detailing data acquisition processes, addressing the complexities of CCP detection and tracking, and evaluating these methods against expertly annotated datasets.

Chapter 1 provides a comprehensive examination of cellular endocytosis, focusing particularly on clathrin-coated pits (CCPs). It explains the essential role of the cellular membrane in endocytosis and delves into the specifics of receptor-mediated endocytosis, emphasizing how clathrin is crucial in the formation of CCPs for the selective intake of substances by cells.

The chapter also discusses the challenges in imaging CCPs, highlighting the limitations of traditional methods such as electron microscopy, especially in the context of live cell studies. To address these imaging complexities, the chapter describes the use of Total Internal Reflection Fluorescence (TIRF) microscopy. This advanced imaging technique offers superior resolution and minimal background noise, making it particularly effective for studying cell membrane processes. The discussion extends to TIRF-SIM, a method that merges TIRF with Structured Illumination Microscopy, enhancing lateral resolution and providing a more detailed view of CCP activity.

Chapter 2 explores Multiple Object Tracking (MOT), an important area in computer vision with wide-ranging applications, including surveillance and biological research. The chapter introduces MOT and its role in tracking diverse objects, such as vehicles, people, and for the purposes of this thesis, biological entities like cells and viral particles.

The chapter highlights the challenges in MOT, such as accurately identifying and consistently tracking objects over multiple video frames. It discusses the difficulties in dealing with false negatives, where objects are missed, and false positives, where non-objects are incorrectly identified as objects. Understanding these issues is key to grasping the complexities of tracking algorithms.

Further, the chapter examines the formal structure of the MOT problem and reviews different strategies and methods developed to address its challenges. It also looks at performance metrics in MOT, which are essential for assessing and comparing the effectiveness of various tracking methods. This chapter not only covers the technical side of MOT but also prepares the ground for its application in tracking clathrin-coated pits in time-lapse biological imaging.

Chapter 3 outlines the critical process of data acquisition and preparation for TIRF-SIM imaging, essential for the study of clathrin-coated pits. It covers the selection and cultivation of specific cell types, detailing the TIRF-SIM setup and data acquisition techniques to ensure high-quality imaging. The chapter also discusses the organization and normalization of the microscopy images, crucial for the effective training of the detection and tracking algorithms. Concluding with the expert annotation of CCPs, this chapter establishes the ground truth for evaluating the performance of the developed algorithms, setting the stage for subsequent discussions on algorithm development and validation.

Chapter 4 introduces the detection algorithm for clathrin-coated pits using TIRF-SIM images, centered around a U-Net architecture trained for per-pixel regression. The chapter details how this method identifies CCPs through local maxima in spatially continuous target images, utilizing regularization techniques like dropout and weight decay to prevent overfitting due to the dataset's limited size. The detector's validation involves a cross-validation regime and fine-tuning of key parameters, such as the number of temporal frames and the detection threshold, to optimize performance and enhance the model's accuracy in identifying CCPs.

Chapter 5 of the thesis delves into the development and evaluation of a tracking algorithm for clathrin-coated pits observed in TIRF-SIM microscopy images. It begins by analyzing the motion characteristics of CCPs, determining whether they follow Brownian motion or display more complex patterns. This analysis is crucial in tailoring the multiple hypothesis tracking (MHT) algorithm for the unique challenges of CCP tracking. The chapter also introduces novel metrics, alongside established ones like MOTA and RMSE, for a comprehensive evaluation of the algorithm's performance. A detailed description of the algorithm's functionality is provided, followed by an analysis of its runtime, emphasizing scalability. Finally, the results section presents an assessment of the algorithm against human annotators, highlighting its efficacy in tracking CCP dynamics.

Chapter 1

Cellular Endocytosis and Imaging of Clathrin-Coated Pits

1.1 Chapter Overview

This chapter offers a look into cellular endocytosis and the advanced imaging techniques used to study clathrin-coated pits (CCPs). It begins with an overview of cellular endocytosis, emphasizing the dynamic nature of the cellular membrane and its importance in cell functionality. The focus then shifts to receptor-mediated endocytosis, highlighting the role of clathrin in forming CCPs and the selective intake of substances by cells.

The latter part of the chapter addresses the challenges in imaging CCPs, discussing the limitations of traditional electron microscopy for live-cell imaging. It introduces TIRF (Total Internal Reflection Fluorescence) microscopy as a solution, a technique that offers high-resolution imaging of the cell membrane with minimal background interference. The chapter concludes with an exploration of TIRF-SIM (Structured Illumination Microscopy integrated with TIRF), an advanced method that overcomes the lateral resolution limitations of TIRF. This section underscores the importance of TIRF-SIM in enhancing our understanding of cellular processes through improved imaging capabilities.

Overall, the chapter underscores the critical role of sophisticated imaging technologies like TIRF and TIRF-SIM in advancing our understanding of cellular endocytosis and the detailed study of CCPs.

Parts of this chapter are taken from [1]. An excellent resource on cellular endocytosis is [2].

1.2 Cellular Endocytosis

At the core of every living cell is the cellular membrane, a boundary that separates the cell's internal components, such as cytoplasm and organelles, from the external environment. This membrane is chemically composed of a phospholipid bilayer, which is further enriched with an array of proteins, glycoproteins, glycolipids, and cholesterol. These elements collectively contribute to the membrane's functional diversity. Unlike being rigid or static, the cellular membrane is dynamic; the phospholipids within it are capable of rearrangement and movement, granting the membrane notable elasticity and flexibility.

Critical to a cell's survival is the continuous exchange of substances with its surrounding environment. This exchange is facilitated through two primary mechanisms: passive and active transport. Passive transport, which involves the movement of substances along natural concentration gradients, does not require external energy sources. It encompasses processes such as diffusion and osmosis. On the other hand, active transport requires the expenditure of chemical energy, typically derived from ATP molecules, to move substances against these gradients. Active transport can occur in two main ways: either through specialized proteins embedded in the cellular membrane or via a process known as *endocytosis*, where the cell ingests chemical substances and larger particles.

Endocytosis is characterized by an inward folding of the cellular membrane, leading to the formation of an *endocytic vesicle*. This vesicle encapsulates external substances, encompassing both dissolved materials and solid particles from the cell's immediate environment. Once the vesicle is formed, it is pinched off from the membrane and internalized into the cell. This results in a phospholipid bilayer-enclosed sphere, known as an endocytic vesicle, containing materials from outside the cell. The cell then transports these vesicles internally through a series of complex processes.

Endocytosis, a crucial cellular process, encompasses three primary types, each with its unique function and mechanism. These types are illustrated in Figure 1.1 and are as follows:

- **Phagocytosis:** Characterized by the cell's ability to engulf larger solid particles, such as microorganisms or their fragments. This process involves the extension of the cell membrane to form pseudopods, which encircle and engulf the target particles.
- **Pinocytosis:** This type of endocytosis involves the cell in non-selectively ingesting extracellular fluid. It allows the cell to absorb dissolved nutrients and other substances, functioning much like the cell 'drinking' the surrounding fluid.
- **Receptor-mediated endocytosis:** Distinct for its specificity, this process occurs when cell surface receptors bind to particular substances. Upon binding, these substances are internalized by the cell, enabling the selective intake of specific molecules based on the receptors on the cell surface.

In the context of this work, the focus is exclusively on receptor-mediated endocytosis, as evidenced by the processing of microscopic images specific to this type. Consequently, while phagocytosis and pinocytosis are integral to understanding the broader spectrum of endocytosis, they are not the primary subjects of this thesis. Henceforth, unless specified otherwise, 'endocytosis' will refer specifically to receptor-mediated endocytosis.

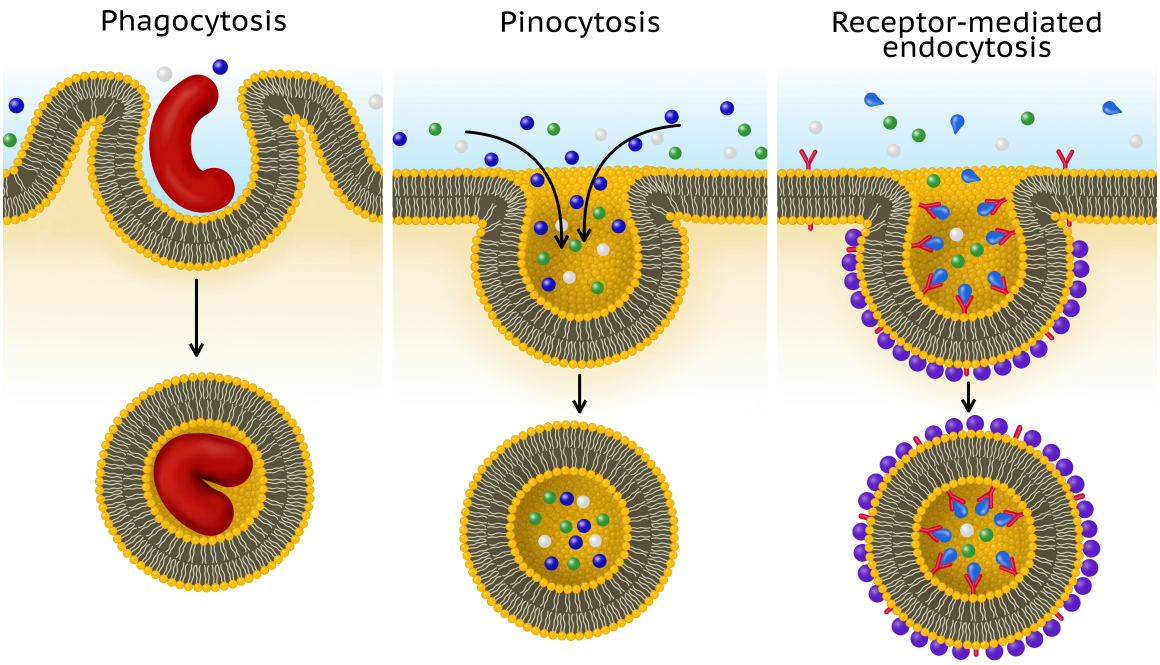


Figure 1.1: Illustration of the three main types of endocytosis: Phagocytosis for large particles, Pinocytosis for nonspecific fluid intake, and Receptor-mediated endocytosis for specific substance intake.

1.3 Receptor-Mediated Endocytosis

Receptor-mediated endocytosis, distinct from processes like pinocytosis, allows cells to specifically intake particular chemical substances. This specificity is achieved through receptors on the cell surface, which bind to certain target molecules. When these molecules attach to the receptors, it triggers the accumulation of specific proteins on the inner side of the membrane. The most crucial among these proteins is clathrin, a key component in the formation of CCPs.

As depicted on the right side of Figure 1.1, the receptors, target molecules, and clathrin are represented by different colors for clarity: receptors in red, target molecules in blue, and clathrin in purple. This illustration, however, is a simplified representation. In reality, clathrin molecules possess a unique structure comprising three spiral arms that emanate from a central point, forming a shape known as a triskelion. These triskelion structures can interconnect through their arms, creating formations with a positive curvature. This interaction leads to the inward folding of the cell membrane, forming a vesicle that internalizes the bound substances - a process central to receptor-mediated endocytosis.

Once internalized, these vesicles remain encased within a framework of clathrin, forming what is known as a clathrin cage, as shown in Figure 1.2. This figure highlights the unique triskelion structure of clathrin molecules, each represented in a different color for visual distinction.

While endocytosis may appear straightforward at first glance, its underlying mechanisms are complex and intricate. Discovered in the 19th century, many facets of endocytosis, particularly the role and regulation of CCPs, are still subjects of active research. Understanding how cells control and regulate endocytosis, a topic well explored in modern molecular biology and summarized in sources like [5], is crucial for advancing various biological fields.

The study of CCPs and their regulation has significant implications. Cells use endocytosis to intake not only nutrients and signaling molecules but also certain synthetic drugs. Additionally, some viruses,

Figure 1.2: The structure of a clathrin cage surrounding an endocytic vesicle. Distinctly colored clathrin molecules highlight the triskelion structure.

including SARS-CoV-2, exploit this process to invade cells. A better understanding of how cellular endocytosis, specifically through CCPs, can be regulated, holds potential for advancements in cytology, virology, oncology, immunology, and other areas of molecular biology and medicine.

1.4 Imaging Clathring-Coated Pits

In the realm of cellular biology, particularly when studying dynamic processes such as those involving CCPs, the choice of imaging technique is paramount. The need for real-time observation of CCPs in their native cellular environment poses a significant challenge. Electron microscopy, while offering high-resolution static images, falls short in live-cell imaging. To effectively observe CCPs in action, a method focusing specifically on the cellular membrane, where these structures form and operate, is essential.

1.4.1 TIRF Technology

Total Internal Reflection Fluorescence (TIRF) microscopy presents an optimal solution to these requirements. This technique specializes in high-resolution imaging of events at the cell membrane with minimal background disturbance. TIRF achieves this by generating an evanescent wave that illuminates only a thin section of the sample, extending a few hundred nanometers from the cell membrane. This targeted approach is especially beneficial for studying CCPs, as it allows for the selective visualization of these pits without the interference from other cellular components.

The principle of TIRF relies on the physics of light behavior at the interface of two media with different refractive indices. When light strikes this interface from an optically denser medium (with a higher refractive index) to an optically rarer medium (with a lower refractive index), it undergoes refraction up to a certain limit, defined by the critical angle. This critical angle, θ_c, is calculated using the equation $\theta_c = \arcsin\left(\frac{n_2}{n_1}\right)$, where n_1 and n_2 are the refractive indices of the denser and rarer mediums, respectively.

Upon exceeding this critical angle, total internal reflection occurs, with the light reflecting back entirely into the denser medium. However, an interesting phenomenon happens at the interface – an evanescent wave is generated in the rarer medium. This wave, predicted by Maxwell's equations, propagates perpendicular to the interface and its intensity decays exponentially with distance from the interface.

TIRF microscopy leverages the evanescent wave generated by total internal reflection to selectively illuminate CCPs at the cell membrane. This targeted illumination is crucial for real-time visualization of CCP activities, allowing for detailed observation of their formation, maturation, and subsequent internalization. Unlike static imaging methods, TIRF offers a dynamic view into cellular processes, making it an indispensable tool for studying the complexities of receptor-mediated endocytosis.

A key component of TIRF microscopy is the use of fluorophores – substances that can absorb light energy and re-emit it at a longer wavelength. This fluorescence property is characteristic of various chemical substances, including a diverse range of proteins, peptides, organic molecules, synthetic polymers, and multi-molecular complexes [12]. In microscopy, fluorophores are used as markers, covalently attached to cellular proteins or other molecules. Their ability to be repeatedly excited by light allows for the monitoring of molecular movements over time, adding a valuable temporal dimension to imaging. This feature is particularly useful in TIRF microscopy, where the precise tracking of CCP dynamics is essential. Moreover, the distinct emission wavelengths of different fluorophores enable simultaneous observation of multiple substances within the cell, using multiple imaging channels [13].

TIRF microscopy ingeniously combines the phenomena of fluorescence and total internal reflection. In a typical setup, the cell sample is placed on a glass slide, an optically denser medium compared to the surrounding liquid and the sample itself. The glass slide is illuminated from below with laser light at an angle exceeding the critical angle for total internal reflection. This setup generates an evanescent electromagnetic wave right above the glass slide, within the sample. The evanescent field thus created excites the fluorophores present in the sample. This arrangement is schematically depicted in Figure 1.3, illustrating how TIRF microscopy efficiently utilizes both fluorescence and total internal reflection to study cellular processes at the membrane level.

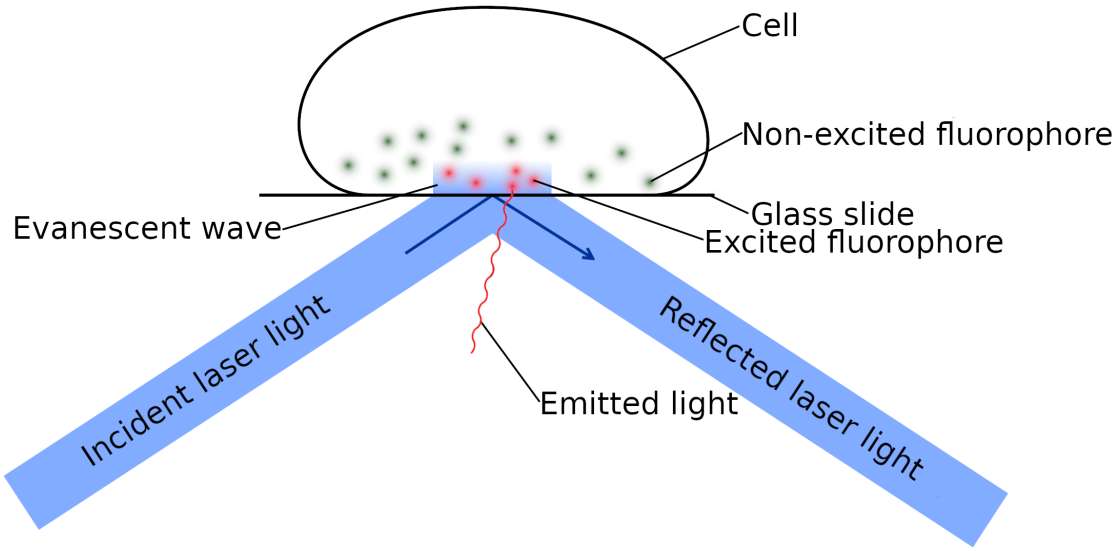


Figure 1.3: Thanks to the total reflection of laser light, an evanescent wave is created in the sample, close to the glass slide, which excites fluorophores that emit a specific wavelength of light.

In TIRF microscopy, the phenomenon of total internal reflection of laser light is key to creating an evanescent wave near the glass slide within the sample. This wave selectively excites fluorophores located close to the slide, causing them to emit light at specific wavelengths. The emitted light is then captured by the objective lens and focused using specialized filters designed to transmit only the relevant wavelengths of light. This setup results in a highly localized illumination area, with the evanescent wave typically extending to an effective thickness of about 100nm from the glass slide [15].

The advantage of such localized excitation is the remarkably high axial resolution it provides. Since the fluorophores emit light only within this narrow region, the resultant images are exceptionally clear, primarily focusing on events occurring at or near the cell membrane. This specificity effectively reduces interference from fluorescent materials located deeper within the cell. Consequently, TIRF microscopy is particularly well-suited for studying processes like cellular endocytosis, which predominantly occur at the cell membrane. This application of TIRF technology allows for detailed observation and analysis of endocytic processes, crucial for advancing our understanding of cellular function and interaction.

1.4.2 TIRF-SIM Technology

TIRF-SIM (Total Internal Reflection Fluorescence Structured Illumination Microscopy) represents a significant advancement in overcoming the lateral resolution limitations inherent in traditional TIRF microscopy. While TIRF offers excellent axial resolution, particularly for imaging processes at the cell membrane, its lateral resolution is fundamentally constrained by the diffraction limit of light, typically around 200-300nm when using visible light wavelengths [11].

The quest to transcend this diffraction barrier led to the development of super-resolution microscopy techniques, among which SIM (Structured Illumination Microscopy) is a pivotal innovation. Recognized for its groundbreaking contributions to the field of microscopy, super-resolution techniques were awarded the 2014 Nobel Prize in Chemistry. SIM specifically enhances lateral resolution by employing spatially varying light intensities, typically generated through optical elements like diffraction gratings. This technique involves illuminating the sample with a series of different patterns, allowing the reconstruction of a higher-resolution image through complex computational algorithms.

Integrating SIM with TIRF microscopy has proven highly effective. This combination, known as TIRF-SIM, has been shown to approximately double the lateral resolution compared to traditional TIRF microscopy [11, 8]. Under certain specific conditions, it is even possible to achieve resolutions better than 50nm [9].

For super-resolution imaging in TIRF-SIM, nine raw images are captured at each time point. These images include three different angles and three phases of structured illumination. The processing and reconstruction of these images into a high-resolution composite are achieved through a specialized TIRF-SIM reconstruction algorithm, as described in [51]. This algorithm plays a crucial role in maximizing the resolution enhancement provided by TIRF-SIM.

The application of TIRF and TIRF-SIM in studying cellular endocytosis involves attaching fluorophores to clathrin and other related proteins. The cell, placed on a TIRF microscope slide, allows for detailed observation of these proteins at the cell membrane. Thanks to the high axial resolution of TIRF microscopy and the enhanced lateral resolution offered by TIRF-SIM, the distribution and dynamics of clathrin during the endocytic process can be studied with unprecedented detail.

In Figure 1.4, we observe the application of TIRF microscopy in visualizing clathrin distribution. The left part of the figure shows the overall distribution, with clathrin concentrated in several small areas. A more detailed examination (right part of Figure 1.4) reveals individual vesicles. However, the lateral resolution limitations of standard TIRF microscopy result in these vesicles appearing as spots, lacking a darker center at each developmental stage. TIRF-SIM technology, with its enhanced resolution capabilities, is pivotal in providing a more detailed view of these structures.

A comparison of Figure 1.4 with Figure 1.5 underscores the significant enhancement in image quality achieved through TIRF-SIM technology. Figure 1.5 displays the same cellular sample as observed using TIRF-SIM, which offers approximately double the lateral resolution compared to traditional TIRF microscopy. The left side of the figure presents the full image of the cell, where a marked improvement in sharpness is immediately noticeable. The zoomed-in view on the right more clearly reveals the structural details of the endocytic vesicles. Many vesicles are now visible as bright rings with distinct darker centers, aligning with the anticipated structural representation of clathrin in the endocytic process.

1.5 Chapter Conclusion

This chapter has delineated the pivotal role of TIRF and TIRF-SIM microscopy in the study of clathrin-coated pits and their involvement in the process of cellular endocytosis. The advancements in imaging technology, particularly the integration of structured illumination with TIRF microscopy, have significantly pushed the boundaries of resolution, enabling researchers to observe and analyze cellular processes with unprecedented clarity.

The comparison of images obtained from TIRF and TIRF-SIM microscopy vividly illustrates the enhancement in resolution and detail, providing a more profound understanding of the dynamic and intricate processes occurring at the cellular membrane. This technological progression not only allows for the visualization of cellular structures and processes with greater detail but also opens new avenues for exploring the molecular mechanisms underlying various cellular functions. As a result, TIRF-SIM microscopy stands as a cornerstone in the modern study of cellular biology, with its applications extending beyond endocytosis to numerous other areas of cell biology research.

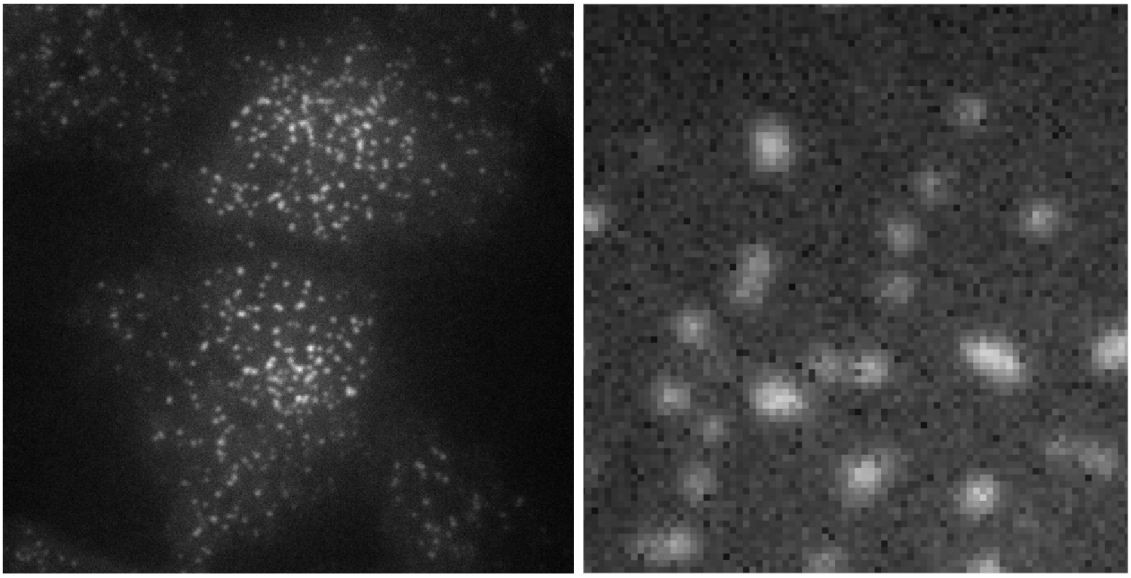


Figure 1.4: Clathrin distribution in endocytosis imaged by TIRF microscopy. The left panel shows the entire cell, and the right panel offers a zoomed-in view, where ring structures are not discernible.

