INAUGURAL – DISSERTATION

zur

Erlangung der Doktorwürde

der

Naturwissenschaftlich-Mathematischen Gesamtfakultät

der

Ruprecht – Karls – Universität

Heidelberg

vorgelegt von

Diplom-Mathematiker Jens Fangerau

geb. in Heidelberg

Tag der mündlichen Prüfung:

23. Januar, 2015

Interactive Similarity Analysis for 3D+t Cell Trajectory Data

Advisors:

Prof. Dr. Heike Leitte

Prof. Dr. Joachim Wittbrodt

Danksagung

An erster Stelle bedanke ich mich bei meiner Doktormutter Professor Heike Leitte für die konstruktiven Besprechungen, die Unterstützung und das Vertrauen in mich im gesamten Verlauf meiner Promotion sowie für die Freiheit, die sie mir gelassen hat um meine Ideen zu realisieren. Genauso bedanke ich mich bei meinem Mentor Alexis Maizel für unzählige hilfreiche Diskussionen, für die Überprüfung biologischer Aspekte in meiner Arbeit und für die Möglichkeit, in seiner Gruppe meine interdisziplinäre Forschung fortsetzen zu können. Weiterhin gilt mein Dank Professor Jochen Wittbrodt für die finanzielle Überbrückung während meiner Promotionszeit sowie Professor Peter Bastian für seine Betreuung besonders am Anfang meiner Promotion. Ich danke auch meiner Mentorin Susanne Krömker für ihre fachliche Unterstützung und das Korrekturlesen meiner Arbeit. Zusätzlich danke ich der Fakultät für Mathematik und Informatik für die Annahme als Doktorand.

Im Laufe meiner Promotion haben viele Diskussionen mit anderen WissenschaftlerInnen neue Denkanstöße geschaffen und mich in meiner Forschung vorangebracht. Ich danke insbesondere Burkhard Höckendorf für die Kooperation und seine stete Bereitschaft, mir biologische Zusammenhänge zu erläutern. Ich danke auch Daniel von Wangenheim und Alexander Schmitz für die Aufbereitung der biologischen Daten und für den fachlichen Austausch in unseren abwechslungsreichen Telefonkonferenzen. Ganz herzlich bedanke ich mich desweiteren bei Bastian Rieck für das Korrekturlesen meiner Arbeit und für all die vielfältigen und konstruktiven Diskussionen, die meine Forschung erheblich vorangebracht haben und aus denen neue Ideen entstanden sind. Ich bedanke mich auch bei Jurgis Pods für das Korrekturlesen sowie bei Elfriede Friedmann, Christian Heine, Bernhard Kausler und Martin Lindner für die Zusammenarbeit und die ergiebigen Besprechungen.

Ich bin der *HGS Math Comp* sehr zum Dank verpflichtet für das Stipendium sowie für die zahlreichen sozialen Aktivitäten, wie zum Beispiel das wöchentliche Fußball spielen, das viele neue Freundschaften hervorgebracht hat. Ich danke allen Mitgliedern der *CoVis* und *vNGG* Arbeitsgruppen für die schöne Zeit während und neben der Arbeit. Weiterhin danke ich den MitarbeiterInnen in Verwaltung und Sekretariat im fünften Stock des *IWR* für die immerzu schnelle Bearbeitung von Anfragen und Bitten.

In besonderem Maße danke ich meiner Familie, die mich jederzeit in vielerlei Hinsicht unterstützt hat. Als besonders wichtigem Menschen in meinem Leben danke ich auch Katarina Boland, die meine Arbeit Korrektur gelesen und mir privat immer Rückhalt gegeben hat.

Abstract

Recent data acquisition techniques permit an improved analysis of living organisms. These techniques produce 3D+t information of cell developments in unprecedentedly high resolution. Biologists have a strong desire to analyze these cell evolutions in order to find similarities in their migration and division behaviors. The exploration of such patterns helps them in understanding how cells and hence organisms are able to ensure a regular shape development. However, the enormous size of the time-dependent data with several tens of thousands of cells and the need to analyze it in 3D hinder an interactive analysis. Visualizing the data to identify and extract relevant features provides a solution to this problem. For this, new visualization approaches are required that reduce the complexity of the data to detect important features in the visual analysis.

In this thesis, novel visual similarity analysis methods are presented to interactively process very large 3D+t data of cell developments. Three main methods are developed that allow different visual analysis strategies. The usefulness of them is demonstrated by applications to cells from *zebrafish* embryos and *Arabidopsis thaliana* plants. Both data sets feature a high regularity in the shape formation of the organs and domain experts seek to research similar cell behaviors that are responsible for this development. For example, the identification of 3D division behaviors in plants is still an unresolved issue. The first method is a novel visualization approach that can automatically classify cell division types in plant data sets with high memory and time efficiency. The visualization is based on the generation of newly introduced *cell isosurfaces* that allow a quantitative and spatial comparison of cell division behaviors among individual plants. The method is applied to cells of the lateral root of Arabidopsis plants and reveals similar division schemes with respect to their temporal order. The second method enables a new visual similarity analysis for arbitrary 3D trajectory data in order to extract similar movement behaviors. The algorithm performs a grouping of thousands of trajectories with an optional level of detail modification. The clustering is based on a newly weighted combination of geometry and migratory features for which the weights are used to emphasize feature combinations. As a result, similar collective cell movements in zebrafish as well as a hitherto unknown correlation between division types and subsequent nuclei migrations in the Arabidopsis plants are detected. The third method is a novel visualization technique called the *structure map*. It permits a compact and interactive similarity analysis of thousands of binary tree structures. Unique trees are pre-ordered in the map based on spectral similarities and substructures are highlighted according to user-selected *tree descriptors*. Applied to cell developments from zebrafish depicted as trees, the map achieves compression rates up to 95% according to spectral analysis and facilitates an immediate identification of biologically implausible events and outliers. Additionally, similar quantities of feature appearances are detected in the center of the lateral root of several Arabidopsis plants.