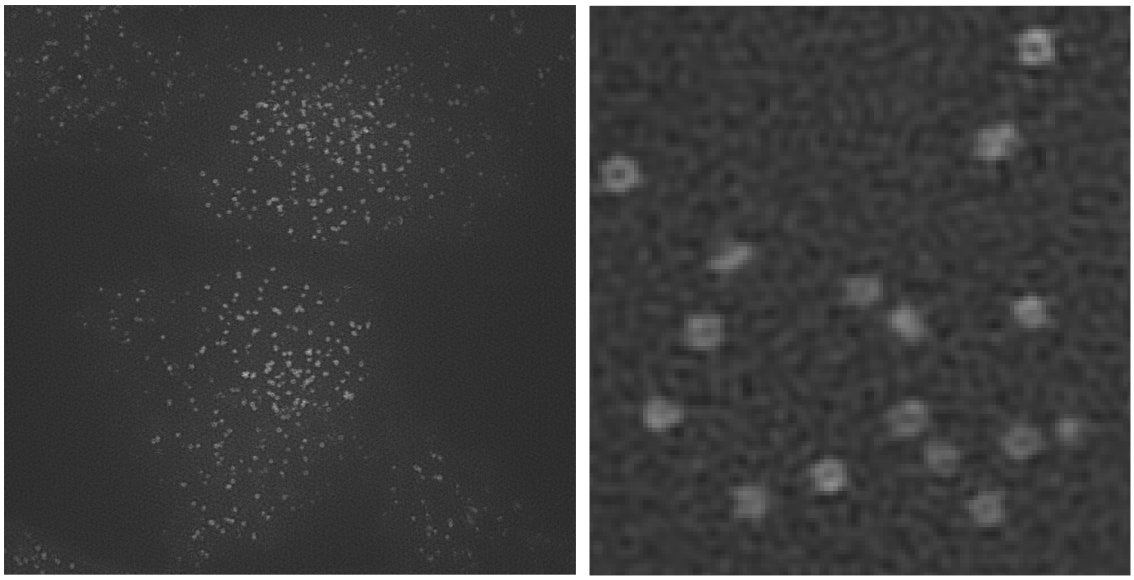


Figure 1.5: Enhanced image of clathrin distribution in endocytosis captured by TIRF-SIM microscopy. The left panel shows the entire cell with improved resolution, while the right panel provides a detailed view where the ring structures of the vesicles are clearly visible.

Chapter 2

Multiple Object Tracking

2.1 Introduction to Multiple Object Tracking

Multiple Object Tracking (MOT) is a fundamental and challenging problem in the field of computer vision. At its core, MOT involves the detection of multiple objects (such as people, vehicles, or other entities) and tracking these entities as they move through a series of frames. MOT is crucial in multiple real-world applications, including surveillance, autonomous vehicles, sports analytics, and air traffic control.

Modern applications of multiple object tracking in biology include tracking living cells [17, 18], tracking viral particles within living cells [19], and tracking extracellular vesicles [20].

The goal of MOT is not only to identify the presence of objects in each frame but also to maintain their identities and track their movements in time. In other words, data association over video frames is the main challenge of MOT. The MOT problem requires solving the following challenges:

- Dealing with occlusions where objects may be temporarily hidden.
- Handling false negatives where the detection algorithm failed.
- Handling false positives (usually called false alarms in context of MOT) where the detection algorithm falsely detected an object in the background noise.
- Accommodating for changes in object appearance.
- Dealing with unpredictable movements of the objects.
- Scaling the tracking algorithms to handle a large number of objects efficiently.

The goal of this chapter is to delve into the formal formulation of the MOT problem and introduce different strategies for tracking as well as discuss some metrics used to assess the performance of these methods.

2.2 Problem Formulation

To formulate the Multiple Object Tracking (MOT) problem, we consider $\mathcal{X} = \{\mathbf{x}_i\}$ to be the set of all observations of objects to be tracked. (Typically, each observation is one response of a detection algorithm.) We can formally define the i-th observation $\mathbf{x}_i$ to be a tuple describing the position, time frame, and other attributes of the observation. For instance:

$$\mathbf{x}_i = (x_i, t_i, s_i, a_i),$$

where x_i is the position, t_i is the time step (frame index), s_i is the scale, and a_i is the appearance of the object. Other attributes can be added if it is suitable to the concrete tracking application.

Now we can define a track hypothesis.

Definition 1 (Single Track Hypothesis)**.** *A single track hypothesis is defined as an ordered list of object observations, i.e.,*

$$T_k = \left(\mathbf{x}_{k_1}, \mathbf{x}_{k_2}, \ldots, \mathbf{x}_{k_{l_k}}\right)$$

where $\mathbf{x}_{k_i} \in \mathcal{X}$ and l_k is the length of T_k.

Definition 2 (Association Hypothesis)**.** *An association hypothesis $\mathcal{T}$ is defined as a set of single track hypotheses, i.e.,*

$$\mathcal{T} = \{T_k\}.$$

The task of finding the best global detection association hypothesis can be formulated via the concept of maximum likelihood. That is:

$$\mathcal{T}_{ML} = \underset{\mathcal{T}}{\operatorname{argmax}}\, P(\mathcal{T}|\mathcal{X})$$

where $\mathcal{T}_{ML}$ represents the most likely association hypothesis given the set of observations $\mathcal{X}$. The function $P(\mathcal{T}|\mathcal{X})$ denotes the probability of the association hypothesis $\mathcal{T}$ being the correct association of observations given the observed data $\mathcal{X}$. This probabilistic approach accounts for various uncertainties and ambiguities in tracking multiple objects.

In this formulation, the goal is to maximize the likelihood of the entire set of trajectory hypotheses, considering all the available observations. This requires an efficient algorithm that can navigate through a potentially vast space of possible trajectories, each with its likelihood of accurately representing the true paths of the objects being tracked.

Using the Bayesian approach, we can express the likelihood of an association hypothesis in terms of conditional probabilities and prior knowledge. The Bayesian formulation of our maximum likelihood problem becomes:

$$\mathcal{T}_{ML} = \underset{\mathcal{T}}{\operatorname{argmax}}\, \frac{P(\mathcal{X}|\mathcal{T})P(\mathcal{T})}{P(\mathcal{X})}$$

In this reformulation, $P(\mathcal{X}|\mathcal{T})$ is the likelihood of observing the data $\mathcal{X}$ given the trajectory hypothesis $\mathcal{T}$, $P(\mathcal{T})$ is the prior probability of the trajectory hypothesis, and $P(\mathcal{X})$ is the evidence or the probability of observing the data.

The Bayesian approach offers several advantages. Firstly, it allows the incorporation of prior knowledge about object movements and interactions through $P(\mathcal{T})$. This can be particularly useful in scenarios where certain trajectories are more likely than others based on historical data or physical constraints.

Secondly, the evidence term $P(\mathcal{X})$, while often difficult to compute directly, does not depend on the hypothesis $\mathcal{T}$ and therefore does not affect the argmax operation. Therefore this means we can ignore this term when computing $\mathcal{T}_{ML}$, so the calculation simplifies to:

$$\mathcal{T}_{ML} = \underset{\mathcal{T}}{\operatorname{argmax}}\, P(\mathcal{X}|\mathcal{T})P(\mathcal{T})$$

This simplification still retains the essence of the Bayesian approach, combining the likelihood of the data given the hypothesis with the prior probability of the hypothesis.

Implementing the Bayesian approach in MOT algorithms involves two key components: estimating the likelihood function $P(\mathcal{X}|\mathcal{T})$ and defining suitable priors $P(\mathcal{T})$. The likelihood function can often be modeled based on the detection accuracy of the object tracking system and the dynamics of object movements. The prior probability $P(\mathcal{T})$ typically incorporates domain-specific knowledge, such as typical object trajectories and speeds.

The space of $\mathcal{T}$ can be very large. To make the search more tractable, a common simplification is to specify that trajectories cannot collide, meaning they cannot share an observation. Formally, this constraint can be stated as:

Definition 3 (Collision-free association hypothesis)**.** *Association hypothesis $\mathcal{T}$ is collision-free if for every $T_k, T_j \in \mathcal{T}$ where $k \neq j$, the tracks do not share observation, that is $T_k \cap T_j = \emptyset$.*

This non-collision constraint significantly reduces the complexity of the search space. Further on, we will, by default, consider that all association hypotheses must satisfy this non-collision constraint.

The complete global maximum search is still computationally unfeasible in practice. Therefore, different approaches to MOT differ in the following:

- how they model $P(\mathcal{X}|\mathcal{T})$
- how they model $P(\mathcal{T})$
- how they search for approximation of globally optimal collision-free association hypothesis $\mathcal{T}_{ML}$

2.3 Network Flow-based Algorithms

2.3.1 Description of the Algorithm

This description of the algorithm is based on [21].

First, to make a significant computational simplification to the model, we assume that the trajectories of the objects are independent from each other. We get:

$$P(\mathcal{X}|\mathcal{T}) = \prod_{\mathbf{x}_i \in \mathcal{X}} P(\mathbf{x}_i|\mathcal{T})$$

$$P(\mathcal{T}) = \prod_{T_k \in \mathcal{T}} P(T_k)$$

Now we can model $P(\mathbf{x}_i|\mathcal{T})$ as follows:

$$P(\mathbf{x}_i|\mathcal{T}) = \begin{cases} 1 - \beta_i & \text{if } \exists T_k \in \mathcal{T}, \mathbf{x}_i \in T_k \\ \beta_i & \text{otherwise} \end{cases}$$

The likelihood function for an observation $\mathbf{x}_i$, given the trajectory hypothesis $\mathcal{T}$, is denoted by $P(\mathbf{x}_i|\mathcal{T})$. This probability is modeled using a Bernoulli distribution, which accounts for the dual possibilities of an observation representing either a correct detection or a false positive. In this context, β_i represents the probability that the observation $\mathbf{x}_i$ is a false alarm.

Next we can model $P(T_k)$ as a Markov chain:

$$P(T_k) = P_{\text{entr}}(\mathbf{x}_{k_1}) \prod_{i=2}^{l_k} P_{\text{link}}(\mathbf{x}_{k_i}|\mathbf{x}_{k_{i-1}}) P_{\text{exit}}(\mathbf{x}_{k_{l_k}})$$

where P_{entr} is the initialization probability, P_{exit} is the termination probability, and $P_{\text{link}}(\mathbf{x}_{k_i}|\mathbf{x}_{k_{i-1}})$ are the transition probabilities for the track T_k.

Next, define following binary variables:

$$f_{\text{en},i} = \begin{cases} 1 & \text{if } \exists T_k \in \mathcal{T}, \text{ and } T_k \text{ starts from } \mathbf{x}_i, \\ 0 & \text{otherwise.} \end{cases}$$

$$f_{\text{ex},i} = \begin{cases} 1 & \text{if } \exists T_k \in \mathcal{T}, \text{ and } T_k \text{ ends at } \mathbf{x}_i, \\ 0 & \text{otherwise.} \end{cases}$$

$$f_{i,j} = \begin{cases} 1 & \text{if } \exists T_k \in \mathcal{T}, \text{ and } \mathbf{x}_j \text{ is right after } \mathbf{x}_i \text{ in } T_k, \\ 0 & \text{otherwise.} \end{cases}$$

$$f_i = \begin{cases} 1 & \text{if } \exists T_k \in \mathcal{T}, \text{ and } \mathbf{x}_i \text{ is in } T_k, \\ 0 & \text{otherwise.} \end{cases}$$

These variables apparently satisfy the following:

$$f_{\text{en},i} + \sum_j f_{j,i} = f_i = f_{\text{ex},i} + \sum_j f_{i,j} \quad \forall i \tag{2.1}$$

Now we can write:

$$\mathcal{T}_{ML} = \underset{\mathcal{T}}{\text{argmax}}\, P(\mathcal{T})P(\mathcal{X}|\mathcal{T})$$

$$\mathcal{T}_{ML} = \underset{\mathcal{T}}{\text{argmax}}\, \log\left(P(\mathcal{T})P(\mathcal{X}|\mathcal{T})\right)$$

$$\mathcal{T}_{ML} = \underset{\mathcal{T}}{\text{argmax}}\, \left(\log P(\mathcal{T}) + \log P(\mathcal{X}|\mathcal{T})\right)$$

$$\mathcal{T}_{ML} = \underset{\mathcal{T}}{\text{argmax}} \left(\log \prod_{T_k \in \mathcal{T}} P(T_k) + \log \prod_i P(\mathbf{x}_i|\mathcal{T})\right)$$

$$\mathcal{T}_{ML} = \underset{\mathcal{T}}{\text{argmax}} \left(\sum_{T_k \in \mathcal{T}} \log P(T_k) + \sum_i \log P(\mathbf{x}_i|\mathcal{T})\right)$$

$$\mathcal{T}_{ML} = \underset{\mathcal{T}}{\text{argmin}} \left(\sum_{T_k \in \mathcal{T}} -\log P(T_k) + \sum_i -\log P(\mathbf{x}_i|\mathcal{T})\right)$$

$$\mathcal{T}_{ML} = \underset{\mathcal{T}}{\text{argmin}} \left(\sum_{T_k \in \mathcal{T}} \left(C_{\text{en},k_0} f_{\text{en},k_0} + \sum_j C_{k_j,k_{j+1}} f_{k_j,k_{j+1}} + C_{\text{ex},k_{l_k}} f_{\text{ex},k_{l_k}}\right) + \sum_i \left(-\log(1-\beta_i) f_i - \log \beta_i (1-f_i)\right)\right)$$

$$\mathcal{T}_{ML} = \underset{\mathcal{T}}{\text{argmin}} \left(\sum_i C_{\text{en},i} f_{\text{en},i} + \sum_{i,j} C_{i,j} f_{i,j} + \sum_i C_{\text{ex},i} f_{\text{ex},i} + \sum_i C_i f_i\right)$$

where:

$$
\begin{aligned}
C_{\text{en},i} &= -\log P_{\text{entr}}(\mathbf{x}_i) \\
C_{\text{ex},i} &= -\log P_{\text{exit}}(\mathbf{x}_i) \\
C_{i,j} &= -\log P_{\text{link}}(\mathbf{x}_j|\mathbf{x}_i) \\
C_i &= \log\left(\frac{\beta_i}{1-\beta_i}\right)
\end{aligned}
$$

And where the binary variables $f_{\text{en},i}$, $f_{\text{ex},i}$, $f_{i,j}$, f_i in the argmin and argmax functions are subjected to equation 2.1.

Now we will construct graph $G(\mathcal{X})$ with a source s and sink t where $c(u,v)$ will denote the given cost of edge from u to v and $f(u,v)$ will the value of flow from u to v to be found. $G(\mathcal{X})$ will be constructed in the following way:

- For every observation $\mathbf{x}_i$ add vertices u_i and v_i
- For every observation $\mathbf{x}_i$ add an edge (u_i, v_i) with cost $c(u_i, v_i) = C_i$ and flow $f(u_i, v_i) = f_i$
- For every observation $\mathbf{x}_i$ add an edge (s, u_i) with cost $c(s, u_i) = C_{\text{en},i}$ and flow $f(s, u_i) = f_{\text{en},i}$
- For every observation $\mathbf{x}_i$ add an edge (v_i, t) with cost $c(v_i, t) = C_{\text{ex},i}$ and flow $f(v_i, t) = f_{\text{ex},i}$
- For every pair of observations $\mathbf{x}_i, \mathbf{x}_j$ for which $P_{\text{link}}(\mathbf{x}_j|\mathbf{x}_i) \neq 0$ add an edge (v_i, u_j) with cost $c(v_i, u_j) = C_{i,j}$ and flow $f(v_i, u_j) = f_{i,j}$

An example of such a graph can be seen in Figure 2.1.

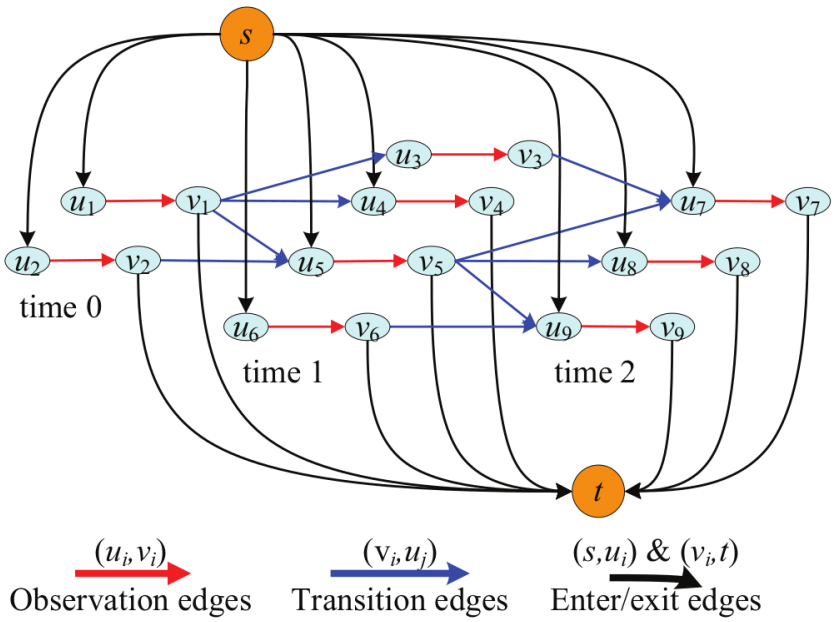


Figure 2.1: An example of the cost-flow network $G(\mathcal{X})$ with 3 timesteps and 9 observations.[21]

The equation 2.1 is equivalent to the flow continuity constraint (conservation of flow constraint) in flow network graphs. Solving for $\mathcal{T}_{ML}$ is then equivalent to solving for minimal cost flow in $G(\mathcal{X})$. In this interpretation, each flow path is equivalent to an object trajectory, and the flow amount sent from s to t is equal to the number of object trajectories. The total cost of the solution then corresponds to the log-likelihood of the association hypothesis.

The minimal cost flow problem is known to be solvable in polynomial time. Several different algorithms achieving this are known. Some of them are described in detail in [25]. In the 2010s improvements were made to the min cost flow based tracking algorithm [22, 23, 24].

2.3.2 Discussion of The Network Flow-based Algorithm

This approach of network flow-based methods to the MOT problem has become popular in recent years in the MOT community because of its computational efficiency and scalability.

The biggest disadvantage is that the method is only able to consider probabilities of linkages from a certain time frame to the very next time frame. In other words, the cost function can only contain unary and pairwise terms. The inability to consider the global attributes of track hypotheses is a costly restriction for some MOT applications. The restrictions include:

- The inability to include even relatively simple particle motion models. The pairwise terms are only able to represent the probability of the particle moving a certain distance from its immediately previous position without the consideration of further history of motion.

- The inability to incorporate the length of the track hypothesis. In certain applications, there are known restrictions on the lifetimes of the objects, which cannot be considered in this method.

- The inability to consider the long-term appearance of the object. Although the pairwise terms can incorporate the restriction of temporal appearance continuity (the object's appearance should not change significantly from frame to frame), the need to address the global considerations of appearance in some applications is not met.

Mathematically the restriction corresponds to considering $P_{\text{link}}(\mathbf{x}_{k_{i+1}}|\mathbf{x}_{k_i})$ instead of $P_{\text{link}}(\mathbf{x}_{k_{i+1}}|\mathbf{x}_{k_1} \dots \mathbf{x}_{k_i})$.

2.4 Multiple Hypothesis Tracking

2.4.1 Introduction to Multiple Hypothesis tracking

Introduced in 1979 by Reid [31], Multiple Hypothesis Tracking (MHT) has established itself as a successful technique across various tracking applications. This method addresses the data association challenge by generating a tree of potential track hypotheses for each target, making it particularly effective for incorporating higher-order information like long-term motion and appearance models. This is because MHT considers the entirety of a track hypothesis when calculating likelihoods.

Despite its advantages, MHT is often viewed as a slow and memory-intensive approach. It requires specific pruning techniques to manage its complexity. While MHT gained significant attention in the vision field during the 1990s, it has not been the primary method for most MOT applications in the recent two decades. Nonetheless, MHT remains a preferred solution in specific scenarios.

The use of MHT is notable in radar target tracking [28, 29, 30] and extends to contemporary applications in areas such as traffic surveillance [33], multi-sensor fusion [32], and the tracking of clusters of people in densely populated settings where tracking of individuals is not feasible [34, 35].

The core idea of the MHT algorithm is to represent track hypotheses using *hypothesis trees*. A hypothesis tree is a rooted tree, in which every node represents the hypothesis of a track starting at the tree's root and ending in said node.

The effectiveness of MHT heavily depends on its ability to prune branches in the hypothesis trees quickly and efficiently. Early attempts to apply MHT in visual tracking were impeded by inaccurate target detectors and less effective motion models, leading to an overwhelming increase in the search space and the need for robust pruning strategies. However, recent advancements in machine learning and deep learning have sparked a renewed interest in MHT, especially its potential to utilize long-term temporal data for tracking purposes. Innovations in tracking-by-detection and developing sophisticated feature representations for object appearance have opened new avenues for MHT applications.

The general illustration of the MHT algorithm can be seen in figure 2.2.

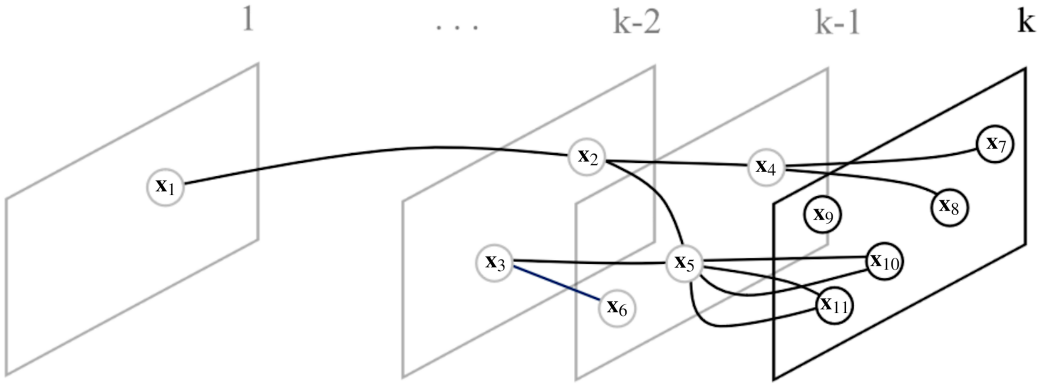


Figure 2.2: General illustration of MHT with four frames and eleven detections shown. [26], modified.

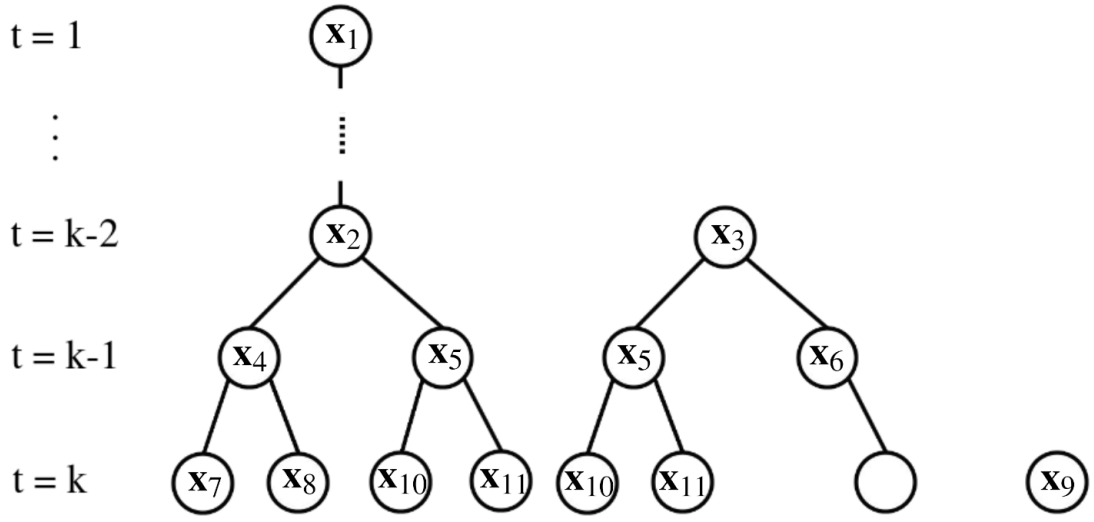


Figure 2.3: Ilustration of a collection of hypothesis trees in MHT corresponding to the situation in Figure 2.2. Note that not all trees are shown for simplicity. [26], modified.

In Figure 2.3, we can see trees corresponding to the situation in Figure 2.2. Not all trees are shown. For simplicity, only trees starting in $\mathbf{x}_1$, $\mathbf{x}_3$ and $\mathbf{x}_9$ are shown.

2.4.2 Description of the MHT algorithm

The MHT algorithm proceeds frame by frame. At k-th frame these steps are taken:

- **New trees foundation:** For each detection at frame k, start a new tree with root in that detection.
- **Existing trees extension:** For every tree leaf L at frame $k-1$ find all detections at frame k, which represent suitable continuations of the track hypothesis corresponding to the node L and add connections from L to all those detection at frame k.
- **Track scoring:** For every new node in existing trees, compute a score based on the modeled tracking problem. In most applications, this is the log-likelihood of the corresponding track hypothesis being represented or log-likelihood ratio between the positive hypothesis (representing a valid track) and the null hypothesis (representing false positive detections).
- **Collision graph construction:** A graph is constructed where the graph vertices correspond to the existing nodes of the trees and any two vertices are connected by an edge if and only if the corresponding track hypotheses are in conflict (they share a detection). Each vertex is given a weight equal to the score of the corresponding track hypothesis computed in the previous step.
- **Global hypothesis formation:** In the mentioned collision graph, a maximum weighted independent set is found. That is, a set of vertices in which none two of them are adjacent and which maximize the sum of their scores.
- **Tree pruning:** In order to keep the number of hypotheses manageable, each tree is pruned. In each tree the track hypothesis selected to the global hypothesis is kept intact, but some other parts of the tree are deleted according to some pruning heuristic.
- **Deletion of not selected trees:** All trees with no nodes selected to the global hypothesis are deleted.

In the last frame, the final global hypothesis is found according to the same rules and this is considered the final output of the MHT algorithm.

We can see that the MHT algorithm is essentially a breath-first search algorithm with added pruning, which keeps the number of hypotheses manageable in the run of the algorithm. Delaying data association decisions by keeping multiple hypotheses active until ambiguities are resolved is a critical tactic in MHT.

Let us discuss some parts of the MHT algorithm in more detail.

New trees foundatios

A track tree represents a collection of hypotheses starting from a single detection. Each frame, a new track tree is created for each detection, indicating the possibility that the observation corresponds to a new object entering the scene. Even if the detection is spatially close to a detection in the previous frame, new tree is still created, because of the possibility that the global considerations will in time show, that it was indeed likely a new object.

Existing trees extension

Existing track trees are also updated to include observations from the current frame. Each track hypothesis is expanded by adding new observations as child nodes within its gating area, with each new observation generating a new branch.

To account for potential missing detections (False negatives in the detection algorithm), a separate branch with a dummy observation is always created as well. Note that this is not shown in Figure 2.3 for simplicity. When adding branches to existing dummy observation nodes, the history of the track hypothesis must be considered. For example, one can limit the number of dummy observations, which can end up in a row. If this limit is reached, the dummy observation node cannot have another dummy observation child added, only real detection children, if there are any. Usually only one missing detection is allowed in a row, so a node corresponding to a missing observation can only take real detection as children.

Gating

Based on motion estimations, a gating region can be predicted for each track hypothesis, indicating where the next track observation is expected to appear.

The simplest motion model utilizes only the previous frame and takes a circular region of a given radius around the coordinates of the prior observation. In this model, no information about the motion from earlier frames is used; the model merely tests if the object has not moved more than a specified distance between two frames. This is suitable for particles that exhibit Brownian motion.

More sophisticated approaches to gating can employ a linear model of motion or even more complex motion models to predict object positions in the subsequent frame and then construct the gating area around that prediction. This is more suitable if the particles are thought to retain inertia between frames for example.

Track scoring

In this step, each track hypothesis is associated with a track score. The most general approach we can consider again uses the concept of log-likelihood. For the m-th track hypothesis starting at frame k_{start} constructed to the k-th frame

$$T_m = \{\mathbf{x}_{m_{k_{\text{start}}}}, \dots, \mathbf{x}_{m_k}\}$$

we can, in general, compute the log-likelihood score as follows:

$$S^m(k) = \log\left(\prod_{j=k_{\text{start}}}^{k} P(\mathbf{x}_{m_j}|\mathbf{x}_{m_{k_{\text{start}}}} \dots \mathbf{x}_{m_{j-1}})\right) = \sum_{j=k_{\text{start}}}^{k} \log P(\mathbf{x}_{m_j}|\mathbf{x}_{m_{k_{\text{start}}}} \dots \mathbf{x}_{m_{j-1}})$$

Another approach for track scoring was introduced in [28]. In this approach, the log likelihood ratio (LLR) is used. For the m-th track hypothesis, the LLR-based score at frame k can be defined as:

$$S^m(k) = \log\left(\frac{\prod_{j=k_{\text{start}}}^{k} P(\mathbf{x}_{m_j}|\mathbf{x}_{m_{k_{\text{start}}}} \dots \mathbf{x}_{m_{j-1}}, H_1)}{\prod_{j=k_{\text{start}}}^{k} P(\mathbf{x}_{m_j}|\mathbf{x}_{m_{k_{\text{start}}}} \dots \mathbf{x}_{m_{j-1}}, H_0)}\right)$$

This can be further simplified to:

$$S^m(k) = \sum_{j=k_{\text{start}}}^{k} \log\left(\frac{P(\mathbf{x}_{m_j}|\mathbf{x}_{m_{k_{\text{start}}}} \dots \mathbf{x}_{m_{j-1}}, H_1)}{P(\mathbf{x}_{m_j}|\mathbf{x}_{m_{k_{\text{start}}}} \dots \mathbf{x}_{m_{j-1}}, H_0)}\right)$$

In this formulation, the numerator represents the likelihood of the track data under the positive hypothesis, and the denominator represents the likelihood under the null hypothesis.

However in this formulation is often to explicitly combine the information about appearance and motion of the objects being tracked. To mitigate this problem, the authors of [26] suggest following formulation:

$$S^m(k) = w_{\text{mot}} S^m_{\text{mot}}(k) + w_{\text{app}} S^m_{\text{app}}(k)$$

where

$$S^m_{\text{mot}}(k) = \sum_{j=k_{\text{start}}}^{k} \log\left(\frac{P(x_{m_j}|x_{m_{k_{\text{start}}}} \dots x_{m_{j-1}}, H_1)}{P(x_{m_j}|x_{m_{k_{\text{start}}}} \dots x_{m_{j-1}}, H_0)}\right)$$

and

$$S^m_{\text{app}}(k) = \sum_{j=k_{\text{start}}}^{k} \log\left(\frac{P(a_{m_j}|a_{m_{k_{\text{start}}}} \dots a_{m_{j-1}}, H_1)}{P(a_{m_j}|a_{m_{k_{\text{start}}}} \dots a_{m_{j-1}}, H_0)}\right)$$

are the motion and appearance scores, respectively. Here, w_{mot} and w_{app} are weights that control the contribution of the motion (position) and appearance information to the overall track score.

The motion score $S^m_{\text{mot}}(k)$ quantifies the likelihood of the sequence of position observations (x^{mot}_m) fitting the model for the target object under H_1 versus the background under H_0. Similarly, the appearance score $S^m_{\text{app}}(k)$ assesses the likelihood of the sequence of appearance observations (a^{app}_m) fitting the target object model versus the background model.

This combined score approach leverages the strengths of both position and appearance information, providing a more robust and accurate assessment of track hypotheses. It is particularly useful in complex environments where relying on a single type of data might lead to increased false positives or missed detections.

Collision graph and global hypothesis formation

In the context of trajectory analysis, we consider a set of trees that represent all potential trajectory hypotheses for various targets. The aim is to ascertain the most likely combination of object tracks at a specific frame, denoted as frame k. To facilitate this, we construct a collision graph wherein the vertices correspond to the nodes of the aforementioned trees. An edge is formed between any two vertices if, and only if, the associated track hypotheses are in conflict, meaning they share a detection. Each vertex is assigned a weight equal to the score of its corresponding track hypothesis.

The task then becomes to identify a subset of vertices within the collision graph that are not pairwise adjacent and whose cumulative scores are maximized. This subset represents the set of non-conflicting hypotheses. This problem aligns with a well-known concept in graph theory and combinatorial optimization, termed the Maximum Weighted Independent Set (MWIS) problem.

The MWIS problem is a fundamental problem in graph theory and has significant applications across various fields, including bioinformatics, networking, and scheduling.

To provide a more formal description:

A graph is denoted as $G = (V, E)$, where V is the set of vertices and E is the set of edges.

Definition 4. *The **neighborhood** of a vertex v, denoted $N(v)$, is the set of all vertices in V that are adjacent to v, i.e., $N(v) = \{u \in V | (u, v) \in E\}$.*

Definition 5. *An **independent set** in a graph is a set of vertices $I \subseteq V$ such that no two vertices in I are adjacent, i.e., for every pair of vertices $u, v \in I$, $(u, v) \notin E$. The **weight** of an independent set I is defined as $W(I) = \sum_{v \in I} w(v)$, where $w(v)$ is the weight of vertex v. A **maximum weighted independent set (MWIS)** is an independent set I for which there is no other independent set J with $W(J) > W(I)$.*

The Maximum Weighted Independent Set problem can be formally defined as follows: Given a weighted graph $G = (V, E)$, with a weight function $w : V \to \mathbb{R}$, the goal is to find a MWIS in G, as formulated in Definition 5.

Note that in the literature, the Maximum Weighted Independent Set (MWIS) problem is sometimes referred to as the Maximum Weighted Clique (MWC) Problem. A clique is defined as a set of vertices in which every pair of vertices is adjacent. Consequently, solving the Maximum Weighted Independent Set problem in a graph is precisely equivalent to solving the Maximum Weighted Clique Problem in the complement of that graph, with identical vertex weights, and vice versa. Thus, any algorithm used to solve the Maximum Weighted Clique Problem can also be applied to the MWIS, making it applicable for the purpose of the MHT algorithm.

It is apparent that the MWIS problem can be solved in each component of the graph independently.

Definition 6. *A **component** of a graph $G = (V, E)$ is a subgraph $C = (V_C, E_C)$ such that:*

- *For every pair of vertices $u, v \in V_C$, there exists a path in C connecting u and v.*
- *C is maximal with respect to this property, meaning there is no vertex $w \in V \setminus V_C$ such that adding w to C along with any edges in E incident to w would still satisfy the first condition.*

Let $\mathcal{C}(G)$ denote the set of all components of the graph G, such that $\mathcal{C}(G) = \{C_1, C_2, \ldots, C_k\}$ where each C_i is a component of G and k is the total number of components in G.

No two components of the graph contain respective vertices with an edge connecting them; this is inherent in the definition of graph components. Therefore, the independent set chosen in each component does not influence the selection of independent sets in other components. This characteristic allows us to approach the MWIS problem on a per-component basis and then aggregate the solutions to form a global solution for the entire graph.

The rationale for solving the MWIS problem independently within each component of $\mathcal{C}(G)$ and then aggregating the solutions is as follows:

- **Independence of Components**: The components in $\mathcal{C}(G)$ are disjoint with respect to their edges, meaning that any independent set within a component C_i does not share vertices with any other component in $\mathcal{C}(G)$. Thus, solving the MWIS problem in one component C_i does not affect the solutions in the other components.
- **Maximization within Components**: In the MWIS problem, the goal is to maximize the total weight of the independent set. By addressing the MWIS problem within each component $C_i \in \mathcal{C}(G)$, we ensure the identification of the maximum weight independent set for that specific component.
- **Aggregation of Solutions**: Upon solving the MWIS problem in each component C_i of $\mathcal{C}(G)$, the independent sets obtained from each component can be combined to construct a global solution for the entire graph G. This combined solution remains valid and independent, as there are no edges between the distinct components in $\mathcal{C}(G)$.

In further paragraphs let us suppose we are only solving the MWIS problem unicomponent graph.

The MWIS problem is known to be NP-hard [40], which suggests that it is highly unlikely for an algorithm to exist that can solve it exactly in polynomial time. The implication of finding such an algorithm would be profound, as it would mean $P = NP$.

Though a brute-force search could theoretically find an exact solution, this approach becomes infeasible even for relatively small graphs due to its exponential time complexity. A more practical approach

is to employ backtracking. The central idea in backtracking is recognizing that any potential solution containing two vertices connected by an edge is invalid, as are all its extensions. Therefore, these solutions and their extensions can be efficiently skipped. An example of an algorithm that utilizes this methodology is Algorithm 1.

Algorithm 1 Backtracking Algorithm for MWIS

```
1:  procedure BacktrackingMWIS(G, v, I, maxI, maxW)
2:      if v is beyond the last vertex of G then
3:          if weight of I > maxW then
4:              maxW ← weight of I
5:              maxI ← I
6:          end if
7:          return
8:      end if
9:      V ← set of vertices in G
10:     w(v) ← weight of vertex v
11:     if v and I are independent then
12:         Call BacktrackingMWIS(G, v + 1, I ∪ {v}, maxI, maxW)
13:     end if
14:     Call BacktrackingMWIS(G, v + 1, I, maxI, maxW)
15: end procedure
16: procedure MWIS(G)
17:     I ← ∅                          ▷ Current independent set
18:     maxI ← ∅                       ▷ Maximum weight independent set
19:     maxW ← 0                       ▷ Maximum weight
20:     Call BacktrackingMWIS(G, 1, I, maxI, maxW)
21:     return maxI
22: end procedure
```

While this backtracking approach is more efficient than a simple brute-force algorithm, the space of all potential independent sets remains prohibitively large, even for graphs of relatively modest size. More sophisticated exact solvers for the MWIS problem do exist. One such popular algorithm is detailed in [36]. A more modern, improved algorithm and experiments can be found in [37]. However, it's important to note that the asymptotic time complexity of these advanced solvers is still exponential, not polynomial, in relation to the number of vertices.

Because of the nature of the exact problem, there is a significant need in practically efficient heuristic algorithms. One simple possibility is an algorithm that is locally greedy in choosing the vertex with maximal weight, which is not connected by an edge to any other vertex already chosen. This algorithm is simply called *greedy MWIS algorithm* in the MWIS literature. This approach can be seen in Algorithm 2.

This simple greedy algorithm can be made highly time efficient. To improve the time efficiency, the vertices can be sorted by their weights in $O(n \log n)$ time, where n is the number of vertices. Subsequently, the algorithm iterates over the vertices in descending order, selecting each vertex that does not conflict with previously selected vertices.

Algorithm 2 Greedy Algorithm for MWIS

```
1:  procedure GreedyMWIS(G)
2:      I ← ∅
3:      V ← set of vertices in G
4:      while V ≠ ∅ do
5:          v ← vertex in V with maximum weight w(v)
6:          I ← I ∪ {v}
7:          V ← V \ {v}
8:          Remove all neighbors of v from V
9:      end while
10:     return I
11: end procedure
```

However, this simple greedy approach can be highly suboptimal, particularly when a vertex with a high weight is connected to multiple vertices with marginally lower weights, which are either unconnected or sparsely connected among themselves. In such cases, the algorithm tends to select the vertex with the highest weight, overlooking the opportunity to select a large number or all of the other vertices. An illustration of this scenario is depicted in Figure 2.4, with the suboptimal solution on the left and the optimal solution on the right.

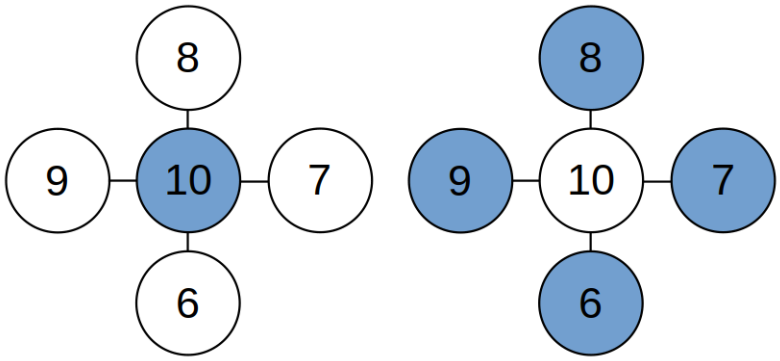


Figure 2.4: An example of a star graph with weights written in the vertices. Left: The simple greedy MWIS algorithm selects the center vertex with the weight value 10 and terminates. Right: the actual exact solution of the MWIS problem. The modified greedy penalty-based MWIS algorithm identifies the solution correctly.

We can modify the greedy algorithm to address the issue illustrated in Figure 2.4. For each vertex $v \in V$, we compute a penalty

$$p(v) = \sum_{u \in N(v)} w(u),$$

where $p(v)$ represents the total weight of neighboring vertices of v that are excluded from the solution when v is selected. The modified algorithm proceeds greedily, at each step choosing a vertex with the smallest penalty that is not adjacent to any vertex already included in the solution.

This heuristic can also be interpreted differently. At each step, the algorithm selects a vertex v to minimize

$$\sum_{u \in N(v)} w(u),$$

which is equivalent to maximizing

$$\sum_{u \in V} w(u) - \sum_{u \in N(v)} w(u) = w(v) + \sum_{u \in V \setminus (\{v\} \cup N(v))} w(u).$$

This expression represents the total potential value from the weights of the remaining vertices. However, it should be noted that after each choosing of a vertex to the solution set, the penalties should be recomputed so that for each vertex, they represent the sum of weights of only those adjacent vertices that still have the potential to be selected into the solution set. This approach is detailed in Algorithm 3.

Algorithm 3 Modified Greedy Algorithm for MWIS with Penalties

procedure GREEDYMWISPENALTIES(G)
  $I \leftarrow \emptyset$ ▷ Chosen set for MWIS
  $V \leftarrow$ set of vertices in G
  For each $v \in V$, compute $p(v) = \sum_{u \in N(v)} w(u)$ ▷ Initial penalties
  while $V \neq \emptyset$ **do**
    $v \leftarrow$ vertex in V with the least penalty $p(v)$
    $I \leftarrow I \cup \{v\}$
    $V \leftarrow V \setminus (\{v\} \cup N(v))$ ▷ Remove v and its neighbors from V
    For each $u \in V$, recompute $p(u) = \sum_{n \in N(u) \cap V} w(n)$
  end while
  return I
end procedure

Algorithm 3 is presented in a different, yet equivalent, form in [38], where the authors frame the problem as a Maximum Weighted Clique Problem. In [38], three other algorithms are introduced as extensions of Algorithm 3. These algorithms tend to yield solutions closer to the optimal.

Other diverse approaches to solving the MWIS problem approximately have been proposed, including various sequential greedy heuristics, different local search heuristics, simulated annealing, genetic algorithms, and neural networks. An extensive survey of these methods is available in [39].

The output of the MWIS algorithm performed on the collision graph is said to constitute the global hypothesis for the given frame.

Hypothesis tree pruning

Pruning is a crucial step in the MHT algorithm. Without pruning, the number of track hypotheses would increase exponentially over time. A commonly used pruning technique, as described in [26], is N-scan pruning. In case of N-scan pruning, the pruning is performed at frame k if $k > N$. During N-scan pruning at frame k, the algorithm first identifies all branches that were selected to the global hypotheses. Subsequently, within each selected branch, all subtrees that diverge from the branch at the node at frame $k - N$ are pruned. Alternatively, this can be described as consolidating connections from nodes at frame $k - N$ to only one of their immediate children at frame $k - N + 1$, effectively implementing a lookahead of N frames into the future in the frame $k - N$. This process is illustrated in Figure 2.5.

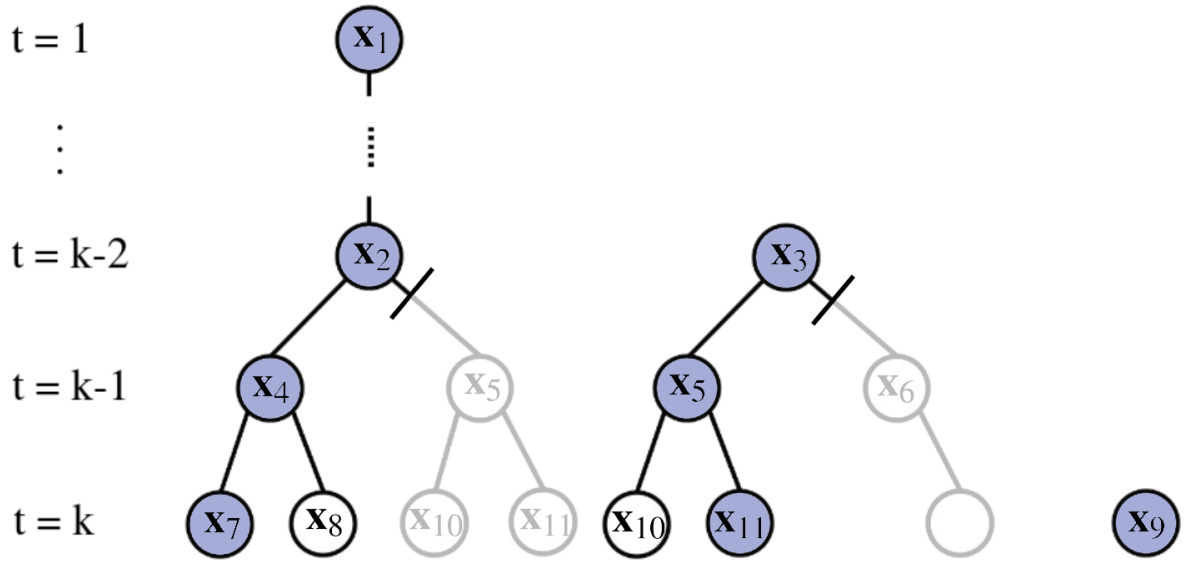


Figure 2.5: Ilustration of a collection of hypothesis trees in MHT corresponding to the situation after N-scan pruning performed at frame k. In this example, three hypotheses were selected to the global hypothesis, highlighted in blue, and $N = 2$. [26], modified.

There are other pruning techniques that can be used on their own or, more commonly, in combination with N-scan pruning. For example:

- Capping the number of leaves: If the number of leaves of a hypothesis tree exceeds threshold $L_{\max}$, Then $L_{\max}$ leaves with the highest scores are identified and all other leaves are deleted.
- Relative threshold pruning: In this approach, a relative threshold is set in comparison to the best hypothesis in the tree. If a hypothesis has a score or probability significantly lower than that of the best hypothesis, typically by a predefined factor or percentage, it is deleted.

At the end of the pruning step, all trees that do not have any nodes selected to the global hypothesis are deleted completely.

2.5 Tracking Performance Metrics

2.5.1 Introduction to Tracking Performance Metrics

In the field of object tracking, performance metrics play a crucial role in evaluating and comparing the effectiveness of various tracking algorithms. The exposition of these tracking performance metrics primarily draws upon [41, 42], which comprehensively describe the most prevalent metrics in the field. For additional types of metrics, refer to [43].

Tracking performance metrics can be broadly categorized into two groups: precision and accuracy.

- **Precision:** Precision in tracking metrics evaluates how well a tracker can estimate the position or trajectory of individual objects. It primarily focuses on the spatial accuracy of the tracked objects' locations. Metrics such as Multiple Object Tracking Precision (MOTP) and Root Mean Square Error (RMSE) are examples of precision measurements. MOTP assesses the overlap between the predicted and actual bounding boxes of objects, while RMSE measures the Euclidean distance between the predicted and true positions of objects. Precision metrics are crucial in scenarios where the exact location of an object is of paramount importance, such as in detailed motion analysis or precise object localization.

- **Accuracy:** On the other hand, accuracy in tracking metrics concerns itself with the correctness of data association in tracking. It evaluates how accurately the tracker maintains consistent identity labels of objects across frames. This aspect of tracking is vital in scenarios where the continuity of object identities is crucial, such as in surveillance or behavior analysis. Accuracy metrics address questions like whether the tracker correctly identifies new objects, avoids identity switches, and effectively handles occlusions and interactions among multiple objects. Metrics such as Multiple Object Tracking Accuracy (MOTA) are designed to capture these aspects.

In summary, while precision metrics like MOTP and RMSE gauge the spatial fidelity of object tracking, accuracy metrics assess the correctness and consistency of object identity tracking over time. Both sets of metrics are integral for a comprehensive evaluation of a tracking system's performance.

2.5.2 Multiple Object Tracking Precision

Multiple Object Tracking Precision (MOTP) measures the precision of object tracking in terms of position and size. It is defined as follows:

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t}$$

where:

- $d_{t,i}$ represents the bounding box overlap between the predicted and the ground truth bounding boxes for the i-th object at time t.