Zusammenfassung

Moderne Datenaufnahmetechniken erlauben eine verbesserte Analyse von lebenden Organismen. Diese Techniken liefern 3D+t Informationen von Zellentwicklungen in bisher unerreicht hoher Auflösung. Es ist Biologen ein großes Anliegen, diese Zellevolutionen zu analysieren, um Ähnlichkeiten in deren Bewegungs- und Teilungsverhalten zu finden. Die Untersuchung von solchen Mustern hilft ihnen zu verstehen, wie Zellen und demzufolge Organismen fähig sind, eine Regelmäßigkeit in ihrer Strukturbildung zu gewährleisten. Allerdings erschweren die enorme Größe der zeitabhängigen Daten von mehreren zehntausenden Zellen und die Notwendigkeit, eine Analyse in 3D durchzuführen, eine interaktive Analyse. Die Visualisierung von Daten, um relevante Eigenschaften erkennen und extrahieren zu können, ist eine Lösung für dieses Problem. Dafür werden neue Visualisierungsansätze gebraucht, die die Datenkomplexität verringern, um wichtige Merkmale in der visuellen Analyse erfassen zu können.

In dieser Doktorarbeit werden neuartige visuelle Ähnlichkeitsmethoden vorgestellt, um sehr große 3D+t Zellentwicklungsdaten interaktiv verarbeiten zu können. Hierzu werden drei Methoden vorgestellt, die verschiedene visuelle Untersuchungsarten unterstützen. Der Nutzen dieser Methoden wird anhand von Analysen auf Zelldaten von *Zebrafisch-Embryos* und *Arabidopsis thaliana* Pflanzen demonstriert. Beide Datensätze weisen eine hohe Regularität bei der Organbildung auf und Fachexperten streben nach der Erforschung von ähnlichem Zellverhalten, das für diese Entwicklung verantwortlich ist. Zum Beispiel ist die Identifikation von dreidimensionalem Teilungsverhalten in Pflanzen noch immer ein ungelöstes Problem. Die erste Methode ist ein neuartiger speicher- und zeiteffizienter Visualisierungsansatz, der eine automatische Klassifikation von Zellteilungstypen in Pflanzendaten ermöglicht. Hierfür werden sogenannte *Zellisoflächen* eingeführt, die einen quantitativen und räumlichen Vergleich von Zellteilungsverhalten in Pflanzen zulassen. Diese Methode wird auf Zellen der Arabidopsis Pflanze bei der Entwicklung der Seitenwurzel angewandt und führt zu Erkenntnissen über Regelmäßigkeiten bezüglich der zeitlichen Abfolge von Zellteilungen. Die zweite Methode ermöglicht eine neue Ähnlichkeitsanalyse von beliebigen 3D Trajektoriendaten, um ähnliche Bewegungsverhalten zu identifizieren. Der Algorithmus führt eine Gruppierung von tausenden von Trajektorien durch, die optional in ihrem Detailgrad angepasst werden können. Für das Clustering wird eine Kombination aus geometrischen und bewegungsbasierten Features verwendet, das eine individuelle Gewichtung von Featurekombinationen erlaubt. So lassen sich ähnliche kollektive Zellbewegungen im Zebrafisch sowie eine bisher unbekannte Korrelation zwischen Zellteilungstypen und anschließender Zellkernbewegung in Arabidopsis Pflanzen erkennen. Die dritte Methode ist eine neuartige Visualisierung, die als *Strukturkarte* bezeichnet wird. Diese ermöglicht eine kompakte und interaktive Ähnlichkeitsanalyse tausender Binärbaumstrukturen. Bäume werden hinsichtlich spektraler Ähnlichkeiten bezüglich ihrer Form vorsortiert und Substrukturen können vom Nutzer durch die Auswahl bestimmter *Baumdeskriptoren* ermittelt und hervorgehoben werden. Angewandt auf als Bäume dargestellte Zellentwicklungen vom Zebrafisch erzielt die Strukturkartenmethode Kompressionsraten von bis zu 95% bezüglich der Spektralanalyse und sie unterstützt das mühelose Auffinden von biologisch unglaubwürdigen Ereignissen und Ausreißern. Zusätzlich werden durch Anwendung der Methode ähnliche Quantitäten an Featurevorkommen im Zentrum der Seitenwurzel mehrerer Arabidopsis Pflanzen entdeckt.

Contents

Chapter 1

Introduction

"Nature knows no pause in progress and development,
and attaches her curse on all inaction."
— Johann Wolfgang von Goethe, 1749–1832

1.1 Motivation

The movement of objects is an event that can be observed almost everywhere in the world. Movement of an entity is defined as the physical change of its position in relation to a reference system which is most commonly the geographical 2D or 3D space. Movement behavior is analyzed in many research areas such as robotics, Geographic Information Science, environmental meteorology, robotics, transportation engineering, environmental meteorology, or molecular and developmental biology. More and more interest has arisen in the analysis of these movements to gain knowledge about collective behavior, patterns and similar properties. Most of this data is generated, processed and analyzed in 2D but only little research has been done in the investigation of 3D movement data. In developmental biology, domain experts aim to explore plenty of 3D+t cell developments to exploit information of similar cell behaviors [HTW12]. This cell data used to be only accessible in 2D via microscopic images which may lead to wrong interpretations or missing facts about growth, cell migrations and cell divisions. However, a valid examination of certain division behaviors can only be achieved in 3D when orientation and position properties of cells are considered. Similarly, domain experts require an analysis in the same 3D space from which the cell data originates. This analysis allows an adequate exploration of cell migrations to formulate hypotheses. Hence, a visual analysis in 3D is required that features a comprehensive representation, e.g. visualization to process very large 3D+t data sets. However, the interactive visualization of tens of thousands of cells is a challenging task. While there are many approaches that are designed to deal with 2D movements, they are mostly not suited for the analysis of 3D data. Furthermore, a three-dimensional visualization of all cell entities may complicate the interpretation of the visual results. For these reasons, new interactive visualization approaches are required that reduce the complexity of huge data sets such that relevant information is readily identifiable.

In this thesis, I present new interactive visualization methods for exploring plenty of 3D+t cell developments. These methods permit the analysis and detection of similar cell divisions

and cell migrations using different visual data representations. I demonstrate the usefulness of my methods applying them to model organisms of the *zebrafish* embryo and the plant model *Arabidopsis thaliana*. In the following, I describe the process of how visualization can help in the analysis of this data followed by background knowledge of the used biological data. After this, I describe the contributions of my work.