- c_t denotes the number of correctly matched object-hypothesis pairs at time t.

- The summation over t extends across all time frames, and the summation over i includes all object-hypothesis pairs.

The bounding box overlap $d_{t,i}$ is commonly calculated using the Intersection over Union (IoU) metric. IoU is defined as the area of overlap between the predicted bounding box and the ground truth bounding box, divided by the area of their union. This is illustrated in Figure 2.6.

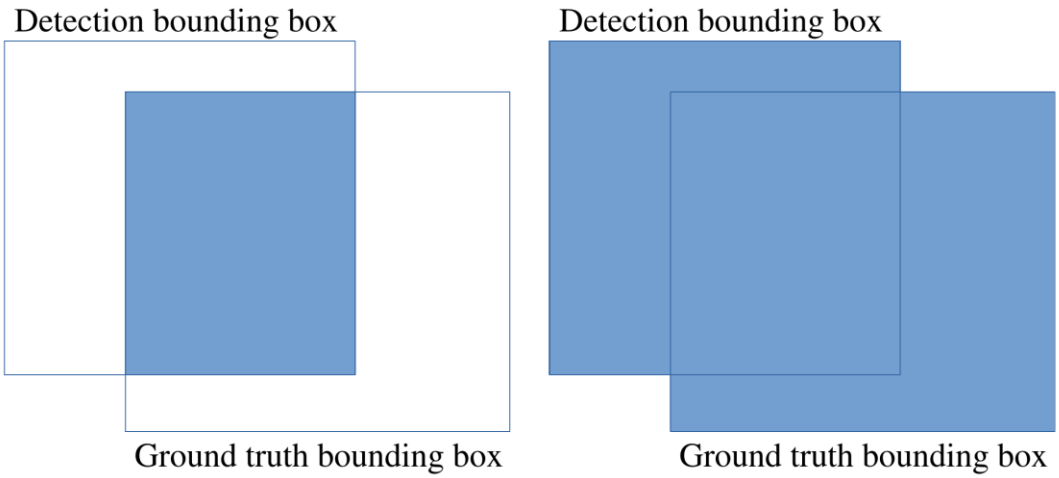


Figure 2.6: Illustration of intersection and union of the ground truth bounding box and the bounding box computed as the output of the detection and tracking algorithm. IoU is the ratio between the areas of intersection and union.

A higher MOTP value indicates greater precision in tracking, as it implies a closer match between the predicted and actual positions and sizes of the objects. Note, however, that while MOTP assesses localization accuracy, it does not evaluate other aspects of tracking performance, such as the consistency of object identities over time.

2.5.3 Root Mean Square Error

Root Mean Square Error (RMSE) is another metric used in object tracking, particularly suitable for tracking particles or objects without defined bounding boxes. RMSE quantifies the tracking accuracy by measuring the Euclidean distance between the predicted positions and the ground truth positions of the objects. It is defined for each object at each frame and then aggregated over all frames and objects. The formal definition is as follows:

$$RMSE = \sqrt{\frac{\sum_{t,i} \|x_{t,i} - \hat{x}_{t,i}\|^2}{\sum_t c_t}}$$

where:

- $x_{t,i}$ is the ground truth position of the i-th object at frame t.
- $\hat{x}_{t,i}$ is the predicted position of the i-th object at frame t.
- c_t denotes the number of correctly matched object-hypothesis pairs at time t.
- The summation over t extends across all time frames, and the summation over i includes all correctly matched object-hypothesis pairs.

The RMSE value provides a direct measure of the tracking accuracy in terms of spatial displacement. A lower RMSE value indicates higher precision in tracking, implying that the predicted positions are, on average, closer to the actual positions of the objects. This metric is especially useful in scenarios where tracking involves estimating precise locations, such as in particle tracking, rather than bounding box overlaps.

2.5.4 Multiple Object Tracking Accuracy

Multiple Object Tracking Accuracy (MOTA) is a pivotal metric in the realm of object tracking, designed to evaluate the overall accuracy of a tracking system. Unlike MOTP, which focuses on the spatial precision of tracking, MOTA provides a comprehensive measure of the tracking system's ability to maintain consistent object identities over time. It combines three critical aspects: missed detections, false positives, and identity switches.

MOTA is formally defined as:

$$MOTA = 1 - \frac{\sum_t (\mathrm{FN}_t + \mathrm{FP}_t + \mathrm{IDSW}_t)}{\sum_t \mathrm{GT}_t}$$

where at each time frame t:

- FN_t represents the number of false negatives or missed detections.
- FP_t denotes the number of false positives.
- IDSW_t is the number of identity switches.
- GT_t indicates the total number of ground truth objects.

The MOTA score can range from $-\infty$ to 1, where a higher score indicates better tracking performance. A score of 1 implies perfect tracking with no missed detections, no false positives, and no identity switches.

MOTA provides a holistic view of a tracker's performance, particularly in complex scenarios where objects interact closely or become occluded. It penalizes trackers for losing track of objects (missed detections), incorrectly adding new objects (false positives), and mixing up object identities (identity switches). Due to its comprehensive nature, MOTA is widely regarded as a crucial metric for assessing the overall effectiveness of object tracking algorithms in various applications, from surveillance to autonomous navigation.

However, it is important to note that while MOTA is a valuable metric, it should be considered alongside other metrics like MOTP for a more complete evaluation of a tracking system's capabilities. While MOTA excels in measuring tracking accuracy, MOTP complements it by providing insights into the spatial precision of the tracking.

Chapter 3

Data Acquisition and Preparation

3.1 Chapter Overview

This chapter delineates the comprehensive process of TIRF-SIM data acquisition and the steps undertaken to process the data for the purpose of developing a U-Net-based detection algorithm aimed at identifying positions of CCPs in TIRF-SIM images and a tracking algorithm based on MHT.

The initial sections introduce the cell type and lineage used in the thesis, as well as the specific cultivation techniques employed to ensure healthy cell growth and optimal imaging conditions. Subsequent sections describe the specific setup of the TIRF-SIM equipment and the data acquisition process itself, which is integral to capturing the high-resolution images necessary for accurate CCP detection.

The chapter then proceeds to expound on the data format, emphasizing the organization and the intrinsic attributes of the microscopy images. Following this, an explanation of the normalization process is provided, which standardizes the images to facilitate more effective algorithm training.

The final section of the chapter focuses on the annotation process, where human experts mark the CCPs within the images. These annotations form the ground truth against which the U-Net algorithm's and the tracker's performance are gauged. The chapter sets the stage for discussions on the development and validation of a tracking algorithm, which builds upon the detection algorithm's ability to recognize CCPs in temporal image sequences.

3.2 Cell Type, Lineage and Cultivation

Human retinal pigment epithelial cells (Figure 3.1) were employed as the primary cell line to obtain the microscopic images used in this thesis.

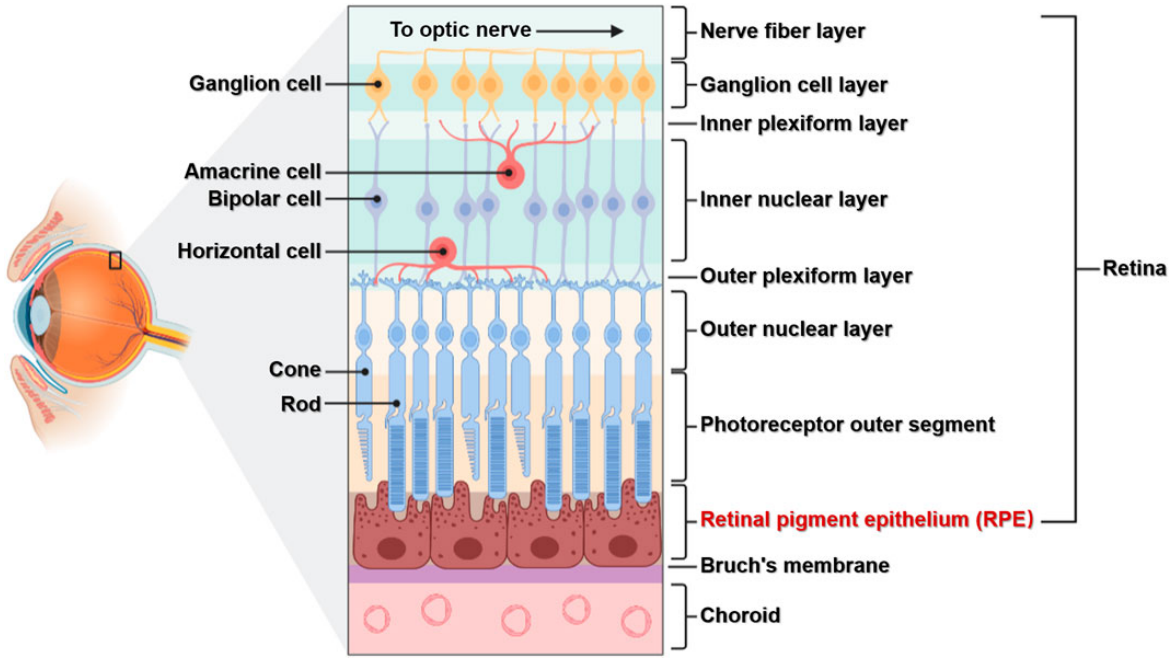


Figure 3.1: Position of human retinal pigment epithelial cells in the human eye. [48]

Retinal pigment epithelial (RPE) cells offer several advantages in studying the process of endocytosis using TIRF technology:

- **Low Phototoxicity**: RPEs demonstrate relatively lower phototoxicity under intense light [45, 46], minimizing cellular damage during imaging.
- **Flat and Regular Morphology**: The flat and regular shape of epithelial cells in general is ideal for the uniform focal plane required in TIRF microscopy.
- **Inherent Pigmentation**: The pigmentation is helpful since it reduces background fluorescence and light scattering, enhancing image contrast and clarity.
- **Polarized Structure**: Highly beneficial for studying cellular processes like protein trafficking and membrane dynamics.
- **Stable Attachment to Substrates**: Exhibits strong adherence to substrates, essential for close contact with the cover glass in TIRF microscopy.
- **Relevance to Ocular Diseases**: Direct relevance to ocular diseases makes them ideal for studying pathological processes at the cellular level.

Specifically, the hTERT-RPE-1 lineage [44] was utilized. This lineage consists of human retinal pigment epithelial cells provided by the American Type Culture Collection (ATCC).

The cells are immortalized using human Telomerase Reverse Transcriptase (hTERT) [47]. This method involves the transfer of exogenous hTERT cDNA into the cells, encoding the catalytic subunit of human telomerase. This process effectively prevents telomere shortening, a key factor in cellular aging, and overcomes telomere-controlled senescence. Consequently, hTERT-mediated immortalization allows cells to maintain prolonged division and growth capabilities without undergoing senescence, and importantly, without inducing cancer-associated changes or significantly altering their phenotypic properties [47].

Comprehensive details, including the product specification, certificate of analysis, and safety data sheet, are accessible in [44]. The hTERT-RPE-1 cell line, in particular, offers several advantages for TIRF imaging:

- **Human Origin**: Offers a more relevant model for human diseases and physiological processes.
- **Stable Genetic and Phenotypic Characteristics**: Retains many characteristics of primary cells, enhancing reproducibility.
- **Barrier Function and Polarization**: Maintains key features for studying drug transport and retinal physiology.
- **Ease of Genetic Manipulation**: Amenable to techniques like CRISPR/Cas9, enabling studies on gene function.
- **Versatility in Genetic Manipulation**: Can be genetically manipulated to express fluorescent proteins, enhancing utility in visualizing specific cellular components and processes.
- **High Reproducibility and Availability**: Can be obtained from multiple repositories with consistent quality.
- **Immortalization**: The immortalization of hTERT-RPE-1 cells facilitates extended culture periods without the typical senescence associated with primary cell lines, making them ideal for long-term studies.

As mentioned, the cell lineage was procured from ATCC and maintained in Dulbecco's Modified Eagle Medium (DMEM, catalog number 31966-021 from Thermo Fisher Scientific-Gibco), supplemented with 10% Fetal Bovine Serum (FBS, obtained from Thermo Fisher Scientific). All cell cultures were incubated at a constant temperature of 37°C in an atmosphere containing 5% CO_2.

Routine testing for mycoplasma contamination was conducted using the MycoAlert Mycoplasma Detection Kit (Lonza). Additionally, HepG2 and HeLa S3 cell lines, also sourced from the ATCC, were utilized for the preparation of cytosolic extracts. These lines were cultured under identical conditions.

3.3 Cell Imaging Techniques and TIRF-SIM Microscopy Setup

The imaging process was designed to ensure high precision and clarity, utilizing a custom-built TIRF-SIM system. The methodology is outlined briefly as follows:

- **Coverslip Preparation**: High precision, round #1.5 coverslips (Product #2850-22, Corning) underwent acid washing and were subsequently coated with a 0.2 mg/ml concentration of gelatin to facilitate cell adherence.
- **Cell Culture Conditions During Imaging**:
  - Throughout the imaging process, cells were maintained at an optimal temperature of 37°C.
  - The culture medium used was DMEM (catalog number 31966-021, Thermo Fisher Scientific-Gibco) supplemented with 10% fetal bovine serum.
- **TIRF-SIM Imaging Equipment**:
  - The TIRF-SIM was performed using a custom-built system at the Kennedy Institute, University of Oxford, described in [49, 50].
  - The system comprises a 1.49-NA 100× objective lens (Olympus) on an inverted microscope body (IX-83, Olympus), coupled with two high-speed sCMOS cameras (ORCA-Flash4.0, Hamamatsu).
  - A key feature is the ferroelectric spatial light modulator (SLM) for generating structured illumination patterns and adjusting TIRF incidence angles.
  - The SLM can switch modes at high repetition rates, enabling fast imaging up to a 200-ms frame rate.
- **Imaging Details**:
  - Excitation wavelength of 488 nm was used to capture the clathrin tagged with corresponding fluorophore.
  - The incidence angle was optimized for the wavelength of 488 nm to ensure a 100-150 nm penetration depth, achieving precise optical sectioning at the basal plane.
  - Nine raw images (three angles and three phases of structured illumination) were captured per time point for the purpose of super-resolution imaging.
- **Image Processing**:
  - Acquired images were processed and reconstructed using the TIRF-SIM reconstruction algorithm described in [51] to enhance the resolution.
  - A consistent Wiener filter parameter of 0.05 was applied across all image reconstructions.

3.4 Data Format

Let us describe how the microscopic images used in this work were structured. The format of the image data is as follows:

- **Data Structure**: Each cell image is presented as a stack consisting of 120 frames.
- **Frame Dimensions**: Individual frames are sized at 1024 by 1024 pixels, providing a detailed view at a high resolution.
- **Channel and Bit Depth**: The images are captured in a single channel format with a depth of 16-bit floating point numbers, ensuring a high degree of precision in pixel values.
- **File Format**: All images are stored in the MRC (Medical Research Council) format [16]. This format is extensively utilized in the field of electron microscopy and various other microscopic disciplines. It is particularly adept at managing multidimensional datasets and also allows for the integration of various metadata.

3.5 Data Normalization

Data normalization plays a crucial role in the preprocessing of microscopic image stacks. For this thesis, each frame within the image stack was normalized individually, adhering to specific criteria:

- **Normalization Process**: The normalization of each frame is conducted such that its mean (average pixel value) is adjusted to 0, and the standard deviation of pixel values is normalized to 1. This process is mathematically represented as:

$$I_{\text{normalized}} = \frac{I - \mu}{\sigma}$$

  where I is the original pixel value, μ is the mean pixel value of the frame, and σ is the standard deviation of pixel values in the frame.

- **Rationale for Frame-by-Frame Normalization**:
  - *Fluctuating Intensity*: Due to flickering effects, there is a fluctuation in the mean intensity of the frames. Normalizing each frame individually accounts for these variations, ensuring consistency across the image stack.
  - *Photobleaching*: In the context of TIRF microscopy, photobleaching refers to the gradual loss of fluorescence in the sample due to prolonged exposure to the excitation light. This phenomenon leads to a decrease in signal intensity over time. Individual frame normalization compensates for the intensity loss due to photobleaching, maintaining uniformity in the data.

The application of this normalization technique ensures that each frame in the stack is independently adjusted for any variations in lighting or photobleaching effects, thereby enhancing the reliability and comparability of the data across different frames.

Histogram of the pixel values can be seen in Figure 3.2.

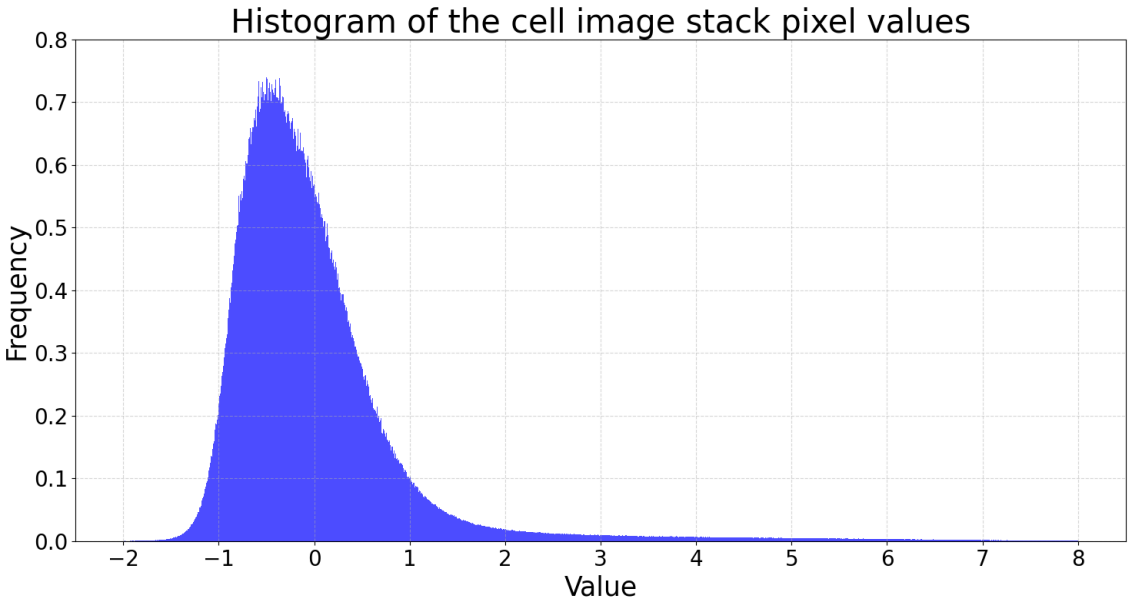


Figure 3.2: A histogram of pixel values constructed over the whole cell image stack after the frame-by-frame data normalization.

3.6 Data Annotation

The process of data annotation was designed to ensure accurate representation of biological processes in the microscopic images. The following steps were taken in the annotation process:

- **Cell Selection**: A single cell, demonstrating successful endocytosis throughout the entire lifespan of the image stack (spanning all 120 frames), was selected for detailed study.
- **Region Selection in the Cell**: Within this chosen cell, a specific region was selected for annotation. This region constituted 1/16th of the total area of the cell's image.
- **Annotator Team**: Three human annotators, each with a comprehensive understanding of the biological processes involved, worked independently to ensure unbiased and varied data annotation.
- **Annotation Task**:
  - The primary task for each annotator was to identify all CCPs within the selected 1/16th region of the cell.
  - The annotators were instructed to track these CCPs throughout their entire lifetime. This included instances where a CCP might drift into or out of the selected region. A CCP was to be tracked from the start to the end of its lifecycle, provided it appeared within the selected region in at least one frame.
- **Annotation Tool**: The annotators utilized a custom-built annotation tool developed in MATLAB specifically for this task. This tool was designed to facilitate the precise marking and tracking of CCPs within the microscopic images.
- **Annotation Output**: The output from each annotator was compiled into a file, presenting all annotated CCP tracks in a tabular format. Mathematically, each CCP track is represented as a sequence of triplets:

$$((f_i, x_i, y_i) \mid i = 1, 2, ..., n)$$

where f_i denotes the frame number, and (x_i, y_i) represents the Cartesian coordinates of the center of the CCP in that specific frame. Here, n signifies the total number of frames in which the CCP is tracked. This structured data format aids in the comparative analysis and validation of the results of this thesis.

Chapter 4

Clathrin-Coated Pits Detection Algorithm

4.1 Detection Algorithm Overview

The core of the CCP detection system is a U-Net architecture, trained in per-pixel regression task, where the presence of CCPs is indicated by local maxima in target images to which the model is fitted. These target images are spatially continuous, a feature that enhances the neural network's pixel-wise predictive capability. Given the limited size of the dataset, regularization techniques such as dropout and weight decay are implemented to prevent overfitting and to foster the generalizability of the model.

The operational principle of the detector is straightforward: the U-Net receives TIRF-SIM images as input, with the expectation that the output's local maxima align with the CCP locations. The identification of these maxima is subject to a threshold criterion; only those exceeding a predefined threshold are considered. Furthermore, a pruning step ensures that selected maxima are sufficiently spaced, eliminating any that are in close proximity to one another. The local maxima that meet these criteria—above the threshold and suitably isolated—are designated as the predicted CCPs.

Validation of the detector's accuracy is conducted through a cross-validation regime, enabling a comprehensive assessment across the entire dataset. This approach confirms the effectiveness of the applied regularization techniques during the training phase. Additionally, the optimal number of temporal frames to be included as input channels is determined through validation metrics, enhancing the detector's temporal context sensitivity. Lastly, the detection threshold is fine-tuned by maximizing the F-score, a strategy aimed at achieving a detection performance balance between specificity and sensitivity.

4.2 Target Data Preparation

This subsection addresses the preparation of targets for the neural network detector. The target resolution and size are designed to match the input dimensions, specifically 120 frames, each of 1024 by 1024 pixels.

Consider a specific frame within the input image data stack. Let $((x_i, y_i) \mid i = 1, 2, ..., n)$ represent the positions of all CCPs as annotated by a particular annotator in this frame, where n denotes the number of targets annotated by the annotator at this frame. For each annotated position (x_i, y_i), we define the functions τ_i as follows:

$$\tau_i(x, y) = \begin{cases} 1 - \frac{\sqrt{(x-x_i)^2+(y-y_i)^2}}{r}, & \text{if } \sqrt{(x-x_i)^2 + (y-y_i)^2} < r \\ 0, & \text{otherwise} \end{cases}$$

where r is a predefined radius parameter. The function τ_i exhibits a cone-like profile, peaking at value 1 at the target position (x_i, y_i) and decaying to 0 within a distance r. The function's profile is depicted in Figure 4.1.

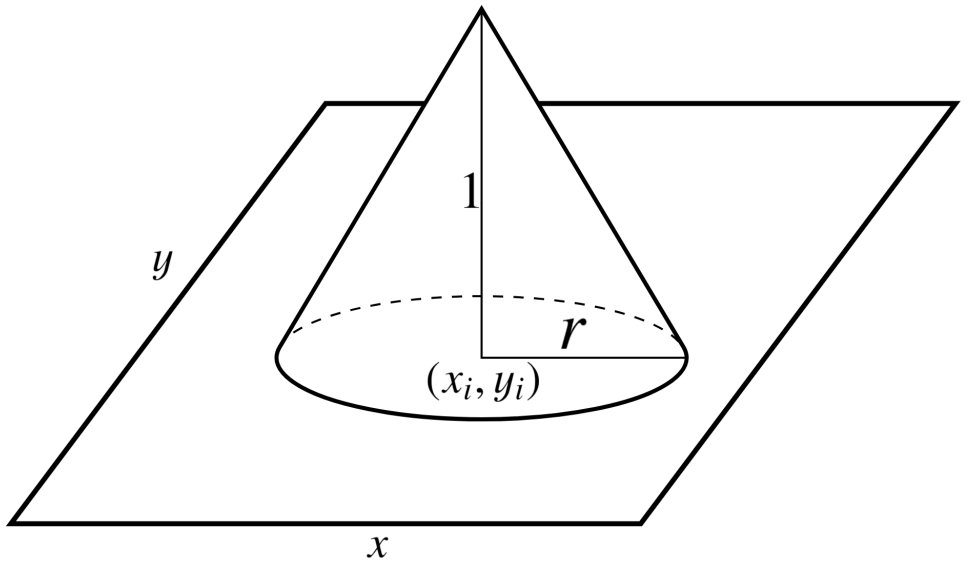


Figure 4.1: The profile of function τ_i is conical, centered at (x_i, y_i) with radius r.