1.1.1 The Role of Visualization

Visualization offers one way to analyze large data sets. It is the presentation of data in such a way that it enables an efficient analysis and interpretation of information concealed in the data. For this purpose, data entities are transformed into graphical features such as geometrical objects, charts, maps or diagrams. Shneidermann [Shn96, p. 2] describes the data exploration process with the commonly known information seeking mantra: "Overview first, zoom and filter, then details on demand". However, when working with extremely huge and complex data sets, adhering to interactivity and overview is a challenging task. In the research area of *visual analytics* introduced by Wong and Thomas [WT04], data mining techniques are used to cope with this very large data. A definition is given by Keim et al. [KKEM10, p. 7]: "Visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets." Thus, automatic tools and techniques are established in order to support people in gaining insights from massive and complex data sets in an interactive manner. In the area of visual analytics, Keim et al. [KMS+08, p. 7] extend the above data processing mantra by the application of automatic analysis methods: "Analyze first - show the important - zoom, filter and analyze further - details on demand". This indicates that a visualization only used as a visual

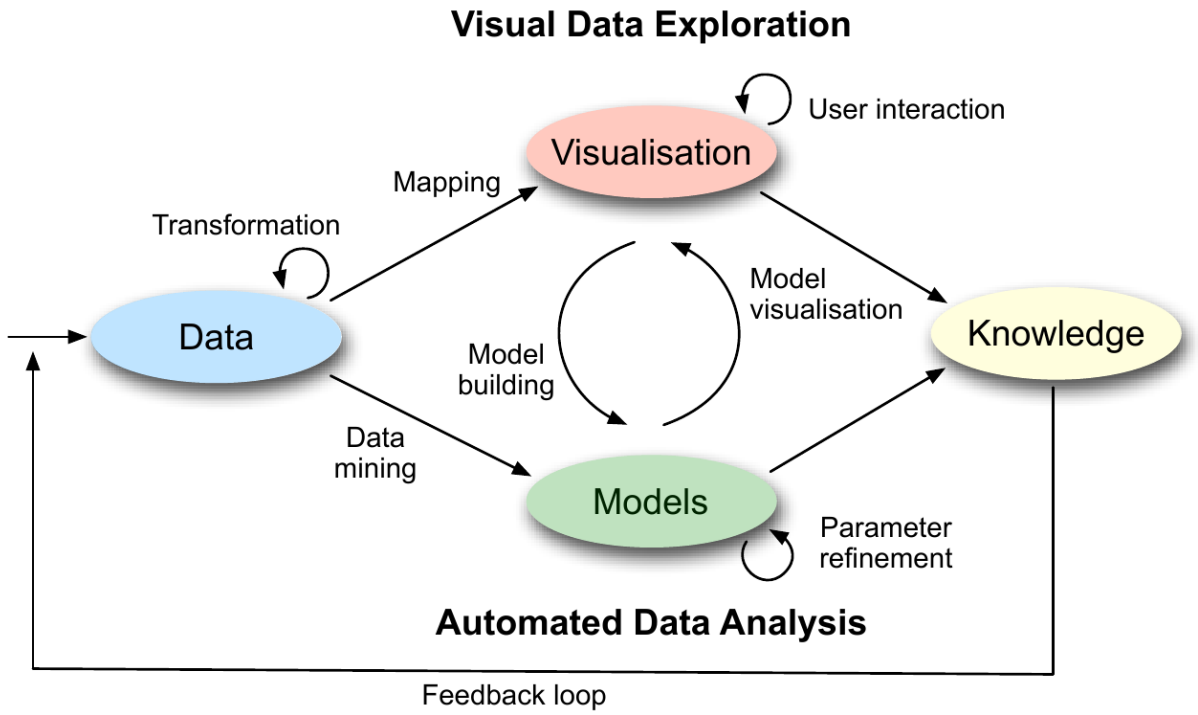


Figure 1.1: Visual analysis process (Figure taken from Keim et al. [KKEM10, p. 10]).

metaphor is not enough to handle large data sets. Instead, a focused analysis of data of interest is required providing further possibilities for interactions on demand. Figure 1.1 illustrates an abstract overview of the visual analysis process. In most cases, the raw data (blue) has to be preprocessed before it is suited for a visual analysis. Among other transformations like data cleaning or normalization this also includes extracting portions of the data selected by the user to focus the analysis on relevant information. This type of preprocessing is also required for the 3D+t data of the model organisms in this thesis. The required steps (segmentation, tracking, feature extraction) are explained in detail in chapter 3. The analysis continues either directly with an interactive visualization (red) of the data or with performing an automatic data mining approach resulting in models (green). In case the analysis starts with a visualization, automatic methods are used to confirm hypotheses by building models combined with user interaction. If an automated approach is chosen first, parameters of the generated model can be steered using a visualization of the model. In summary, the visual analysis process is a combination of visual data exploration, automatic data analysis and the repeated interaction between visualization and model that results in gained knowledge (yellow). I apply this successive process to analyze large 3D+t cell development data and to detect similarities of cell behaviors.

1.1.2 Biological Data

Recent advances in 3D+t data acquisition methods in developmental biology allow a fundamentally new access to the analysis of cell developments in embryos. The use of high-resolution light-sheet fluorescence-based live imaging [KSWS08] enables biologists to follow single cells during the *embryogenesis*, i.e. the process by which the embryo forms and develops. The analysis of cell developments facilitates the generation of complete *cell lineages* [KSWS08, OLOD$^+$10]. A *cell lineage* [Chi01] is the visualization of a single cell evolution in a binary tree structure. This lineage tree represents all biological events of such a cell and is referred to as a *cell fate map*. Fate maps allow the investigation and observation of individual cell developments from an early embryo to various tissues in particular regions of the adult organism. The choice of a cell's fate and consequently its identity influences all properties of its behavior such as morphology, migration, and proliferation. For example, liver cells are specialized in detoxification, muscle cells in contraction, neurons in electrical activity, and white blood cells in immunity [Fur10]. However, the cell lineages depend strongly on the quality of the *segmentation* and *tracking* process in which individual cells are detected in the raw data and tracked over time. The set of cell lineages *(lineage diagram)* can often be defective featuring biologically implausible events like immediate subsequent cell divisions. A similarity analysis of these cell lineages can be used for two purposes: First, the detection of patterns yields important insights of cell developments sharing similar fates. This supports domain experts in understanding how cells are organized and structured. Second, a similarity measure helps identify erroneous substructures in tracked cell developments. Finding patterns can either be used to exclude them from further analysis or to explore their origins in the imaging process.