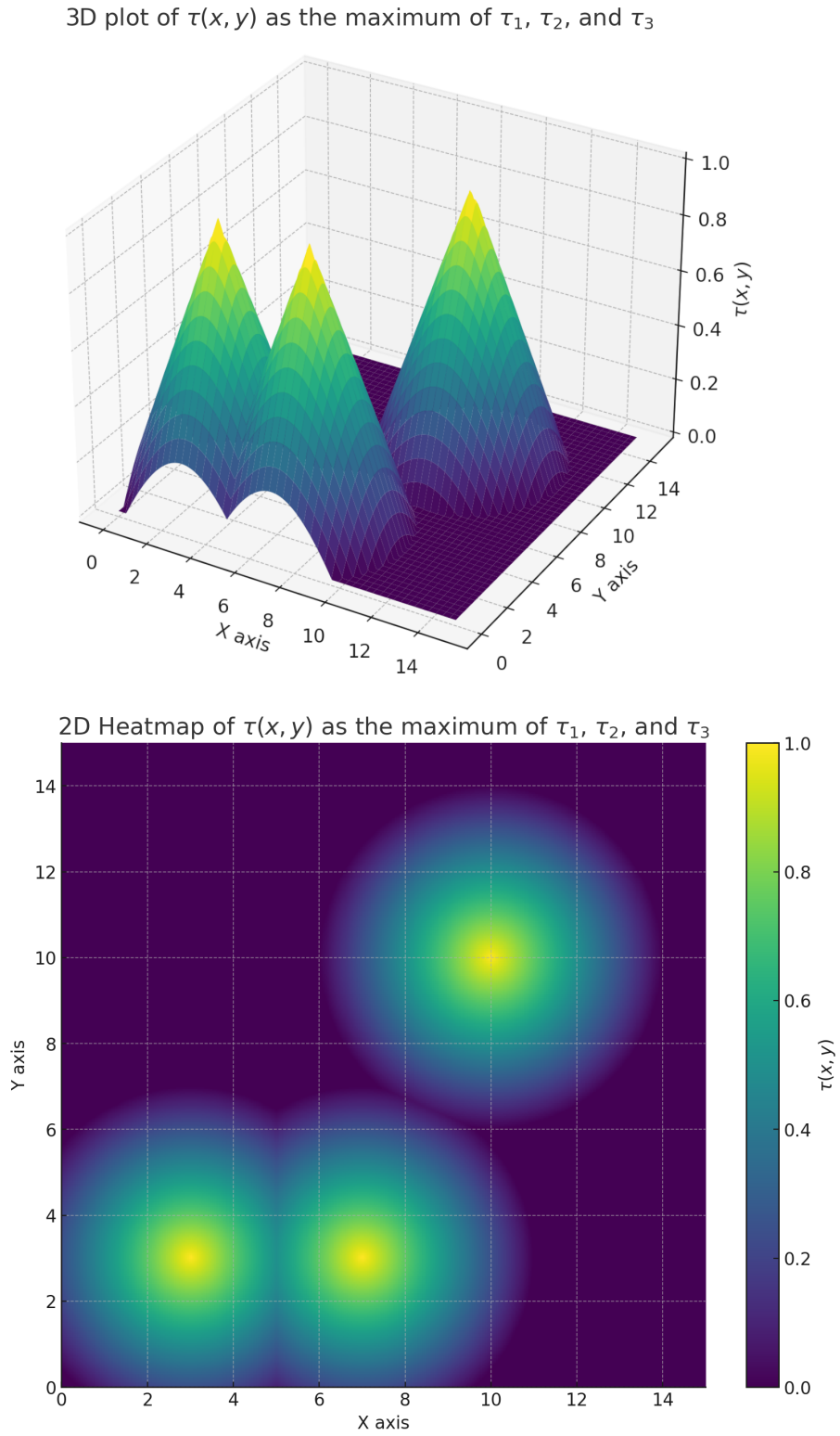


Figure 4.2: Plot of particular example of function τ, where $n = 3$, The target positions are $(3, 3), (7, 3), (10, 10)$ and $r = 4$. Top: 3D plot, Bottom: 2D heatmap of the same function.

Subsequently, we define an overarching function τ for the same frame and annotator as:

$$\tau(x, y) = \max_i \tau_i(x, y)$$

The visualization of function τ is presented in Figure 4.2. The function τ, defined as the maximum of the continuous functions τ_i, exhibits several key properties:

- **Continuity**: τ is continuous as it is the maximum of a set of continuous functions τ_i, adhering to the principle that the maximum of continuous functions retains continuity.
- **Local Maxima**: All the local maxima of τ have value of 1 and lie at the target positions (x_i, y_i), aligning with the annotated CCP locations.
- **Absence of Other Local Maxima**: τ does not have any other local maxima, focusing solely on the annotated target positions.
- **Invariance to Rotations and Reflections**: The creation of τ is invariant under rotations and reflections, ensuring the preservation of its properties and shape regardless of image orientation.

To combine the τ functions from three independent annotators, we use the arithmetic mean. Let $\tau^{(\text{Annotator 1})}$, $\tau^{(\text{Annotator 2})}$, and $\tau^{(\text{Annotator 3})}$ denote the τ functions from the first, second, and third annotators, respectively. The combined function $\tau^{(\text{Combined})}$ is then computed as:

$$\tau^{(\text{Combined})}(x, y) = \frac{1}{3}\left(\tau^{(\text{Annotator 1})}(x, y) + \tau^{(\text{Annotator 2})}(x, y) + \tau^{(\text{Annotator 3})}(x, y)\right)$$

At each frame, the function $\tau^{(\text{Combined})}$ represents the target to which the model is fitted.

A value $r = 4$ was selected for our purpose since matured CCPs have an apparent radius of approximately 4 pixels in the TIRF-SIM images.

4.3 Neural Network Architecture

The architecture of the CCP detector neural network, structured around the U-Net model, offers significant advantages for the detection of CCPs in microscopy images. Notably, the network's output dimensions are identical to its input dimensions, which is crucial for precise per-pixel analysis. The network incorporates convolutional blocks and pooling layers. The key components of this architecture are:

- **Input-Output Symmetry**: One of the primary advantages of the U-Net architecture in our application is its ability to produce output with the same spatial dimensions as the input. This feature is critical for per-pixel target fitting in CCP detection.
- **Contracting Path**: Comprising a series of convolutions, activation functions, and pooling layers, the contracting path effectively captures the context and salient features of the input images. The layer-wise doubling of feature channels enriches the feature extraction process.
- **Bottleneck**: Serving as a bridge between the contracting and expanding paths, the bottleneck processes the most abstract features of the input data, ensuring the seamless transition from feature extraction to feature mapping.

- **Expanding Path**: Mirroring the contracting path, the expanding path uses up-sampling and concatenation with feature maps from the contracting path. This design is essential for restoring the spatial resolution of the output, aligning it precisely with the input dimensions.
- **Final Convolution Layer**: The network culminates in a 1x1 convolution layer that maps the multi-channel feature data to the desired 1-channel output format.
- **Relevance in CCP Detection**: The U-Net's design, particularly its output dimensionality matching the input, makes it an ideal choice for CCP detection. This ensures that every pixel in the output can be directly correlated with the corresponding pixel in the input image, crucial for detailed and accurate CCP analysis.

Overall, the U-Net architecture is optimized not just for feature extraction but also for maintaining the spatial dimensions of the input, which is vital for the detailed per-pixel analysis required in CCP detection tasks.

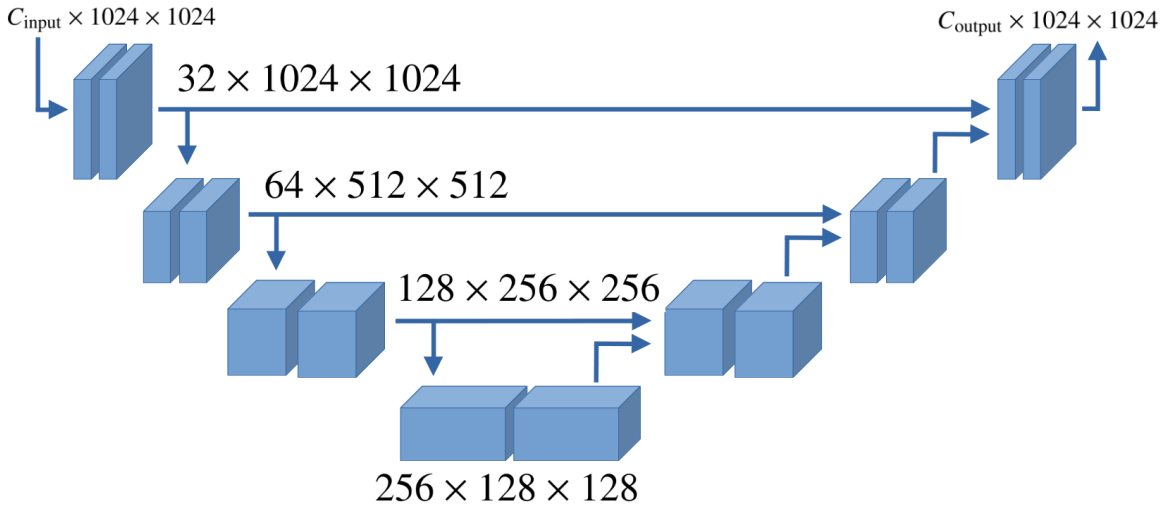


Figure 4.3: Illustration of the U-Net architecture showcasing the flow of tensor dimensions through the contracting and expanding paths. The batch dimension is ignored here. The spatial dimensions illustrated are 1024 by 1024, but can be any multiples of 8. C_{input} denotes the number of input channels and C_{output} denotes the number of output channels.The network incorporated convolutional blocks and pooling layers, decreasing spatial dimensions while increasing feature depth. This is followed by a series of up-convolutions and concatenations, reversing the process to return to the original spatial dimensions.

The overall architecture of the network is depicted in Figure 4.3. Each pair of cuboids in the figure represents a convolution block. The structure of such a convolution block is represented in Figure 5.1.

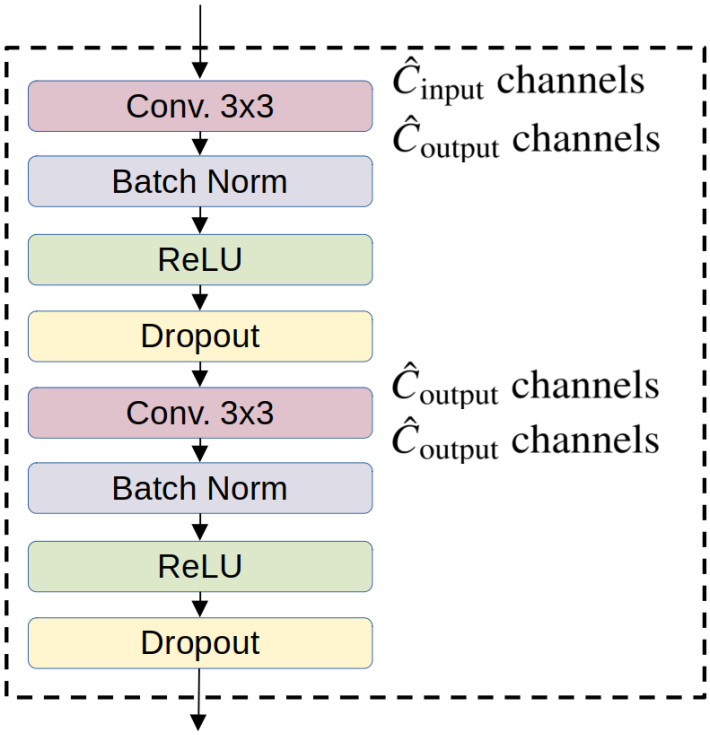


Figure 4.4: Structure of the convolution block corresponding to each pair of immediately adjacent cuboids in Figure 4.3. $\hat{C}_{\text{input}}$ denotes the number of input channels of the block and $\hat{C}_{\text{output}}$ denotes the number of output channels of the block.

The convolution block consists of layers that perform consecutive operations: convolution, batch normalization, activation, and dropout. The convolution layers apply filters to the input to extract features, while the ReLU activation function introduces non-linearity, allowing the network to capture complex patterns.

Batch normalization follows convolution, normalizing the layer outputs to improve the stability and speed of the network's training. It achieves this by addressing internal covariate shift, making the learning process less sensitive to the learning rate and other hyperparameters.

Dropout is used as a regularization technique after the activation function. By randomly setting a fraction of the input units to zero during training, specified by the dropout_prob parameter, it prevents neurons from co-adapting too much. This technique effectively reduces overfitting, forcing the network to learn more generalized features that are not reliant on specific weights, enhancing the network's ability to perform well on unseen data.

The code defining the architecture in PyTorch [6, 7] follows:

Code segment 4.1: UNet Class Implementation in PyTorch

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class UNet(nn.Module):
    def __init__(self, in_channels, out_channels, dropout_prob=0.5):
        super(UNet, self).__init__()
        # Contracting Path
        self.enc1 = self.conv_block(in_channels, 32, dropout_prob)
        self.enc2 = self.conv_block(32, 64, dropout_prob)
        self.enc3 = self.conv_block(64, 128, dropout_prob)
        self.pool = nn.MaxPool2d(2)
        # Bottom layer
        self.bottleneck = self.conv_block(128, 256, dropout_prob)
        # Expanding Path
        self.upconv3 = nn.ConvTranspose2d(256, 128, kernel_size=2, stride=2)
        self.dec3 = self.conv_block(256, 128, dropout_prob)
        self.upconv2 = nn.ConvTranspose2d(128, 64, kernel_size=2, stride=2)
        self.dec2 = self.conv_block(128, 64, dropout_prob)
        self.upconv1 = nn.ConvTranspose2d(64, 32, kernel_size=2, stride=2)
        self.dec1 = self.conv_block(64, 32, dropout_prob)
        self.out = nn.Conv2d(32, out_channels, kernel_size=1)
    def forward(self, x):
        # Contracting Path
        enc1 = self.enc1(x)
        enc2 = self.enc2(self.pool(enc1))
        enc3 = self.enc3(self.pool(enc2))
        # Bottom layer
        bottleneck = self.bottleneck(self.pool(enc3))
        # Expanding Path
        dec3 = self.upconv3(bottleneck)
        dec3 = torch.cat((enc3, dec3), dim=1)
        dec3 = self.dec3(dec3)
        dec2 = self.upconv2(dec3)
        dec2 = torch.cat((enc2, dec2), dim=1)
        dec2 = self.dec2(dec2)
        dec1 = self.upconv1(dec2)
        dec1 = torch.cat((enc1, dec1), dim=1)
        dec1 = self.dec1(dec1)
        return self.out(dec1)
    def conv_block(self, in_channels, out_channels, dropout_prob):
        block = nn.Sequential(
            nn.Conv2d(in_channels, out_channels, kernel_size=3, padding=1),
            nn.BatchNorm2d(out_channels),
            nn.ReLU(inplace=True),
            nn.Dropout(dropout_prob),  # Add dropout after activation
            nn.Conv2d(out_channels, out_channels, kernel_size=3, padding=1),
            nn.BatchNorm2d(out_channels),
            nn.ReLU(inplace=True),
            nn.Dropout(dropout_prob)  # Add dropout after activation
        )
        return block
```

4.4 Data Loading and Augmentation

The process of data loading and augmentation is crucial for preparing the input for the neural network. This procedure involves several steps to ensure that the data fed into the model is representative and varied, enhancing the model's ability to generalize. The specific steps are as follows:

- **Selection of Image Region**: A randomly rotated square region is randomly placed in a particular image frame. This region is chosen such that it lies entirely within the designated 1/16th area of the cell where annotators marked the data (illustrated as a green square in Figure 4.5). The initial selection is indicated by a dashed red line in the figure.
- **Margin Addition**: To address potential issues at the borders of the images due to padding in the convolution layers of the U-Net, the initially selected square region is expanded by a certain margin. The resulting region for loading, which is larger than the initial square, is depicted with a solid red line in Figure 4.5.
- **Rotation and Interpolation**: The selected region undergoes rotation. The transformation applied during this rotation uses bicubic interpolation, ensuring smooth transitions in the rotated image.
- **Flipping**: With a probability of 0.5, the loaded region is also flipped horizontally. This step introduces further variability into the dataset, which is beneficial for training robust models.
- **Target Loading**: The targets corresponding to the input images are loaded using the exact same transformations (rotation, margin addition, and flipping). This ensures that the target data aligns precisely with the transformed input data, maintaining the integrity of the annotations.

These steps in data loading and augmentation play a pivotal role in preparing the dataset for effective training of the neural network. By introducing variations through rotation, flipping, and margin adjustments, the model is better equipped to handle diverse scenarios.

In addition to the spatial transformations previously described, we propose an approach to enhance the contextual understanding of the neural network by loading multiple temporal frames as channels. This method is hypothesized to provide the detector with additional temporal context, potentially improving its performance in identifying targets. The process is as follows:

- **Temporal Frame Selection**: Let k denote the current frame being loaded, and m represent the total number of frames to be loaded as channels. The selected frames range from $k - \lceil m/2 \rceil$ to $k + \lfloor m/2 \rfloor$, inclusive. This ensures that m frames are loaded, centered around the frame k along the temporal axis. (Or approximately centered if m is even.)
- **Temporal Padding**: In cases where the frame range $k - \lceil m/2 \rceil$ to $k + \lfloor m/2 \rfloor$ extends beyond the available frame indexes (1 to 120), temporal padding is applied to maintain consistency. This padding is performed using 'edge padding', where frames indexed 0, -1, and so forth are treated as duplicates of frame 1, while frames indexed 121, 122, and beyond are considered duplicates of frame 120. This approach ensures that the neural network always receives m frames as input, regardless of the position of k within the image stack.

This method of loading multiple temporal frames as channels aims to provide the neural network with a richer set of data, encompassing not only the spatial information within a single frame but also the temporal context across multiple frames. By doing so, we anticipate an improvement in the detector's ability to accurately identify and track dynamic processes occurring over time.

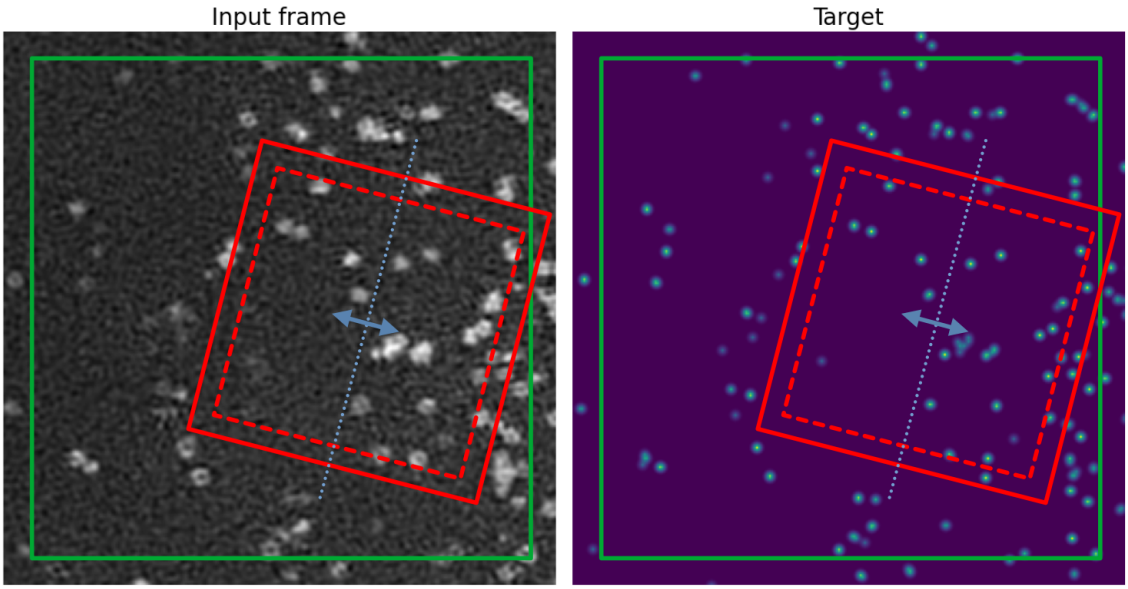


Figure 4.5: Illustration of the data loading and augmentation process. The initial selection (dashed red) is a randomly rotated square, which is then randomly placed in the annotated 1/16th area (green). The region to be loaded is then expanded by a margin (solid red), possibly outside the green region. The region to be loaded is flipped with a probability of 0.5. The region and margin sizes are not necessarily precisely to scale.

In the augmentation of the dataset, additional techniques are incorporated to introduce variability, thus enhancing the neural network's robustness. These methods involve alterations in the statistical properties of the data and are described as follows:

- **Variation in Mean and Standard Deviation**:
  - For each data point, random variations are applied to the mean and standard deviation. The variation factor, denoted as δ, is drawn from a uniform distribution, $\delta \sim \mathcal{U}(-\alpha, \alpha)$, where $\alpha = 0.2$. Consequently, the modified mean and standard deviation are given by $\mu' = \mu + \delta \cdot \sigma$ and $\sigma' = \sigma + \delta \cdot \sigma$, where μ and σ are the original mean and standard deviation, respectively.

- **Addition of Gaussian Noise**:
  - Gaussian noise is added to the input, where the noise is characterized by a mean of zero and a variable standard deviation. The standard deviation of the noise, σ_{noise}, is drawn from an exponential distribution, $\sigma_{\text{noise}} \sim Exp(\lambda)$, where the rate parameter λ is set such that the mean of the exponential distribution equals $\beta \cdot \sigma$, with $\beta = 0.1$. This implies $\lambda = 1/(\beta \cdot \sigma)$.

These statistical augmentations to the dataset are designed to simulate a range of natural variations, thereby preparing the neural network to generalize effectively from the training data to new, unseen datasets.

4.5 Training the Detector

The training of the neural network was conducted using a dataset partitioned into training and testing sets. This section details the training process, loss function, and optimization algorithm.

- **Dataset Partitioning**:
  - The dataset was divided into two distinct sets: the first 100 frames were allocated for training, and the subsequent 20 frames were reserved for testing. This partitioning ensures a balanced distribution of data for both training and evaluation purposes.
- **Loss Function**:
  - The Mean Square Error (MSE) was employed as the loss function. Mathematically, the MSE for a single data point is defined as:

$$\text{MSE} = \frac{1}{N}\sum_{x,y}(p(x,y) - \tau(x,y))^2$$

    where N is the number of pixels in the region over which MSE is computed, $\tau(x,y)$ is the target value, and $p(x,y)$ is the predicted value by the network.
  - It is important to note that the MSE was computed only over the region of the image that excludes the margin, as described in the data loading process (the region denoted with dashed red line in Figure 4.5). The size of this square region over which the MSE is computed was set to 64×64, and the margin was set to 32.
- **Optimization Algorithm**:
  - The optimization of the network was performed using the Adam algorithm [10], with the following parameters: learning rate $lr = 1 \times 10^{-4}$, weight decay $= 1 \times 10^{-3}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$.
- **Batch Size and Training Duration**:
  - The batch size was set to 8, meaning each batch contained 8 randomly augmented frames from the training set, in accordance with the augmentation methods outlined in the previous chapter.
  - The frame indices for loading were randomly shuffled at the start of each epoch. There were $\left\lceil \frac{100}{8} \right\rceil = 13$ batches per training epoch.
  - The network underwent training over 1000 epochs, ensuring adequate exposure to the training data for learning and convergence.

As depicted in Figure 4.6, the Mean Square Error (MSE) loss values for both the training and validation sets are plotted over the course of 1000 epochs. A discernible convergence of loss values is observed around the 300-epoch mark. Beyond this point of convergence, the training loss fluctuates around an approximate value of 0.02 with a standard deviation of 2×10^{-3}, while the validation loss shows fluctuations around an approximate value of 0.04 with a standard deviation of 8×10^{-3}. This stabilization indicates that the network has effectively learned the underlying patterns and suggests a robustness in prediction, with reasonably low overfitting as indicated by the relatively close tracking of training and validation loss values.

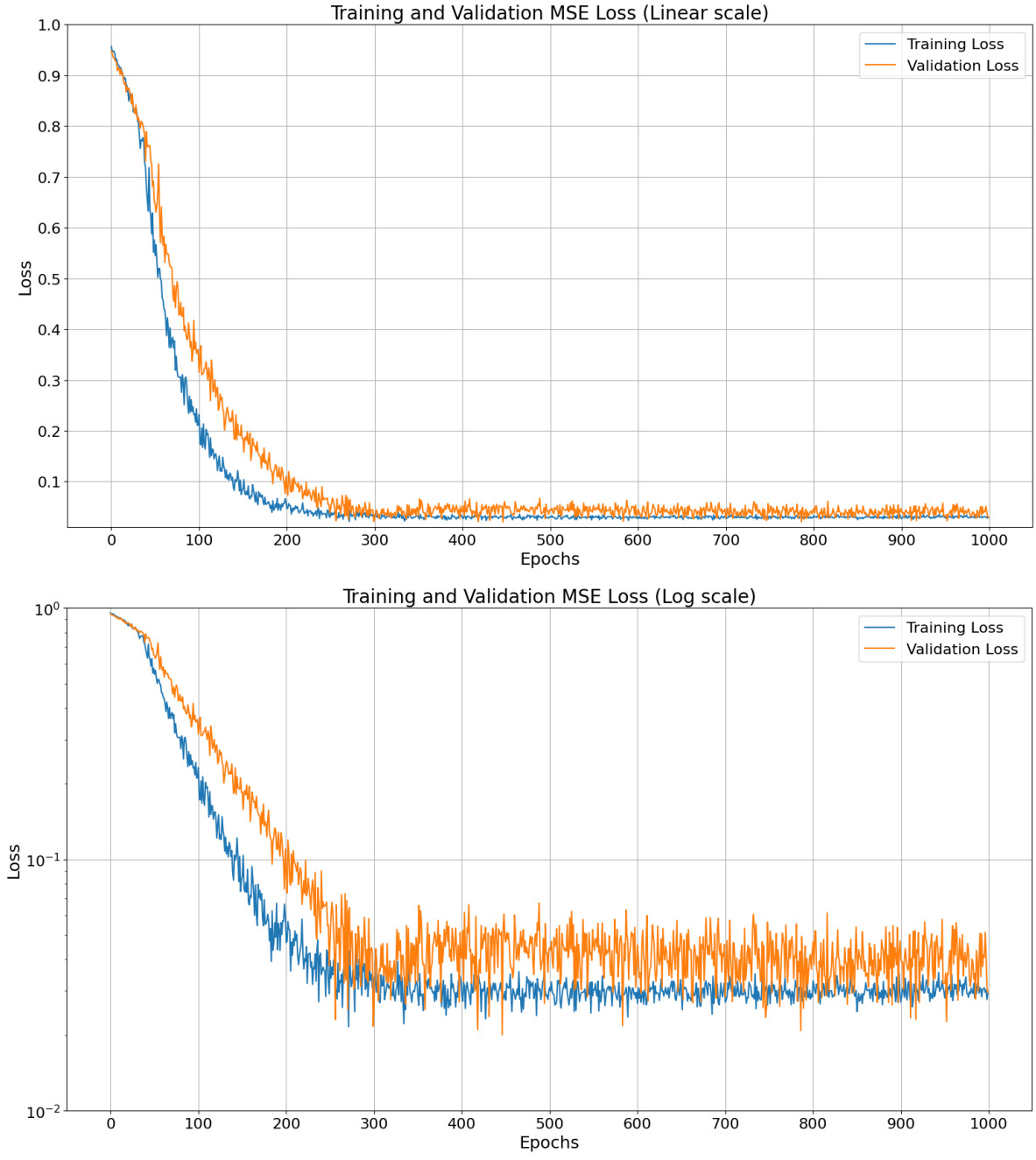


Figure 4.6: Evolution of training and validation MSE loss over 1000 epochs. A convergence trend is observed after approximately 300 epochs, after which the loss values stabilize. Top: linear y scale, Bottom: logarithmic y scale for the same plot.

4.6 Detector Evaluation and Parameter Finetuning

The evaluation of the detector's performance employed a cross-validation strategy to ensure comprehensive assessment across the entire dataset. The dataset, comprising 120 frames, was divided into six groups of 20 frames each. In this cross-validation scheme, often referred to as 'folds', six iterations of the detector training were conducted, with each iteration withholding a different group of frames from training to be used as a validation set.

For evaluation, the outputs of the detectors were aggregated in a manner that the combined prediction output is entirely based on validation data, thus eliminating any training data influence. Within each frame, local maxima were identified where a pixel is considered a local maximum if it has a higher value than all of its four immediate neighbors. Only maxima surpassing a threshold ρ were selected. This threshold is subject to optimization, which will be described later.

The maxima were then sorted in descending order of their value. Starting from the highest, any maxima within a 3-pixel radius were eliminated from the list. This is expressed in Algorithm 4. This culling process ensures the preservation of only the most significant maxima, which were then compared against the ground truth annotations by human annotators.

Algorithm 4 Identification and Selection of Local Maxima