Common model organisms for research are the *zebrafish (Danio rerio)*, frog, mouse, chick, fruit fly, and worm in animal biology or the *Arabidopsis thaliana* in plant biology. The zebrafish has several advantages for cellular studies of vertebrate embryonic development. It features a short generation time of few months and produces up to 200 embryos per mating. The embryos are transparent throughout the early development which permits monitoring and analysis of all morphological events. In addition, the zebrafish can be used as models of a wide variety

of human diseases [LC07]. The transparency also allows microscopic data acquisitions and in combination with injection of tracer dyes the generation of cell lineages and 3D+t cell development data [Kel13, KSWS08]. In the zebrafish, cell migration and acquisition of a specific cellular identity (fate) are interlinked. Fate decisions can result in a certain cell migration while the fate itself can be influenced by the position of the cell within the developing embryo. To understand the logic articulating fate acquisition and cell migration, an analysis of cell migration patterns is required. The plant model Arabidopsis thaliana has a small size and a rapid life cycle and can produce several thousands of translucent seeds making them well-suited for cell imaging with fluorescence microscopy. Unlike animals, plants constantly form new organs throughout their lives and the shapes of these organs follow a high regularity. This robustness contrasts with the unparalleled ability of plants to adapt their growth to a highly variable environment, a process called *plasticity*. For biologists, there is a need to understand how the plant is able to cope with both plasticity and robustness. These phenomena can be investigated on a cellular scale by analyzing similar cell migrations as well as cell division patterns to understand how the plant is able to manage its regular shape development. In contrast to animals, there is no movement of plant cells but only intracellular nuclei displacements over time. The plant cells are firmly attached to each other and the shape of the plant results only from cell divisions and cell growth. Recently, 3D+t cell data of *lateral roots* of the Arabidopsis plant has been acquired [MvWF$^+$11, vWDL$^+$14] that is used in this thesis to demonstrate the usefulness of my methods.

1.2 Contributions

The contribution of this thesis is the development of novel visualization methods that permit a similarity analysis to interactively process very large amounts of 3D+t cell developments. The usefulness of the methods is demonstrated by applications to cells from zebrafish and Arabidopsis data sets. I perform the analysis on two different visual representations to illustrate migration information: *3D Cell trajectories* and *2D cell lineages*. A *trajectory (geospatial lifeline)* describes the path of a moving entity in space with respect to time. More precisely, it is defined by a mapping from a set of time steps to positions in space. Structural properties are analyzed in 2D cell lineages which were explained above. Both visualization types originate from the same data but they provide different interpretation possibilities to investigate cell migrations and cell divisions in a completely new way. The visualization methods enable the detection of expected behavior as well as the identification of unexpected similarities and correlations. The following list gives a detailed overview of the three methods and their contributions:

- Primarily designed for plant data sets, I developed a novel automatic classification algorithm (see chapter 4) that can distinguish between three cell division types (*anticlinal, periclinal, radial*) occurring during the growth of lateral roots. The classification is realized in less than one second with a space usage smaller than one MiB. It is based on the generation of newly introduced 3D triangulations called *cell isosurfaces* that are formed by nuclei positions. These colored surfaces represent the developing shape of the lateral root in each time step. The surface color refers to its isovalue which is the frequency of periclinal divisions for all associated cells. The triangulation is realized using *alpha shapes* to approximate the tissue growth of the lateral root. The algorithm determines the division type of a cell by comparing the angles between the division orientation and the vertex normal of the associated cell's isosurface. The angles are then compared to user-selected thresholds to define

the type of the division. The visualization of the isosurfaces allows a geometric and spatial comparison of periclinal divisions among several plant data sets. These divisions are of high biological interest because they are mainly responsible for the lateral root growth in height. The results are analyzed in 3D with a color-coded lineage diagram and additional information of division sequences. As a result for a set of Arabidopsis plants, similar division schemes with respect to their temporal order are identified.

- I created a novel visual similarity analysis method for 3D+t cell trajectories [FHWL12] described in chapter 5. The similarity measure is based on a weighted combination of geometrical and migratory features. These weight parameters are checked for cluster stability and permit a new analysis approach to emphasize specific combinations of features. For each pair of trajectories, the geodesic distances between spherical coordinates of migration directions as well as differences of cell cycle lengths and velocities are computed. Additionally, the *coupling distance* [EM94b] is determined in order to compare the shapes of a pair of trajectories. All these features are used to capture similar migration behaviors and have not been used before in this combination. The results are stored in a similarity matrix that is processed in a hierarchical clustering approach with computation times of a few seconds and a space usage of at most 30 MiBs for the investigated data sets. The analysis of trajectory data is improved by omitting outliers and by applying a level of detail approach. Color-coded trajectories are displayed in 3D to inform the user about clusters of trajectories with coherent cell motions. More information is given in a *dendrogram* in which the cluster hierarchies are presented. An additional lineage diagram represents the colored cluster memberships of individual cell migrations. The method is applied to both model organisms but it can be used to analyze any kind of trajectory data. For the zebrafish, similar features of cell trajectories are detected. For the Arabidopsis data, the visualization allows the identification of a hitherto unknown correlation between the cell division orientations (generated with the first analysis method) and subsequent nuclei displacements.

- I developed a new visualization method called the *structure map* [FHR$^+$15] explained in chapter 6. It enables an interactive overview and comparison of similar structures and features in thousands of trees. This map is a matrix-based 2D visualization in which unique trees in squared tiles are arranged in a compact and uniform design. The trees are ordered in a few seconds using *principal component analysis* based on the similarities of tree spectra, i.e. the set of eigenvalues. The ordered trees are then aligned using a space-filling *Hilbert curve*. The structure map features both a global analysis based on user-selected *tree descriptors* and a local investigation of these descriptors in each single tree structure. The similarities of trees are indicated by color-coded tiles while individual substructures within the trees are highlighted in another color. The map can be used to visualize any kind of tree data or even graphs but I focus on cell lineages from the zebrafish and Arabidopsis data sets. Descriptors can be set arbitrarily. To answer specific questions in this area, I defined them in a way suitable to analyze biological events. The descriptors can be steered by the user with an immediate visual feedback of the colored map and the highlighted structures. For the zebrafish data sets, the map features compression rates between 82% and 95% according to spectral analysis. As a result, substructures and outliers that are biologically implausible can be identified immediately (see also video [FHR$^+$14]). For the Arabidopsis plants, plenty of similar features are observed in a specific region within the lateral root called the *master cell file* introduced later.

All presented methods are integrated into the visualization software *Scifer* (http://www.scifer.info) that is developed in the Computer graphics and Visualization group (CoVis) of Prof. Dr. Heike Leitte in Heidelberg University. This software is designed for the interactive analysis of scientific data and to cope with large 3D+t data [LFL+12]. It is written in C++, using OpenSceneGraph (http://www.openscenegraph.org/) as a graphics toolkit and Qt (http://qt-project.org/) for the user interface.

1.3 Overview of Thesis

The thesis continues with related work and a description of all preprocessing steps. Afterwards, all three visual analysis methods are described in detail followed by a concluding discussion. The following enumeration gives more information about each chapter.

In **chapter 2**, I provide an overview of existing work related to this research. Because two different visualization types are used to display cell developments (cell trajectories and cell lineages), I consider data given as trajectories and tree data. The chapter starts with an overview of existing visual analysis methods of trajectory data in general. Afterwards, I focus on state-of-the-art visualization methods for cell trajectories and visual analysis techniques for investigating tree data from biology.

In **chapter 3**, I explain the different required preprocessing steps from the living model organisms to the ready-made data sets suitable for the visual analysis methods. These steps consist of the data acquisition process, the segmentation and tracking of data. Based on that, my contribution starts with the computation of specific features relevant for the analysis and the definition of migration terms used in this thesis.

Chapter 4 focuses on the first of three visual analysis methods. Mainly designed for plant data sets, I introduce a novel algorithm to automatically classify the different division types of the plant cells. This method is applied to several Arabidopsis data sets and the results of the division schemes are compared with each other. The division types are further used as an additional feature for the other two visual analysis methods to gain more knowledge about the data.

In the following **chapter 5**, I introduce the second method that provides a similarity analysis of 3D cell trajectories. Prior to the visual analysis, a method for modifying the level of detail can be applied to the trajectories in order to reduce data complexity and to focus on certain migration properties. The similarities based on combined migratory and geometrical features are computed and used in a hierarchical clustering approach. The performance as well as the cluster validity of the algorithm are examined before its usefulness is verified by applications to the zebrafish and Arabidopsis data sets.

Chapter 6 presents the third visual analysis method that focuses on the similarity analysis of thousands of tree structures. I introduce the new visualization method called the structure map and the underlying methods: Spectral analysis, principal component analysis, and alignment of trees using a Hilbert curve. The identification of features in trees is based on user-selected tree descriptors. I explain the functionality of the map, discuss its performance analysis, and present its application to the zebrafish and Arabidopsis data sets.

In the last **chapter 7**, I compare the presented methods to existing related techniques and discuss why I made the corresponding design decisions. These discussions yield ideas for future work and possible enhancements for the methods.

Chapter 2

Related Work

"The knowledge of all things is possible."
— Leonardo da Vinci, 1452–1519

The visualization and analysis of movement behavior can yield new insights and improvements in many areas. Examples are new optimization strategies of transportation routes, the analysis of atmospheric phenomena, or a better understanding of biological events in model organisms and eventually in human beings. Recently, more and more effort has been spent on the visual analysis of characteristics and similarities of moving objects. Especially in developmental biology, depending on the studied organism, the systematic similarity analysis of 3D+t cell behavior in an entire embryo requires the visualization of thousands of cell positions. The visualization of such huge time-dependent data is a challenge for each visualization method. However, the interactive access to this kind of data is a fundamental requirement for users to investigate the underlying structure of movement data.

2.1 Visual Analysis of Trajectory Data

The movement information is commonly visualized by two approaches: It is shown as a *trajectory* rendered by a simple polygonal line in which the time-dependent positions are plotted in 2D or 3D. The additional time parameter is either ignored or given as a feature for each individual position. This type of visualization is predominantly chosen for the visualization of 2D traffic or transport data on road maps, for example. Although the pure visualization of trajectories gives an intuitive representation of the movement data, it often suffers from *overplotting* which is the plotting of data on top of a previous plot. The second approach that is applied in the area of developmental biology is the visualization of a cell development in a binary tree structure called a *cell lineage* [Chi01]. Read from top to bottom, the cell development with migrations or divisions is illustrated by nodes with one or two siblings, respectively. While this tree structure allows a direct comparison of cell developments over time, the corresponding position information of cells is missing. Both visualization approaches (trajectory and cell lineage) can be analyzed with different types of similarity measures. In this thesis, I use both visualizations in order to analyze similar cell migration behavior.

The similarity analysis of trajectories can be classified into two categories: *Complete* and *partial similarity* [DLM+98]. The first one denotes the matching of complete trajectories of same length considering each one as a single unit. The second term refers to a pair of trajectories with different lengths that are compared by finding the best match of a subsequence of one trajectory in the other one. The cell trajectories in this thesis are defined as the cell migration range between subsequent cell divisions (Section 3.5) and therefore, I focus on the similarity analysis of complete trajectories. Through this, biologists can compare complete cell cycle lengths instead of partial cell migrations. A *similarity measure* can be defined by a *distance function* that quantifies the similarity between two trajectories. In literature, there are a lot of different distance functions determining the spatial similarity between trajectories. Overviews are given by Dodge [Dod11, p. 21–36] and Wang et al. [WSZ+13]. In this thesis, the values of such a function are mapped to $[0, 1]$ with zero defining a perfect match and one representing highest dissimilarity. Several features can be taken into account for the computation of similarity. Dodge et al. [DWL08] present a conceptual framework for the properties and classification of different moving objects and movement patterns. For the movement parameters, they distinguish between *primitive parameters* such as the position, *primary derivatives* like direction, velocity, and length/duration of a movement, and *secondary derivatives*, e.g. sinuosity or acceleration. After discussion with domain experts, I focus on primitive parameters and primary derivatives such as shapes, durations, dynamics of speed, and directions of movements to quantify the similarity between cell developments.

There are a lot of visualization methods that focus on the investigation of 2D traffic and transport trajectories drawn on top of a geographical map. Andrienko et al. [AA13] give an extensive overview of different approaches and tools used for the visual analysis of traffic movement data. Common choices are *static* and *animated maps* [AAG00]. The animated map features a focused visualization based on an user-selected time filter. These maps can be extended by a *space-time cube (STC)* [Kra03] (left image in Figure 2.1). Here, the third dimension on top of the map is used for the time parameter, permitting an analysis of trajectories for different time intervals. However, using STC does not scale well to thousands of cell trajectories from biology. Tominski et al. [TSAA12] use animated maps and include them in a hybrid 2D/3D display. On top of a 2D geographical map, trajectories are visualized as stacked bands colored based on their attribute values (right image in Figure 2.1). They combine this with a time graph showing the dynamics of the trajectories and their properties at different time ranges. In addition to the limitation to 2D trajectories the coloring allows only a limited view of movement characteristics which is not suitable to investigate cell developments.