```
maxima ← empty list
for each frame in the dataset do
    for each pixel (x, y) in the frame do
        if pixel value > all four immediate neighbors' values then
            if pixel value > threshold ρ then
                Append (x, y, pixel value) to maxima
            end if
        end if
    end for
    Sort maxima in descending order of pixel values
    selectedMaxima ← empty list
    for each (x, y, value) in maxima do
        if no selected maximum within a 3-pixel radius of (x, y) then
            Append (x, y) to selectedMaxima
        end if
    end for
end for
```

The rationale behind the 3-pixel radius for local maximum suppression and the optimization of the threshold ρ are rooted in biological constraints: it is biologically implausible for CCPs to overlap, and their size consistently exceeds 3 pixels. The optimization of ρ will be addressed later; it is critical to finetune the detector's sensitivity and specificity.

For each frame, detector detections were matched with ground truth annotations provided by each of the three annotators. A detection was considered a match to a ground truth target if their Euclidean distance was less than 3 pixels. In instances where multiple ground truth targets were in proximity, a greedy algorithm was employed to associate the detection with the nearest ground truth target based on the minimal Euclidean distance, ensuring a one-to-one correspondence where each detector detection is associated with at most one ground truth target and vice versa.

This matching process was systematically applied across all frames and annotators, tallying the instances of True Positives (TP), False Negatives (FN), and False Positives (FP). A True Positive reflects

a detector detection correctly associated with a ground truth annotation; a False Negative indicates a missed detection by the detector where the annotator has identified a target; and a False Positive denotes an erroneous detection by the detector not corroborated by the annotator. The counts for TP, FP, and FN were aggregated across all frames and annotators to yield the total numbers for each category.

Subsequent to the aggregation of TP, FP, and FN counts, Precision and Recall metrics were computed as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Precision measures the proportion of correct detections among all detections made by the detector, while Recall assesses the proportion of actual targets that were correctly detected.

The aforementioned evaluation was conducted iteratively for varying values of the threshold ρ, which was systematically altered from 0 to 1 in increments of 0.01. This variation enabled the generation of Precision-Recall curves, which are instrumental in identifying the optimal threshold value that balances the trade-off between Precision and Recall, effectively optimizing the detector's performance.

The initial configuration of the U-Net architecture sets m, the number of temporal input channels, to 1. Optimization of this parameter is deferred to subsequent iterations of model refinement.

In an effort to curtail overfitting—particularly pressing given the modest size of the dataset—regularization strategies were implemented within the U-Net framework. The network employs dropout with a probability of 0.5 and is trained with a weight decay of 1×10^{-3}. These regularization techniques are essential in preventing the model from overfitting to the training data.

To evaluate the efficacy of these regularization methods, two sets of network instances were trained: one set with dropout and weight decay (regularized) and another without these features (not regularized). Both sets were subjected to the same cross-validation process, and their performances were assessed. Precision and recall curves were calculated for each set based on the outputs on the training datasets, facilitating a comparative analysis of model behavior on training versus validation data, thereby quantifying the degree of overfitting.

As depicted in Figure 4.7, the precision-recall curves for the regularized and not regularized models are presented for both the training and validation datasets. Notably, the regularized model demonstrates a decrease in performance on the training data but improved performance on the validation data. This contrast signifies the successful application of regularization, confirming an enhancement in the model's ability to generalize and a reduction in the risk of overfitting.

Subsequent to regularization assessment, the parameter m, representing the number of temporal input channels, was subjected to optimization. A series of evaluations was conducted using cross-validation on the regularized model to ascertain the optimal number of temporal frames to include as input channels. Four different values of m—1, 2, 3, and 5—were tested to determine their impact on the detector's performance.

The outcomes of these evaluations are presented in Figure 4.8. For clarity, a single precision-recall curve corresponding to the training dataset for $m = 1$ is included as a point of reference. The additional training curves for $m = 2, 3$, and 5 are omitted due to their near-identical nature, which would otherwise offer no discernible distinction on the graph.

The analysis revealed a measurable improvement in performance with $m = 3$ as opposed to $m = 1$ or $m = 2$, suggesting that the inclusion of three temporal frames strikes an optimal balance for model efficacy. Conversely, no further improvement was observed when m was increased to 5. Therefore, based on these findings, m was set to 3 for subsequent use in the detector.

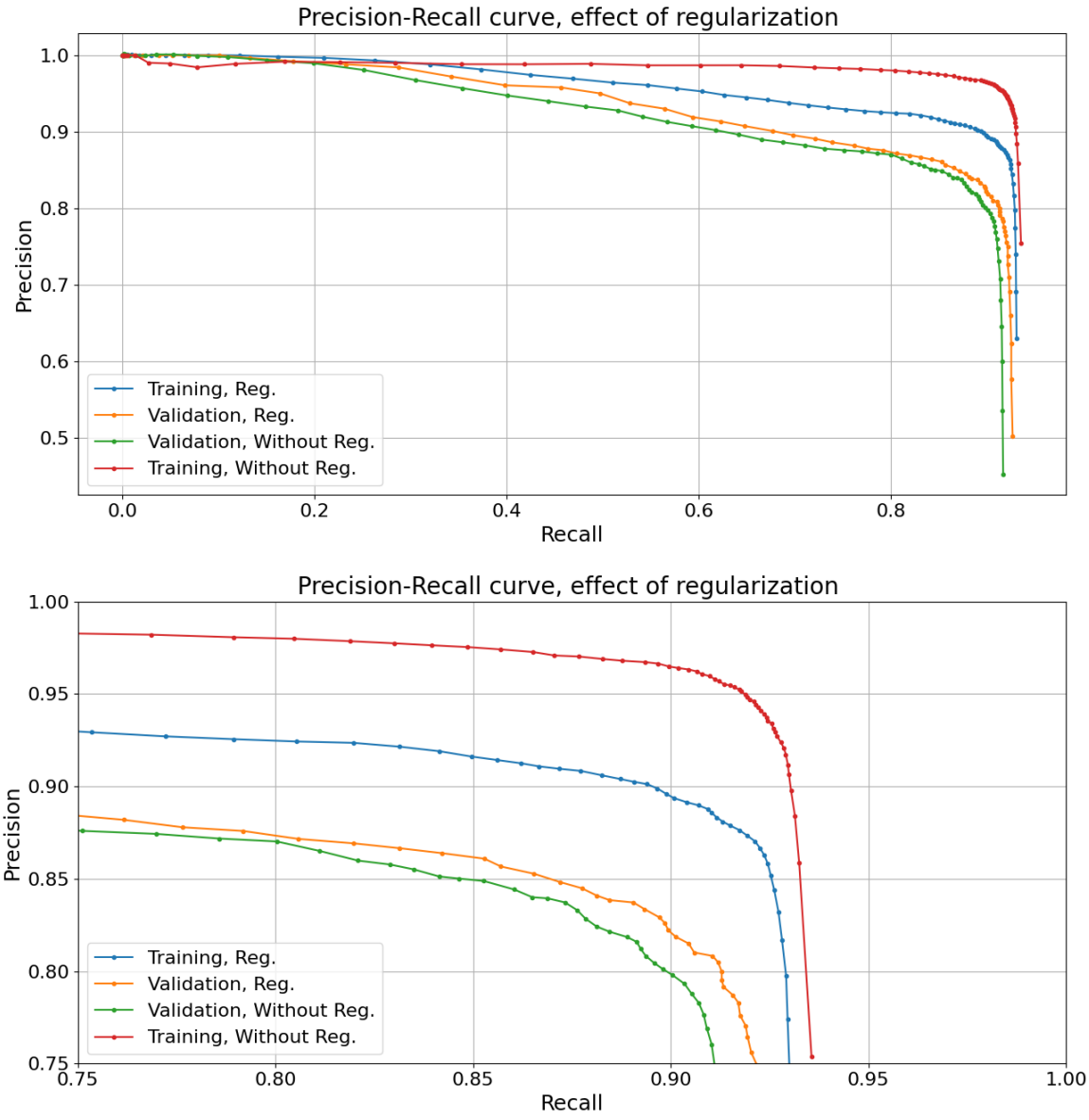


Figure 4.7: Precision-Recall curves illustrating the effect of regularization on the U-Net architecture. Four curves are depicted: Regularized on Validation, Regularized on Training, Not Regularized on Validation, and Not Regularized on Training. The bottom plot has identical content, only a higher zoom.

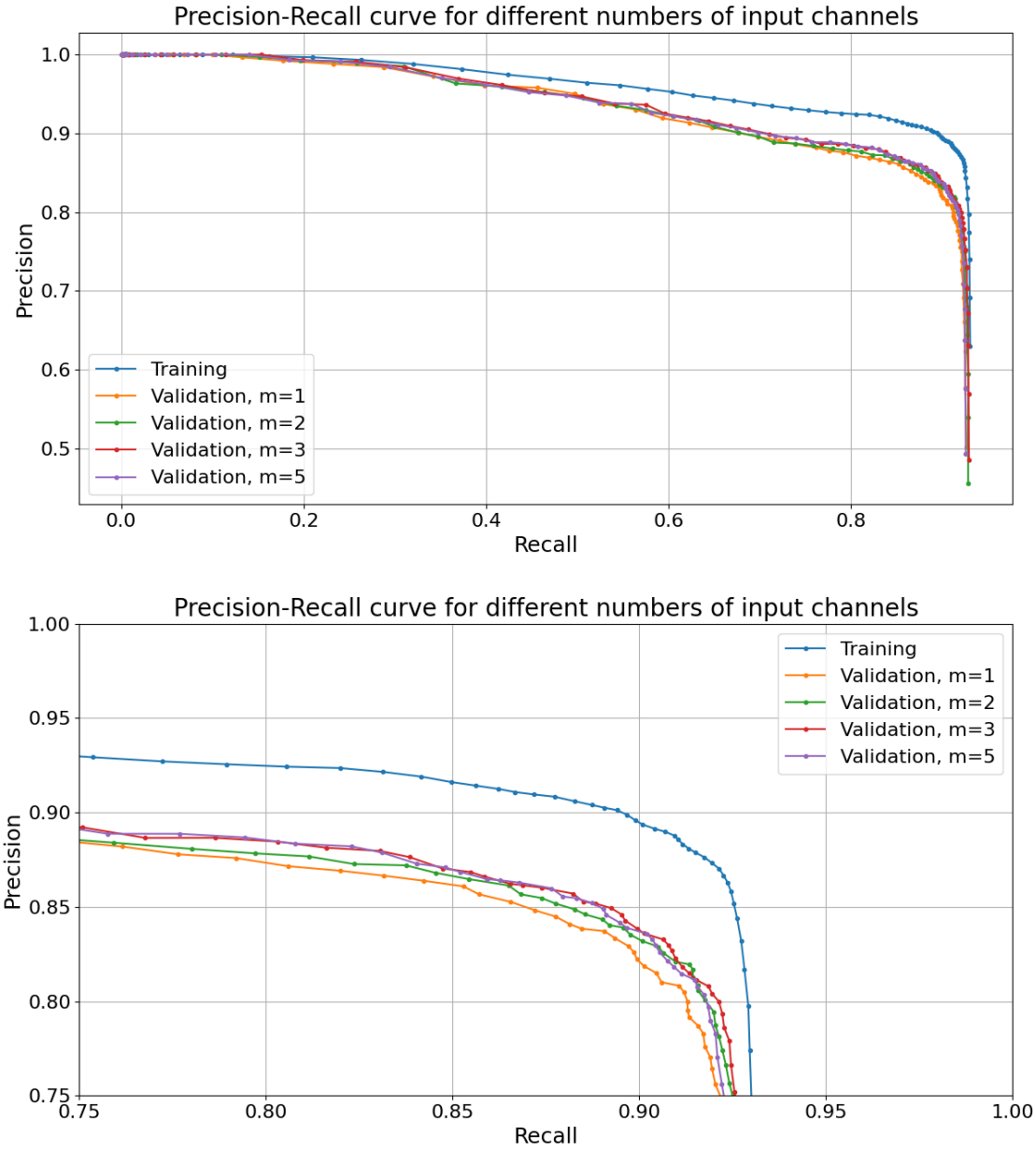


Figure 4.8: Precision-recall curves for different values of m on the validation dataset, alongside the curve for $m = 1$ on the training dataset for reference. The bottom plot has identical content, only a higher zoom.

With the number of temporal input channels set to $m = 3$, the subsequent phase involved optimizing the threshold parameter ρ. This parameter is pivotal for the selection of local maxima from the network's output. To this end, ρ was varied from 0 to 1 in increments of 0.01. Corresponding to each value of ρ, precision and recall were computed within the cross-validation setup. The F-score, which is defined as the harmonic mean of precision and recall, was also calculated for each threshold value using the formula:

$$\text{F-score} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The choice of the F-score as the optimization metric is justified by its sensitivity to the balance between precision and recall. When precision and recall are nearly equal, the F-score approaches their arithmetic mean, thus reflecting an even balance. However, if either precision or recall is significantly low, the F-score is heavily penalized, more so than by the arithmetic mean. This characteristic makes the F-score a suitable metric for maximization, as it ensures that both false positives and false negatives are simultaneously minimized.

The graphical representation of precision, recall, and the F-score as functions of ρ is presented in Figure 4.9. The optimal value of ρ is determined by the maximum F-score. This optimum is achieved at $\rho = 0.36$, corresponding to a precision of 0.849, a recall of 0.892, and an F-score of 0.870, indicating a balanced trade-off between precision and recall.

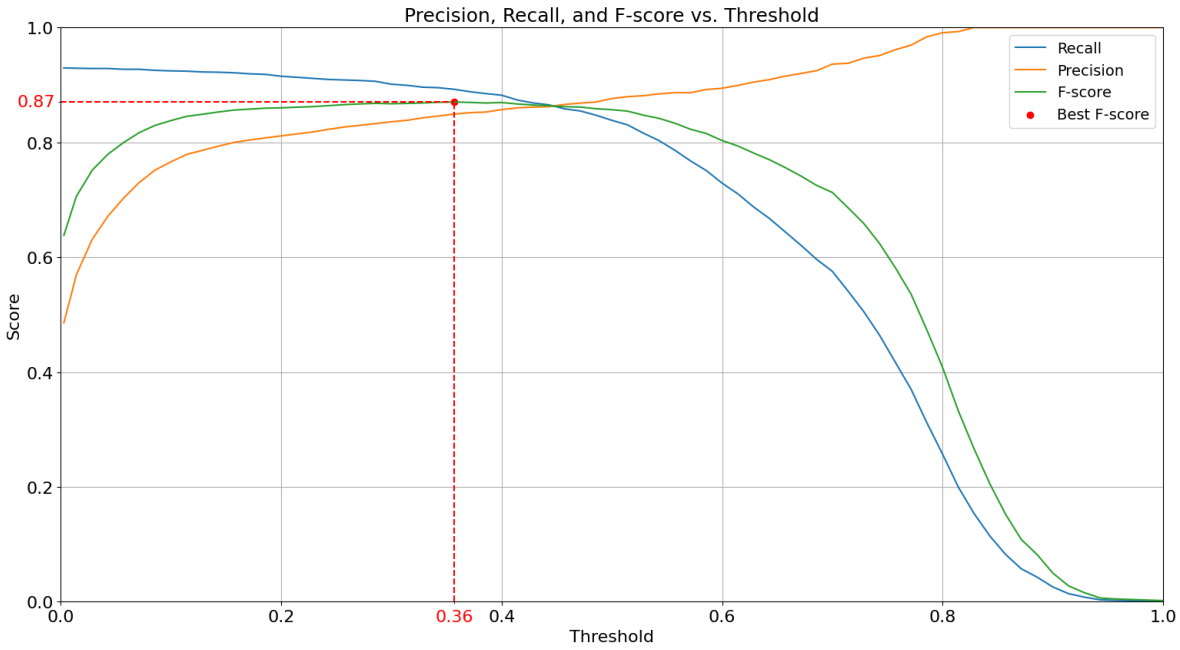


Figure 4.9: Variation of precision, recall, and the F-score with the threshold parameter ρ. The optimal threshold is identified by the peak F-score, denoting a balanced compromise between precision and recall.

The maximization of the F-score at the selected threshold confirms the optimal balance between false positives and false negatives, which is crucial for the accurate detection of CCPs by the network.

Chapter 5

Clathrin-Coated Pits Tracking Algorithm

5.1 Chapter Overview

This chapter delves into the intricacies of developing and evaluating a tracking algorithm specifically designed for CCPs observed in TIRF-SIM microscopy images. The chapter is structured into several key sections, each addressing a different aspect of the algorithm's development and performance assessment:

- **Measuring Characteristics of CCP Motion:** This section provides an analysis of CCP motion based on expert annotations. It investigates whether CCP movement is consistent with Brownian motion or if it displays more complex, directional patterns.
- **Selecting Tracking Evaluation Metrics:** Here, the focus is on the selection of appropriate metrics for evaluating the tracking algorithm. The section introduces novel metrics devised specifically for CCP tracking, in addition to discussing established metrics like MOTA and RMSE.
- **Algorithm Description:** This section describes the multiple hypothesis tracking algorithm adapted for CCP tracking. Detailed information on the algorithm's function, including the management of hypothesis trees and the step-by-step process for each frame, is provided.
- **Runtime Analysis of the Tracking Algorithm:** An analysis of the algorithm's runtime, particularly in relation to the pruning look-back parameter N, is presented. This section offers insight into the scalability of the algorithm.
- **Results:** The results section details the outcomes of applying the detection algorithm to TIRF-SIM image stacks in a cross-validation setup. It assesses the tracking algorithm's performance using the consolidated detection data and compares it with human annotators.

Each section builds upon the previous ones to create a comprehensive picture of the CCP tracking algorithm, from its theoretical foundations and design considerations to its practical application and performance evaluation.

5.2 Measuring Characteristics of CCP Motion

This section delves into the motion characteristics of CCPs as derived from annotated tracks provided by the three annotators. The key question addressed is whether CCP movement is better described by Brownian motion, characterized by its random nature, or by motion with inertia, suggesting the presence of directional trends or more sophisticated patterns.

Metrics such as displacement distribution and velocity autocorrelation are assessed from the annotated tracks to measure the randomness and potential persistence in the directional movement of CCPs between frames. These analyses are instrumental in determining the appropriate modeling approach for the CCP tracking algorithm.

The findings from these investigations are pivotal in guiding the development of the tracking algorithm, influencing whether a stochastic or predictive motion model is more appropriate. The subsequent paragraphs will detail the outcomes of these analyses and their implications for the tracking algorithm design.

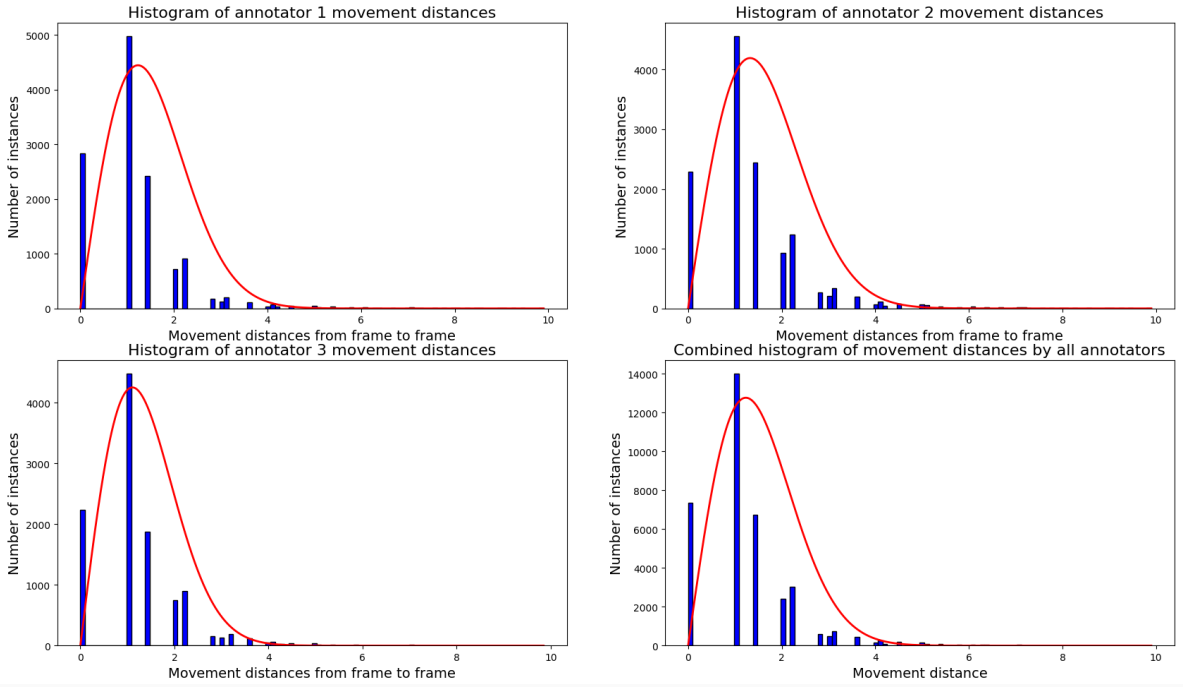


Figure 5.1: Histograms of CCP movement distances as annotated by three annotators, with the Rayleigh distribution overlaid. The observed distributions indicate a visual correspondence to the Rayleigh distribution, suggestive of Brownian motion.

The Rayleigh distribution, defined by the probability density function (PDF)

$$f(r;\sigma) = \frac{r}{\sigma^2} e^{-r^2/(2\sigma^2)},$$

for $r \geq 0$ and scale parameter σ, is a suitable model for the distance traveled by a particle undergoing Brownian motion in two dimensions per time step. The scale parameter σ is equivalent to the standard deviation of the underlying Gaussian distributions of the x and y components of the motion.

Figure 5.1 presents the histograms of the CCP movement distances for each annotator, with the corresponding Rayleigh distribution superimposed. These histograms reflect the frequencies of various distances the CCPs travel from frame to frame. A visual inspection of the overlaid Rayleigh distribution

suggests a reasonable fit to the empirical data, thereby supporting the hypothesis that CCP motion between frames can be modeled as a Brownian process. However, the precision of this fit is limited by the discretization of distances, as they are derived from the square roots of summed squares of integer pixel positions.

The histograms and the Rayleigh fits collectively serve as a preliminary indication that CCP motion is consistent with random walks, as characterized by Brownian motion. This foundational understanding informs the design of the tracking algorithm, which can incorporate stochastic elements reflective of this random movement behavior.

To further probe the nature of CCP motion, an analysis was conducted to determine the correlation of CCP movement between consecutive frames. This analysis is crucial to ascertain if the motion in the x and y dimensions exhibits temporal correlation, which would suggest inertia or directed movement, as opposed to a purely Brownian, random motion.

Figure 5.2 presents a series of plots, including scatter plots and log density plots, which visualize the correlation of movement distances in both x and y coordinates from one frame to the next. A lack of discernible pattern or trend in these plots would indicate the absence of a significant correlation, aligning with the characteristics of Brownian motion.

To quantify the degree of correlation, the Pearson correlation coefficient r was calculated, defined by the formula:

$$r = \frac{\sum_i (u_i - \bar{x})(v_i - \bar{y})}{\sqrt{\sum_i (u_i - \bar{u})^2} \sqrt{\sum_i (v_i - \bar{v})^2}}$$

where u_i and v_i are the individual sample values (corresponding to successive differences in position in the selected axis) indexed by i, and $\bar{u}$ and $\bar{v}$ are the mean values of the respective samples. The resulting coefficients were $r = -0.11$ for the x-position differences and $r = -0.10$ for the y-position differences. Given these values, the motion of CCPs can be described with high confidence as Brownian, devoid of any substantial autocorrelation in displacement from one frame to the subsequent one.

The implications of this finding are profound for the CCP tracking algorithm, as it suggests that the motion model need not account for inertia or directionality, but can be effectively based on the assumption of random, uncorrelated movement from frame to frame.

5.3 Selecting Tracking Evaluation Metrics

In evaluating the CCP tracking algorithm, it is crucial to utilize metrics that capture the unique aspects of tracking CCPs. While conventional MOT metrics serve as a standard in the literature, we propose an additional, novel approach tailored to the nuances of CCP tracking. Alongside the newly proposed metrics, we also employ established metrics MOTA and RMSE, as previously defined in section 2.5, to ensure a comprehensive evaluation.

Definition 7 (Temporal Intersection over Union (TIoU))**.** *The temporal intersection of two tracks is defined as the number of frames in which both tracks have detections within a Euclidean distance of at most 3 pixels from each other. Temporal union is the number of frames in which at least one of the two tracks has a detection. Then, the TIoU is calculated as:*

$$TIoU = \frac{Temporal\ Intersection}{Temporal\ Union}.$$

The TIoU metric ranges from 0 to 1, with 0 indicating no spatial or temporal overlap and 1 representing complete congruence of the tracks both spatially and temporally. The metric is illustrated in Figure 5.3.

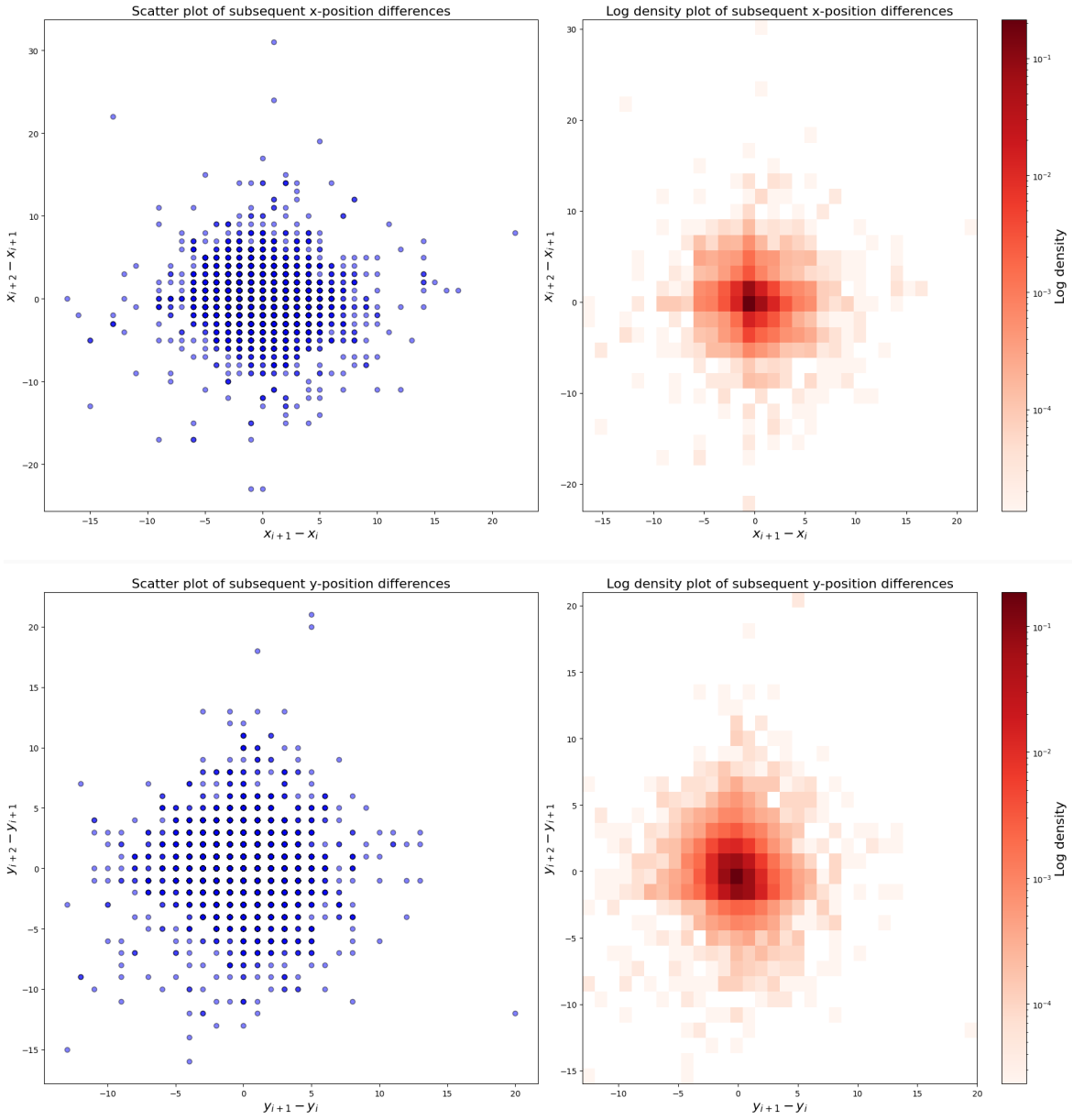


Figure 5.2: Top Left: Scatter plot of subsequent x-position differences. Top Right: Log density plot for x-position differences. Bottom Left: Scatter plot for subsequent y-position differences. Bottom Right: Log density plot for y-position differences. These plots illustrate the frame-to-frame motion correlation for CCPs.

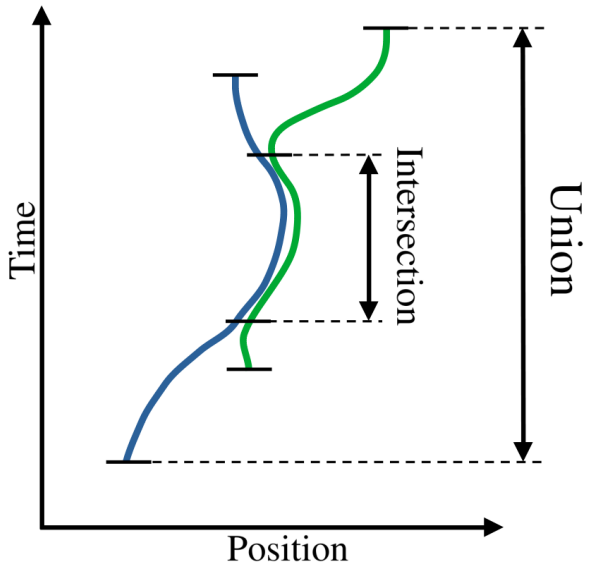


Figure 5.3: Illustration of temporal intersection over union (TIoU) between two tracks.

Definition 8 (Association Precision, Recall, and F-score)**.** *Given a set of tracks from the tracking algorithm and ground truth tracks from annotators, an association between a predicted track and a ground truth track is established if their TIoU is at least 0.5. True positives (TP) are the correctly associated tracks, false positives (FP) are the predicted tracks not associated with any ground truth, and false negatives (FN) are ground truth tracks not associated with any predictions. Based on these associations, the metrics are defined as follows:*

- *Association Precision (P): The ratio of TP associations to the sum of TPs and FPs.*

$$P = \frac{TP}{TP + FP}.$$

- *Association Recall (R): The ratio of TPs to the sum of TPs and FNs.*

$$R = \frac{TP}{TP + FN}.$$

- *Association F-score: The harmonic mean of precision and recall, providing a single measure of the test's accuracy.*

$$F = 2 \cdot \frac{P \times R}{P + R}.$$

Furthermore, the mean TIoU among associated pairs (TPs) is proposed as a measure, ranging from 0.5 to 1, to quantify the mean overlap of the tracking algorithm's outputs with the ground truth annotations.