Spretke et al. [SBJ+11] introduce a visual interface called *Animal Ecology Explorer* for the analysis of 2D animal trajectories on top of a static map. Multiple trajectories are rendered with different colors illustrating various types and properties of movements. They use *brushing and linking* which is the connection of several views of the same data in such a way that an interactive change in one view affects the representations in all other views. Through this, they can cross-compare animal movements with additional attribute information given in line charts. However, these visualizations often suffer from visual occlusion and overplotting problems. In order to reduce these negative effects they use *k-means* [For65] clustering applied to movement features to draw simplified 2D trajectories with less overplotting in a second map visualization. Several features such as speed, distance, and duration are computed for which the user can define range parameters to split trajectories for detailed events of interest. These features and

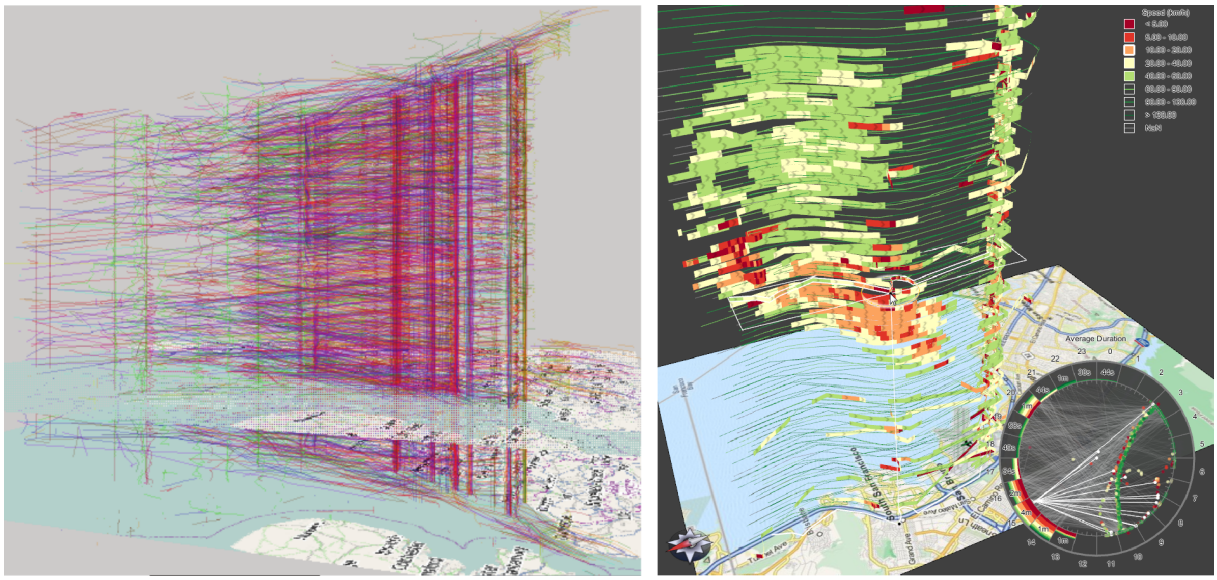


Figure 2.1: Examples for using the third dimension as a time parameter. The left image shows an interactive space-time cube (STC) of ship trajectories with 20% opacity colored by ship types (Figure taken from Andrienko and Andrienko [AA13, p. 4]). The right image shows stacked bands representing trajectories colored by velocities (Figure taken from Tominski et al. [TSAA12, p. 7]).

the usage of range parameters are also relevant for analyzing cell trajectories but their method does not scale to thousands of trajectories and their visualizations have no support for the third dimension to investigate cell migration data.

Multiple Views of Movement Data

Other visual analysis techniques provide a combination of multiple views. Wang and Yuan [WY14] introduce time line visualizations of 2D trajectories in order to better compare several movement properties. They focus on the comparison of geometry features such as velocity, curvature, straightness and a measure for the contributions of turns of a trajectory. The results are visualized using static maps, *heat maps*, and *scatterplots*. The time lines are generated for each computed feature and realized in a 2D heat map and a 3D terrain visualization. These can be sorted based on similarity to reveal similar feature patterns. However, their method is not designed for 3D trajectories and the terrain visualization is hard to interpret for several hundreds of movements. Liu et al. [LGL+11] develop an analysis technique to investigate the route diversity of taxi drivers and to compare different routes. They compute the diversity using a statistical entropy formula. Their system consists of four visualizations providing information about global and local route diversities with heat maps as well as a so-called *trip view* to compare source/destination trajectories and a *road view* to analyze the diversity through a specific road. The multiple displays provide more user flexibility to analyze the data. However, the system is not suited for large data sets from biology because of increasing visual clutter problems. Furthermore, the diversity computation is only designed for static routes.

Hurter et al. [HTC09] present *FromDaDy* for the exploration of aircraft trajectories (left image in Figure 2.2) using visual designs like scatterplots, brushing, pick and drop, and juxtaposed views. Users can set a visual configuration by brushing and picking a region or trajectory

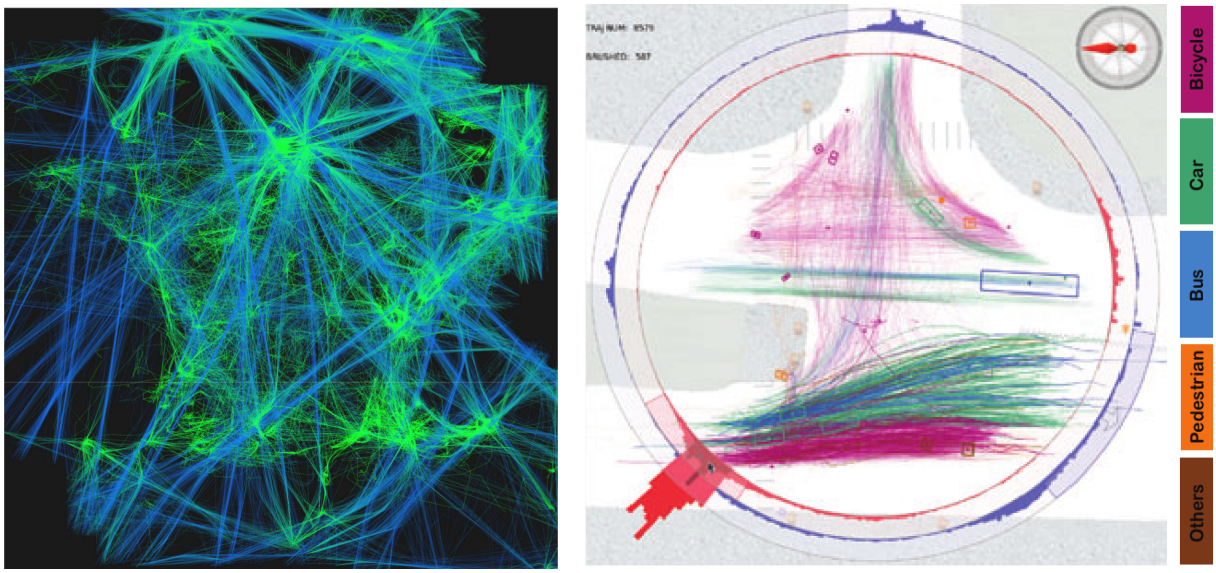


Figure 2.2: Examples for visualizing 2D trajectory data. The left image shows aircraft trajectories over France colored by a gradient from green to blue that represents the altitude (Figure taken from Hurter et al. [HTC09, p. 2]). The right image illustrates traffic trajectories colored by type with an additional circular histogram (Figure taken from Guo et al. [GWY+11, p. 4]).

of interest. The selection can be dropped into a new view for detailed investigation. While the system incorporates a lot of features for the visual analysis of 2D movements, it is not appropriate for processing 3D cell trajectories. Furthermore, the interpretation gets complicated due to overplotting. Guo et al. [GWY+11] present an interactive visual analytics system called *Triple Perspective Visual Trajectory Analytics (TripVista)*. They combine three different perspectives showing spatial, temporal and multidimensional views of the trajectories. They use additional visualizations showing scatterplots and *parallel coordinates* [Ins85] for the purpose of analyzing traffic trajectory data in a region of interest. This region is further analyzed by circular histograms (right image in Figure 2.2). However, their analysis is limited to a static area and does not include any automatic approaches to detect relevant features which is required to process a huge set of cell migrations.

Clustering of Movement Data

In order to cope with large data sets, data mining techniques are used to group trajectories based on movement-based parameters. Andrienko et al. [AAW07] develop an interactive visual framework for the analysis of car and truck movements. They use the density-based clustering algorithm *OPTICS* [ABKS99] in order to group similar trajectories based on user-selected features. The clustering methods is based on neighborhoods of elements defined by user-selected radii. To simplify the visualization, they use different thicknesses for trajectories that correspond to the number of movements with similar directions. In order to apply different distance functions for the similarity measure, Rinzivillo et al. [RPN+08] extend the work and introduce the procedure of *progressive clustering* where a different distance function is used for several cluster steps applied to a sub cluster. Through this, the resulting structured trajectories or clusters can be further refined. Andrienko et al. [AAH+13] refine the visual analysis framework by integrating event clustering of relevant places and the analysis of the aggregated data. For

trajectories, the flow commonly indicates the aggregates movement between the start and end positions of a trajectory (left image in Figure 2.3). The density-based clustering approach could also be applied to cell trajectories but a hierarchical clustering is chosen because biologists are interested in the hierarchy structure of merged cell developments. Furthermore, these approaches are limited to the analysis of 2D trajectories on top of maps.

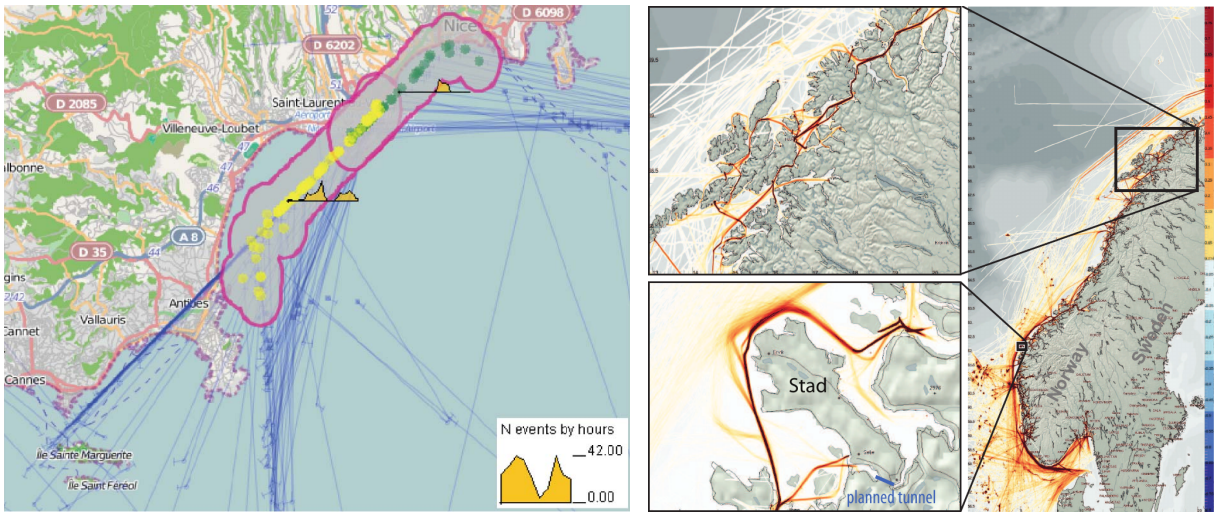


Figure 2.3: Examples of clustered movement data and density fields. The left image shows an example of two clusters (yellow and green dots) grouped based on direction and position that represents two different landing directions of planes (Figure taken from Andrienko et al. [AAH+13, p. 11]). The right image shows density fields of vessel movements around the coast of Norway (Figure taken from Lampe et al. [LKH10, p. 2]).

In another approach, 2D trajectory data can also be grouped spatially into continuous density fields (right image in Figure 2.3) as it is done by Willems et al. [WvdWvW09]. They visualize the fields by illuminated height maps using a kernel density estimation (KDE) method applied to trajectories of vessels. They combine two fields with a large and small kernel to give both an overview of area movements and details of speed variations. Lampe et al. [LKH10] extend this work using a GPU-based implementation and interactive views. Their idea is that the analysis is performed through an iteration of different views to compare attributes such as time, type, and speed. They use a combination of multiple views for each day of a week, histograms, and scatterplots to analyze frequency-based vessel movements. Density maps are well-suited for the visualization of transport behavior of frequently used routes. However, they cannot be applied to 3D cell trajectories because cells can move arbitrarily in the developing embryo following no specific route. This behavior will complicate the visual analysis of many generated density fields.

2.2 Visualization of Cell Trajectories

All analysis approaches explained above are based on 2D trajectory data. The interpretation and analysis of this data is often supported by including road maps. This processing can also be transferred to the analysis of biological cell data to a certain extent. For example, in the 2D case, a projection of the microscopic raw data can be placed underneath the digital cell data or trajectory to provide an overview of cell positions among the organism. In this context, Peng [Pen08]

gives an overview of the advances in the area of *bioimage informatics*, including applications, techniques, visualizations and tools. He points out that an interactive system is required for accessing large-scale biological data. O'Donoghue et al. [OGG+10] and Keefe et al. [KERC09] further present central requirements of a visualization software to process biological data: Usability, visual analytics, and multi-scale representation.