The rationale for the adoption of these metrics is detailed in the subsection on the advantages of TIoU for CCP tracking, emphasizing their importance in enforcing stringent identity conservation in tracking, which is critical for the integrity of longitudinal analyses in CCP studies.

5.4 Algorithm Description

In this thesis, the tracking of CCPs is accomplished through a multiple hypothesis tracking algorithm. The specific adaptation of the MHT algorithm for CCP tracking follows the framework outlined in Section 2.4. This section details the particularities of the algorithm as applied to the CCP tracking problem.

The pruning look-back parameter is denoted N. The algorithm operates with two sets of hypothesis trees: the active trees, denoted as $\mathcal{A}$, and the finalized trees, denoted as $\mathcal{F}$. As the algorithm processes each frame k of the TIRF-SIM image stack, it performs the following steps:

Foundation of New Trees

For each detection at frame k, a new tree with its root at that detection is initiated and added to the set $\mathcal{A}$.

Expansion of Existing Trees

In our implementation, hypothesis trees consist of three types of nodes:

- Real detection nodes, representing the detection algorithm's responses.
- Missing nodes, signifying potential false negatives in the detection algorithm.
- Empty nodes, hypothesizing that the track terminated at an ancestor node.

While it is feasible to implement MHT without empty nodes, using them as we do—albeit more memory-intensive—offers programmatic convenience by keeping all hypotheses accessible at the current frame.

At frame k, the expansion of each leaf at frame $k-1$, belonging to a tree in $\mathcal{A}$, is contingent on its type:

- An empty node is extended with only another empty node as its child.
- A missing node is expanded exclusively with children corresponding to real detections at frame k that are within a 5-pixel radius.
- A real node is branched into all three types: an empty child, a missing child, and children tied to real detections at frame k within a 5-pixel radius.

Figure 5.4 depicts the growth of such a tree, simplified to show at most one real detection child per node.

Track Scoring

Each leaf in the active trees $\mathcal{A}$ at frame k represents a distinct track hypothesis. A score is assigned to each leaf to quantify the hypothesis's plausibility. Given the Brownian motion nature of CCP movement and the challenges in utilizing appearance features due to the similarity of CCPs and noise in the data, a more heuristic scoring function is adopted. Despite its simplicity, we have found scoring function to perform reasonably well in practice.

The scoring scheme is as follows:

- The score for a root node is set to 10.

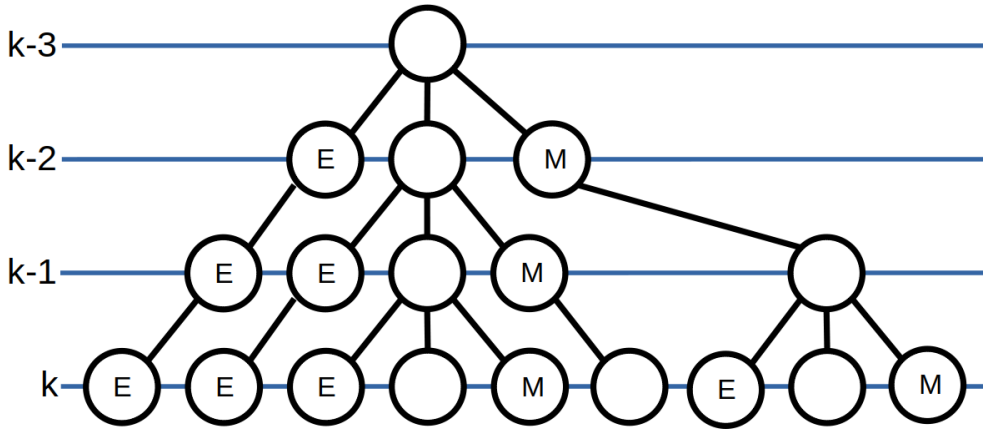


Figure 5.4: Illustration of the growth of a hypothesis tree in the MHT algorithm, showing the expansion of nodes with at most one real detection child for simplicity.

- For a node representing a real detection, the score is computed using the formula:

$$\text{Score}_{\text{real}} = 10 + \text{Score}_{\text{parent}} - \text{EuclideanDistance}(\text{node}, \text{parent}),$$

  where $\text{Score}_{\text{parent}}$ is the score of the parent node and $\text{EuclideanDistance}(\text{node}, \text{parent})$ is the Euclidean distance between the current node and its parent.

- For a node representing a missing detection, the score is:

$$\text{Score}_{\text{missing}} = 5 + \text{Score}_{\text{parent}} - \text{EuclideanDistance}(\text{node}, \text{parent}),$$

  with the same definitions as above.

- For a node representing an empty detection, the score is identical to its parent score:

$$\text{Score}_{\text{empty}} = \text{Score}_{\text{parent}}.$$

This scoring mechanism, while arbitrary, is tailored to the characteristics of CCP tracking, balancing the need for simplicity with the requirement for effective track hypothesis evaluation.

Collision Graph Construction

At this juncture, a collision graph G is constructed. Each vertex within G symbolizes a leaf at frame k from the set $\mathcal{A}$. The weight assigned to each vertex is determined by the score computed in the preceding step. An edge is established between any two vertices if, and only if, the tracks they represent are in conflict, meaning they share a real detection at the same frame at the same position.

The assembly of G is one of the more computationally intensive tasks due to its quadratic relationship with the number of leaves. To enhance efficiency, within each tree in $\mathcal{A}$, all vertices that correspond to its leaves are interconnected as they inherently conflict. Before iterating over all unique vertex pairs to identify potential collisions, the x and y positions associated with each vertex's detection are collated into arrays. This allows for a rapid preliminary distance check during the pair-wise iteration: only those pairs where both the x and y distances are less than or equal to $(2N \times 5\text{px} + 1\text{px})$ are considered further. Here, N denotes the number of frames considered in N-scan pruning, and this distance threshold ensures that only vertices representing detections that could feasibly collide—given the maximum distance a CCP could travel over N frames—are evaluated, streamlining the graph construction process.

Solving MWIS and Global Hypothesis Construction

Once the collision graph G has been established, the next step involves solving the Maximum Weighted Independent Set (MWIS) problem approximately on G. This is achieved by initially segmenting G into its constituent components through a standard breadth-first search algorithm. Within each component, the MWIS solver then identifies two candidate solutions for the approximate MWIS: one using a greedy algorithm (see Algorithm 2) and the other employing a modified greedy algorithm that incorporates penalties (see Algorithm 3). The candidate with the greater cumulative weight of selected nodes is adopted as the solution for that component. The final global hypothesis is the aggregate of solutions from all individual components.

The leaves at frame k corresponding to the vertices chosen in the MWIS delineate the global hypothesis. This global hypothesis embodies the most probable set of non-conflicting tracks up to frame k, considering the weighted scores and the constraints imposed by potential collisions as encoded in G.

Tree Pruning

The pruning of trees within the active set $\mathcal{A}$ involves a retrospective analysis spanning N frames. Specifically, for each tree with leaves at frame k that have been included in the global hypothesis, the algorithm prunes all divergent subtrees that originate from the node at frame $k-N$. This action effectively reinforces the chosen path of the track, consolidating its trajectory from frame $k-N$ to frame k.

Unselected Trees Deletion

Trees within $\mathcal{A}$ that do not contribute any leaves to the global hypothesis are deemed redundant and, as such, are excised from the active set. This step is pivotal in maintaining the computational efficiency of the algorithm by eliminating hypotheses that are no longer viable.

Transferring Finalized Trees to $\mathcal{F}$

The algorithm then sifts through $\mathcal{A}$ to identify and transition trees that have reached a conclusive state—those that are characterized by having an empty detection at frame $k-N$. Such trees are transferred from the active set $\mathcal{A}$ to the finalized set $\mathcal{F}$. This transfer is predicated on the fact that these trees, having only empty nodes beyond frame $k-N$, are no longer subject to branching and are incapable of further collisions. By moving them to $\mathcal{F}$, the algorithm ensures that the active set remains at a manageable size and that the complexity of constructing the collision graph G does not escalate indefinitely over time.

Finalization Steps for the Last Frame

Upon processing the final frame of the image stack, the algorithm executes a series of concluding operations to finalize the set of track hypotheses:

1. **Pruning of Selected Trees:** Trees that are part of the global hypothesis at the last frame undergo a comprehensive pruning process. This extends beyond the standard $k-N$ frame lookback, as all subsequent frames up to and including the last frame are also pruned to reinforce the finality of the chosen tracks.

2. **Transfer to Finalized Set:** Any trees that remain within the active set $\mathcal{A}$ are transferred to the finalized set $\mathcal{F}$. This transfer ensures that $\mathcal{F}$ encompasses a complete collection of non-branching trees, each corresponding to a distinct track hypothesis.

3. **Removal of Empty and Missing Detections:** For each tree within $\mathcal{F}$, sequences of empty detections are excised. Additionally, if the terminal node of a tree is indicative of a missing detection, it is likewise removed to refine the track representation.

4. **Deletion of Short Tracks:** Trees within $\mathcal{F}$ that consist of five nodes or fewer are deemed insufficiently substantiated and are consequently purged from the set.

With these steps completed, the set $\mathcal{F}$ contains the finalized tracks, which constitute the ultimate output of the algorithm. These tracks have been subjected to rigorous curation to ensure their validity and reliability as representations of the CCPs' trajectories.

5.5 Runtime Analysis of the Tracking Algorithm

The Multiple Hypothesis Tracking (MHT) algorithm, as implemented for this study, was developed in C++ without the use of parallelization techniques. An important aspect of assessing the practicality of any algorithm is its computational efficiency, particularly in terms of runtime. In the case of the MHT algorithm used for tracking CCPs, the runtime was closely monitored in relation to the pruning look-back parameter, denoted as N.

The parameter N plays a crucial role in the algorithm, influencing how far back in time the algorithm looks to prune and consolidate hypothesis trees. As N increases, the algorithm examines a broader range of historical data, which can significantly impact its computational demands. The following table presents the measured runtimes for different values of N, along with the multiplicative factor by which the runtime increases as N is incremented. This data provides valuable insights into the scalability of the algorithm and the trade-offs between increased look-back depth and computational load.

Table 5.1: Runtime of the MHT algorithm with respect to the pruning look-back parameter N, along with the multiplicative factor increase in runtime for each subsequent value of N.

N	Runtime	Multiplicative Factor
1	23s	
2	48s	2.09
3	2 min 21 s	2.94
4	7 min 29 s	3.18
5	26 min 13 s	3.50
6	1h 40 min	3.81
7	6h 49 min	4.09

The data presented in Table 5.1 highlight a significant trend in the runtime of the MHT algorithm as it relates to the pruning look-back parameter N. Notably, the increase in runtime with each incremental rise in N appears to be more than just linear or even exponential, indicating a steep computational cost for higher values of N.

This trend underscores the fact that while the MHT algorithm, in its current form, may not be feasible for real-time applications—especially at higher values of N—it remains a practical solution for post-processing tasks, particularly at lower values of N.

5.6 Results

The detection algorithm, as delineated before, was applied to the TIRF-SIM image stack employing a cross-validation regime to ensure comprehensive validation. The dataset comprising 120 frames was partitioned into six subsets, each containing 20 frames. This partitioning facilitated a six-fold cross-validation process, wherein each instance of the detector training deliberately omitted a different subset from the training phase, subsequently using it as the validation set. The outcomes from each iteration's validation set were then consolidated, providing a unified detection dataset that spanned the entire image stack.

While an independent set of annotated data would be ideal for validation purposes, the limited availability of such data necessitated the use of a cross-validation approach to evaluate the detector's performance comprehensively.

The consolidated detections, obtained from the cross-validated application of the detection algorithm, served as the input for the tracking algorithm. The pruning look-back parameter N was set to $N = 5$. The performance of the tracking algorithm was then assessed against the annotations provided by the annotators. The evaluation employed the metrics delineated in this chapter, specifically tailored to reflect the nuances of CCP tracking.

In the following series of tables, we present a comparative analysis of the performance between the three annotators and the tracking algorithm. Each table is dedicated to a different metric, illustrating how each annotator, as well as the tracking algorithm, performed against the annotations of every other annotator. These metrics include RMSE, MOTA, Association Precision, Recall, F-score, and Mean TIoU, encompassing various aspects of tracking accuracy and reliability. Notably, the last row in each table, although not conventionally relevant as the tracking algorithm is not the ground truth, is included for completeness and symmetry. This row depicts how the annotators' performance aligns with the tracking algorithm, offering an inverse perspective. The tables collectively provide an exhaustive quantification of the tracking algorithm's capabilities in relation to the human annotators, thereby facilitating a robust assessment of its efficacy in detecting and tracking CCPs.

Table 5.2: RMSE values between the annotators and the tracking algorithm. Rows represent the ground truth and columns represent the subject evaluated by the RMSE calculation.

	Annotator 1	Annotator 2	Annotator 3	Tracking Algorithm
Annotator 1	0.00	1.05	0.99	0.90
Annotator 2	1.05	0.00	1.07	0.94
Annotator 3	0.99	1.07	0.00	0.97
Tracking Algorithm	0.90	0.94	0.97	0.00

Table 5.3: MOTA values between the annotators and the tracking algorithm. Rows represent the ground truth and columns represent the subject evaluated by the MOTA calculation.

	Annotator 1	Annotator 2	Annotator 3	Tracking Algorithm
Annotator 1	1.00	0.82	0.72	0.58
Annotator 2	0.83	1.00	0.70	0.61
Annotator 3	0.83	0.85	1.00	0.62
Tracking Algorithm	0.57	0.64	0.55	1.00

Table 5.4: Association precision values between the annotators and the tracking algorithm. Rows represent the ground truth and columns represent the subject evaluated by the precision calculation.

	Annotator 1	Annotator 2	Annotator 3	Tracking Algorithm
Annotator 1	1.00	0.67	0.83	0.61
Annotator 2	0.68	1.00	0.71	0.58
Annotator 3	0.73	0.61	1.00	0.54
Tracking Algorithm	0.64	0.61	0.66	1.00

Table 5.5: Association recall values between the annotators and the tracking algorithm. Rows represent the ground truth and columns represent the subject evaluated by the recall calculation.

	Annotator 1	Annotator 2	Annotator 3	Tracking Algorithm
Annotator 1	1.00	0.68	0.73	0.64
Annotator 2	0.67	1.00	0.61	0.61
Annotator 3	0.83	0.71	1.00	0.66
Tracking Algorithm	0.61	0.58	0.54	1.00

Table 5.6: Association F-score values between the annotators and the tracking algorithm. Rows represent the ground truth and columns represent the subject evaluated by the F-score calculation.

	Annotator 1	Annotator 2	Annotator 3	Tracking Algorithm
Annotator 1	1.00	0.67	0.78	0.62
Annotator 2	0.67	1.00	0.65	0.59
Annotator 3	0.78	0.65	1.00	0.59
Tracking Algorithm	0.62	0.59	0.59	1.00

Table 5.7: Mean TIoU values between the annotators and the tracking algorithm. Rows represent the ground truth and columns represent the subject evaluated by the TIoU calculation.

	Annotator 1	Annotator 2	Annotator 3	Tracking Algorithm
Annotator 1	1.00	0.86	0.88	0.83
Annotator 2	0.86	1.00	0.86	0.86
Annotator 3	0.88	0.86	1.00	0.85
Tracking Algorithm	0.83	0.86	0.85	1.00

Following the detailed analysis provided in the preceding tables, we present an aggregated comparison of the performance metrics. This comparison is elucidated in Table 5.8, which consolidates the data into two primary columns for ease of interpretation. The first column, titled "Tracking Algorithm vs Annotators," represents the average metric values obtained when the tracking algorithm's performance is evaluated against each of the three annotators. This averaging process encompasses the algorithm's performance as gauged by all individual annotators, offering a comprehensive view of its effectiveness across different human assessments.

The second column, "Annotator vs Annotator," provides an average of each metric taken across all pairwise annotator comparisons, excluding self-comparisons. This column serves to illustrate the average level of agreement or consistency among the annotators themselves, thereby offering a benchmark for the variability inherent in human annotations. The comparison between these two columns sheds light on the relative performance of the tracking algorithm against a backdrop of human annotator consistency, highlighting areas where the algorithm excels or falls short in relation to human judgment.

Table 5.8: Comparison of average metrics between the tracking algorithm evaluated against annotators and annotator vs. annotator evaluations.

Metric	Tracking Algorithm vs Annotators	Annotator vs Annotator
RMSE	0.94	1.04
MOTA	0.60	0.79
Association Precision	0.58	0.71
Association Recall	0.64	0.71
Association F-score	0.60	0.71
Mean TIoU	0.84	0.87

The results presented in Table 5.8 reveal important insights about how the tracking algorithm compares with human annotators in tracking CCPs. The data show that the algorithm is not as accurate as humans when it comes to correctly associating tracks, as seen in metrics like MOTA, precision, recall, and F-score. This indicates that automated tracking still faces challenges, especially in tasks that require detailed understanding and interpretation, such as CCP tracking.

However, the tracking algorithm performs relatively well in other areas. Its performance in Mean Temporal Intersection over Union (TIoU) and Root Mean Square Error (RMSE) is close to that of the human annotators. A Mean TIoU of 0.84, compared to the human average of 0.87, suggests that the algorithm is quite good at aligning tracks in time and space. Even better, an RMSE of 0.94, compared to the human average of 1.04, shows that the algorithm is more precise in locating detections than human annotators.

Continuing with the assessment of the tracking algorithm's performance, the lengths of the tracks produced were analyzed to gain further insights into the algorithm's tracking capabilities over time. The distribution of these track lengths is visualized in a histogram, as shown in Figure 5.5. This representation allows us to observe that while the algorithm is capable of tracking a small number of tracks over extended periods, the majority of the tracks are relatively short, which may indicate a limitation in sustaining long-term tracking.

To put this into perspective, a comparison is made with the track lengths annotated by human annotators, depicted in Figure 5.6. A noticeable discrepancy between the two histograms highlights a potential issue with the tracking algorithm. The distribution of track lengths from the human annotators is expected to reflect a more accurate representation of CCP behavior. Consequently, the differences in the distributions suggest that the algorithm might either be prematurely terminating tracks or failing to consistently follow CCPs through their entire lifetime.

In summary, the comparison between the tracking algorithm's output (Figure 5.5) and the annotations by human experts (Figure 5.6) reveals key differences in track lengths, suggesting the algorithm might prematurely terminate tracks or inconsistently follow CCPs. While the algorithm demonstrates proficiency in aligning tracks and locating detections, it falls short of human annotators in association accuracy. These insights indicate areas for improvement, reinforcing the algorithm's role as an adjunct to human analysis and guiding further development for more effective CCP tracking.

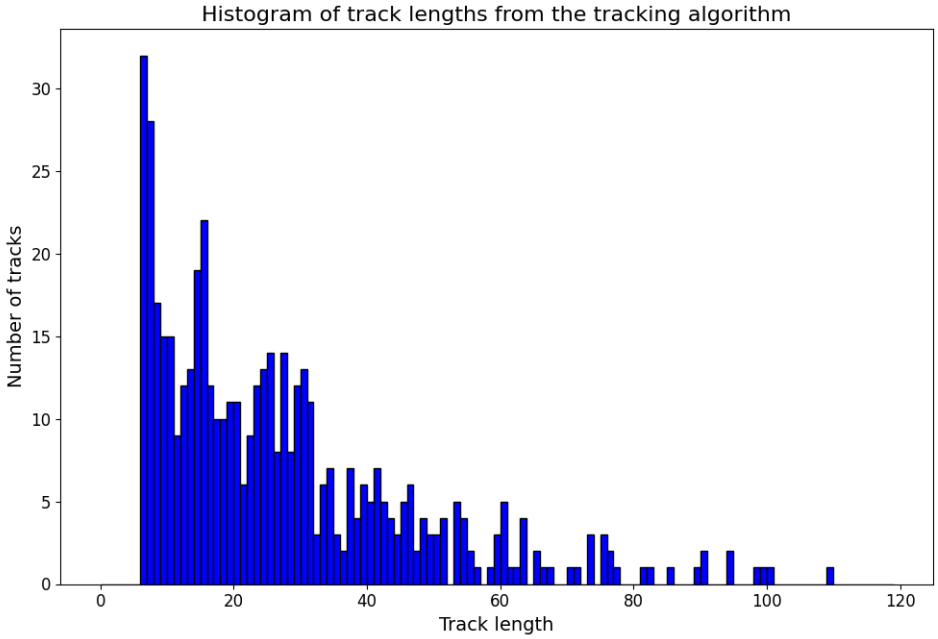


Figure 5.5: Histogram of track lengths from the tracking algorithm.

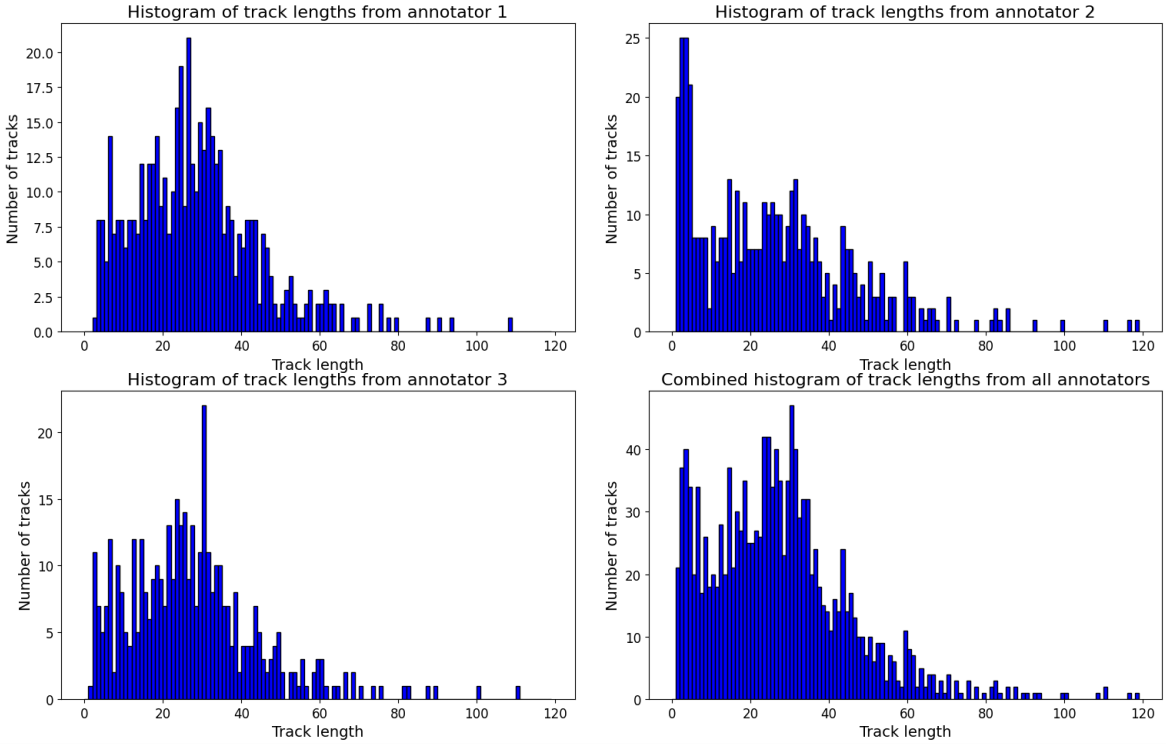


Figure 5.6: Histograms of track lengths annotated by the three human annotators. Bottom Right: histogram of aggregation of lengths of tracks from all annotators.

Conclusion

In conclusion, this thesis presents a comprehensive study on the process of endocytosis, focusing on clathrin-coated pits (CCPs) and employing advanced TIRF-SIM microscopy for detailed observation. Throughout the chapters, we have explored various facets of this biological process, from the fundamental mechanisms of endocytosis and the role of CCPs to the challenges and solutions in capturing their dynamics through high-resolution imaging techniques.

The development and evaluation of computational tools for detecting and tracking CCPs have been a pivotal aspect of this research. The implementation of a U-Net-based detection algorithm, tailored for per-pixel regression in TIRF-SIM images, has shown promising results in accurately identifying CCP locations. The subsequent adaptation of the Multiple Hypothesis Tracking (MHT) algorithm to this specific context marks a significant stride in tracking these dynamic structures.

The comparison focused on the algorithm's ability to accurately reconstruct the tracks of CCPs, taking into consideration the temporal and spatial precision of the detected positions. The metrics used in this evaluation, including the Temporal Intersection over Union (TIoU) and the adapted versions of Multiple Object Tracking Accuracy (MOTA) and Root Mean Square Error (RMSE), provided a multifaceted perspective on the tracking algorithm's efficacy.

Through rigorous evaluation against expert annotations, the algorithms have demonstrated their effectiveness, with the tracking algorithm aligning closely with CCP dynamics observed by human experts, especially in position precision metrics, although there is significant room for improvement in association accuracy measures. The novel evaluation metric introduced in this thesis has proven instrumental in assessing the tracking algorithm's performance, offering a quantifiable measure of it.

Despite these advancements, the research also highlights another area for improvement. The discrepancies observed in the tracking algorithm's output, particularly in terms of track length compared to human annotations, suggest opportunities for refining the algorithm to enhance its reliability.

The thesis has also underscored the broader implications of this research in the field of cellular biology. There is a growing interest in understanding how various factors influence CCP motion, density, and lifetimes, as exemplified in the analysis of track length histograms presented at the end of this thesis. Precise measurement and statistical analysis of these aspects are crucial for advancing our understanding of cellular endocytosis and its various biological impacts.

However, data annotation by human annotators, while accurate, is a labor-intensive process. In the realm of statistical biology, there is a pressing need for precise and automated methods to assess statistical data from microscopic time-lapse images of cellular endocytosis. This thesis makes a significant step towards developing automated software for CCP statistics measurement. By streamlining this process, it opens up new possibilities for more efficient and comprehensive studies in cellular biology, paving the way for deeper insights and discoveries in the field.

In essence, the methods and findings of this thesis contribute not only to the specific study of CCPs but also hold promise for broader applications in cellular biology, particularly in the automated and accurate analysis of complex biological data.

Bibliography

[1] A. Janich: *Klasifikace buněčných procesů na časosběrných snímcích pořízených TIRF-SIM mikroskopem*. Bachelor Thesis, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, 2022.

[2] G. M. Cooper: *The Cell: A Molecular Approach*. 2nd edition. Sunderland (MA): Sinauer Associates, 2000. *Endocytosis*.
https://www.ncbi.nlm.nih.gov/books/NBK9831/

[3] N. M. Willy et al.: *Endocytic Clathrin Coats Develop Curvature at Early Stages of Their Formation.* bioRxiv, 2019.
https://doi.org/10.1101/715219

[4] A. Fotin, Y. Cheng, P. Sliz, N. Grigorieff, S. C. Harrison, T. Kirchhausen, T. Walz: *Molecular model for a complete clathrin lattice from electron cryomicroscopy.* Nature, 432(7017): 573-579, 2004.
https://doi.org/10.1038/nature03079

[5] M. Mettlen, P.-H. Chen, S. Srinivasan, G. Danuser, S. L. Schmidt: *Regulation of Clathrin-Mediated Endocytosis.* Annual Review of Biochemistry, 87(1): 871-896, 2018.
https://doi.org/10.1146/annurev-biochem-062917-012644

[6] A. Paszke et al.: *PyTorch: An Imperative Style, High-Performance Deep Learning Library.* Advances in Neural Information Processing Systems, 32: 8024–8035, 2019.
https://doi.org/10.48550/arXiv.1912.01703

[7] PyTorch documentation.
https://pytorch.org/docs/stable/index.html

[8] M. G. L. Gustafsson: *Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy*. Journal of Microscopy, 198(2): 82-87, 2000.
https://doi.org/10.1046/j.1365-2818.2000.00710.x

[9] M. G. L. Gustafsson: *Nonlinear structured-illumination microscopy: Wide-field fluorescence imaging with theoretically unlimited resolution.* Proceedings of the National Academy of Science, 102(37): 13081-13086, 2005.
https://doi.org/10.1073/pnas.0406877102

[10] D. P. Kingma, J. Ba: *Adam: A Method for Stochastic Optimization*. 3rd International Conference for Learning Representations, San Diego, 2017.
https://doi.org/10.48550/arXiv.1412.6980

[11] M. Beck, M. Aschwanden, A. Stemmer. *Sub-100-nanometre resolution in total internal reflection fluorescence microscopy*. Journal of Microscopy, 232(1): 99-105, 2008.
https://doi.org/10.1111/j.1365-2818.2008.02075.x

[12] J. Liu, C. Liu, W. He: *Fluorophores and Their Applications as Molecular Probes in Living Cells*. Current Organic Chemistry, 17(6): 564-579, 2013.
https://doi.org/10.2174/1385272811317060003

[13] H. Li, J. C. Vaughan: *Switchable Fluorophores for Single-Molecule Localization Microscopy*. Chemical Reviews, 118(18): 9412-9454, 2018.
https://doi.org/10.1021/acs.chemrev.7b00767

[14] J. Marini. *Three major types of endocytosis*. Shutterstock, Stock Illustration ID: 1452304331.
http://www.shutterstock.com/image-illustration/three-major-types-endocytosis-phagocytosis-cell-1452304331

[15] M. Oheim, A. Salomon, A. Weissman, M. Brunstein, U. Becherer: *Calibrating Evanescent-Wave Penetration Depths for Biological TIRF Microscopy.* Biophysical Journal, 117(5): 795-809, 2019.
https://doi.org/10.1016/j.bpj.2019.07.048

[16] A. Cheng, R. Henderson, D. Mastronarde et al.: *MRC2014: Extensions to the MRC format header for electron cryo-microscopy and tomography*. Journal of Structural Biology, 192(2): 146-150, 2015.
https://doi.org/10.1016/j.jsb.2015.04.002

[17] R. Li, Q. Gao, K. Rohr: *Multi-Object Dynamic Memory Network For Cell Tracking in Time-Lapse Microscopy Images*. 18th IEEE International Symposium on Biomedical Imaging, Nice, France, 2021.
https://doi.org/10.1109/ISBI48211.2021.9433828

[18] M. Chen: *Cell tracking in time-lapse microscopy image sequences*. Computer Vision for Microscopy Image Analysis, Academic Press, 2021.
https://doi.org/10.1016/B978-0-12-814972-0.00005-9

[19] S.-L. Liu, Z.-G. Wang, H.-Y. Xie, A.-A. Liu, D. C. Lamb, and D.-W. Pang: *Single-virus tracking: from imaging methodologies to virological applications*. Chemical Reviews, 120(3): 1936-1979, 2020.
https://doi.org/10.1021/acs.chemrev.9b00692

[20] M. Liebel, J. Ortega Arroyo, V. S. Beltran, J. Osmond, A. Jo, H. Lee, R. Quidant, N. F. van Hulst: *3D tracking of extracellular vesicles by holographic fluorescence imaging*. Science Advances, 6(45), 2020.
https://doi.org/10.1126/sciadv.abc2508

[21] L. Zhang, R. Nevatia: *Global data association for multiobject tracking using network flows*. 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, USA, 2008.
https://doi.org/10.1109/CVPR.2008.4587584

[22] H. Pirsiavash, D. Ramanan, C. C. Fowlkes: *Globally optimal greedy algorithms for tracking a variable number of objects.* 2011 IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, USA, 2011.
https://doi.org/10.1109/CVPR.2011.5995604

[23] J. Berclaz, E. Turetken, F. Fleuret, P. Fua: *Multiple object tracking using K-shortest paths optimization*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(9): 1806-1819, 2011. https://doi.org/10.1109/TPAMI.2011.21

[24] A. A. Butt, R. T. Collins: *Multi-target tracking by Lagrangian relaxation to min-cost network flow*. 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, USA, 2013. https://doi.org/10.1109/CVPR.2013.241

[25] R. Ahuja, T. Magnanti, J.Orlin: *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[26] C. Kim, F. Li, A. Ciptadi, J. M. Rehg: *Multiple Hypothesis Tracking Revisited*. 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015. https://doi.org/10.1109/ICCV.2015.533

[27] S. S. Blackman et al.: *Multiple hypothesis tracking for multiple target tracking.* IEEE Aerospace and Electronic Systems Magazine, 19(1): 5-18, 2004. https://doi.org/10.1109/MAES.2004.1263228

[28] S. S. Blackman, R. Popoli: *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[29] J. B. D. Cabrera, L. I. Finn, S. Fairbrother: *Radar Resource Management for Multiple Hypothesis Tracking*. 2018 21st International Conference on Information Fusion (FUSION), Cambridge, UK, 2018. https://doi.org/10.23919/ICIF.2018.8455769

[30] R. Du, L. Liu, X. Bai, F. Zhou: *New Scatterer Trajectory Association Method for ISAR Image Sequence Utilizing Multiple Hypothesis Tracking Algorithm*. IEEE Transactions on Geoscience and Remote Sensing, 60: 1-13, 2022. https://doi.org/10.1109/TGRS.2021.3087192

[31] D. Reid: *An algorithm for tracking multiple targets.* IEEE Transactions on Automatic Control, 24(6): 843-854, 1979. https://doi.org/10.1109/TAC.1979.1102177

[32] K. Yoon, Y. M. Song, M. Jeon: *Multiple hypothesis tracking algorithm for multi-target multi-camera tracking with disjoint views*. IET Image Processing, 12(7): 1175-1184, 2018. https://doi.org/10.1049/iet-ipr.2017.1244

[33] H. A. Abdelali, H. Derrouz, Y. Zennayi, R. O. H. Thami, F. Bourzeix: *Multiple hypothesis detection and tracking using deep learning for video traffic surveillance.* IEEE Access, 9: 164282-164291, 2021. https://doi.org/10.1109/ACCESS.2021.3133529

[34] M. Mucientes and W. Burgard: *Multiple Hypothesis Tracking of Clusters of People*. 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 2006. https://doi.org/10.1109/IROS.2006.282614

[35] B. Lau, K. O. Arras, W. Burgard: *Tracking groups of people with a multi-model hypothesis tracker*. 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 2009. https://doi.org/10.1109/ROBOT.2009.5152731

[36] P. R. J. Ostergard: *A new algorithm for the maximum-weight clique problem*. Nordic Journal of Computing 8(4): 424–436, 2001.
https://dl.acm.org/doi/abs/10.5555/766502.766504

[37] H. Jiang, C.-M. Li, F. Manya: *An Exact Algorithm for the Maximum Weight Clique Problem in Large Graphs*. Proceedings of the AAAI Conference on Artificial Intelligence, 31(1), 2017.
https://doi.org/10.1609/aaai.v31i1.10648

[38] S. Busygin: *A new trust region technique for the maximum weight clique problem*. Discrete Applied Mathematics, 154(15): 2080-2096, 2006.
https://doi.org/10.1016/j.dam.2005.04.010

[39] I.M. Bomze, M. Budinich, P.M. Pardalos, M. Pelillo: *Handbook of Combinatorial Optimization (Supplement Volume A)*. Kluwer Academic, Dordrecht, 1999.

[40] M. Garey, D. Johnson: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman and Company, New York, 1979.

[41] L. Cehovin, A. Leonardis, M. Kristan: *Visual Object Tracking Performance Measures Revisited*, IEEE Transactions on Image Processing, 25(3): 1261-1274, 2016.
https://doi.org/10.1109/TIP.2016.2520370

[42] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, K. Schindler: *MOT16: A benchmark for multi-object tracking*. arXiv preprint, arXiv:1603.00831, 2016
https://doi.org/10.48550/arXiv.1603.00831

[43] F. Yin, D. Makris, S. A. Velastin: *Performance evaluation of object tracking algorithms.* 10th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Rio de Janeiro, Brazil, 2007.
https://www.researchgate.net/publication/228873288_Performance_evaluation_of_object_tracking_algorithms

[44] The American Type Culture Collection Website. *hTERT-RPE1 (ATCC®CRL-4000™)*. Retrieved January 2, 2024.
https://www.atcc.org/products/crl-4000

[45] R. D. Glickman: *Phototoxicity to the retina: mechanisms of damage.* International Journal of Toxicology, 21(6):473-90, 2002.
https://doi.org/10.1080/10915810290169909

[46] P. N. Youssef, N. Sheibani, D. M. Albert: *Retinal light toxicity* Eye, 25(1):1-14, 2011.
https://doi.org/10.1038/eye.2010.149

[47] K. M. Lee, K. H. Choi, M. M. Ouellette: *Use of exogenous hTERT to immortalize primary human cells*. Cytotechnology, 45(1-2):33-8, 2004.
https://doi.org/10.1007/10.1007/s10616-004-5123-3

[48] S. Yang, J. Zhou, D. Li: *Functions and diseases of the retinal pigment epithelium*. Frontiers in pharmacology, 12: 727870, 2021.
https://doi.org/10.3389/fphar.2021.727870

[49] D. Li, H. Colin-York, L. Barbieri, Y. Javanmardi, Y. Guo, K. Korobchevskaya, E. Moeendarbary, M. Fritzsche: *Astigmatic traction force microscopy (aTFM)*. Nature Communications, 12(1): 2168, 2021.
https://doi.org/10.1038/s41467-021-22376-w

[50] L. Barbieri, H. Colin-York, K. Korobchevskaya, D. Li, D. L. Wolfson, N. Karedla, F. Schneider, B. S. Ahluwalia, T. Seternes, R. A. Dalmo, M. L. Dustin, D. Li, M. Fritzsche: *Two-dimensional TIRF-SIM-traction force microscopy (2D TIRF-SIM-TFM)*. Nature Communications, 12(1): 2169, 2021.
https://doi.org/10.1038/s41467-021-22377-9

[51] M. G. L. Gustafsson, L. Shao, P. M. Carlton, C. J. R. Wang, I. N. Golubovskaya, W. Z. Cande, D. A. Agard, J. W. Sedat: *Three-dimensional resolution doubling in wide-field fluorescence microscopy by structured illumination*. Biophysical journal, 94(12): 4957-4970, 2008.
https://doi.org/10.1529/biophysj.107.120345