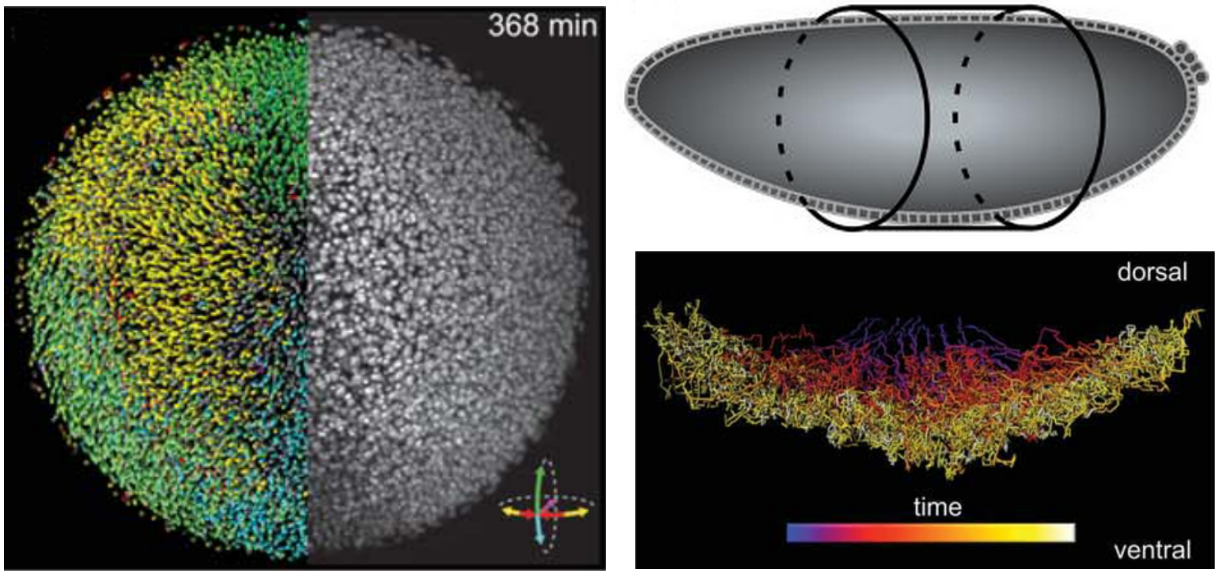


Figure 2.4: Examples of visualizing 3D+t trajectories. The left image shows a split view of colored cell migrations and a maximum intensity projection of the microscopy *zebrafish* data (Figure taken from Khairy and Keller [KK11, p. 8]). The top right image illustrates the cylindrical coordinate system for the *Drosophila* while the image at the bottom right shows cell trajectories colored by their temporal appearances (Figures taken from McMahon et al. [MSFS08, p. 6–7]).

There are several analysis systems for 2D cell trajectories generated from microscopic movie data. Walter et al. [WHN+10] present a framework in which they study cellular phenotypic kinetics in genes. They use segmentation to detect single nuclei from raw movie data and extract several features for each cell. They further apply hierarchical clustering using *Ward's method* based on the Euclidean distances between starting, middle and end points of the trajectories. Additionally, the cells are classified into morphological classes such as *interphase* or *cell death*. This classification is then visualized in a time series representation to observe the occurrence of cell classes over time. However, their framework is based on 2D data and the similarity analysis is limited to position information of trajectories. Costa and Schubert [dFCS03] introduce another framework of measurements to characterize 2D cell trajectories given by a movie. They focus on the behavior of individual cells, the interaction between a pair of cells and the interaction of a cell with its environment. For this purpose, they consider properties like the *displacement effectiveness*, the *maximum dispersion* or *instant attraction* of trajectories. But their method does not consider 3D cell trajectories and additional features such as their shapes. Slater et al. [SLM13] present a visualization to analyze collective cell migrations. They visualize 2D human cell trajectories as streams colored by their motion directions in the process of wound healing. In order to detect collective behavior they compare the angle of displacement of surrounding cells to the angle of displacement of the comparison cell. If the angle is smaller than a certain threshold both cells are designated as correlated. The visualization can

handle thousands of cell trajectories but their similarity analysis is limited to the observation of directions in a bounded neighborhood which is not suited for a similarity analysis of different features all over the embryo.

Only very little research has been done in the visual analysis of 3D trajectories. Especially in developmental biology, domain experts are interested in the development of cells in the complete embryo. The reason for the lack of visual analysis methods in this area is due to missing of prior techniques to generate this 3D+t data. Additionally, the creation of interactive visual similarity analysis techniques for 3D data is a huge challenge. One example for an existing visualization is the augmentation of the raw data with additional information to obtain spatial properties of cells. A popular approach is the color-coding of them based on derived features such as the directions of motion (left image in Figure 2.4) as it is done for the early *zebrafish embryo* [KSWS08, KK11]. However, this representation does not allow any similarity analysis between individual cell movements. McMahon et al. [MSFS08] investigate collective cell migrations in the fruit fly embryo called *Drosophila*. Based on the geometry of the model organism, they use a cylindrical coordinate system to compute and visualize the motion directions of 3D cell trajectories (right image in Figure 2.4). However, their quantitative analysis is designed for the *Drosophila* and thus it cannot be easily applied to any other model organisms. Langenberg et al. [LDO+06] develop a tool called *TracePilot* that enables the interactive manipulation and visualization of tracking data. They investigate 3D+t cell movements in the developing zebrafish brain which are illustrated by moving spheres colored by four different group assignments. The tool allows an investigation of single cell displacements but it does not feature any similarity analysis methods applicable to cell trajectories.

2.3 Visual Analysis of Tree Collections in Biology

In developmental biology, cell developments are commonly depicted as cell lineages (binary trees). Thus, in order to explore them, similarity analysis techniques for tree structures are required. Landesberger et al. [vLKS+11] present a general review of available state-of-the-art methods for the visual analysis of large graphs. The visualization of trees (static graphs) is usually simpler than the one of general graphs. In this context, Ziemkiewicz and Kosara [ZK08] point out that the applicability of a certain tree visualization depends not only on the task, but also on the formulation of the task assignment. Graham and Kennedy [GK10] as well as Schulz [Sch11] give extensive reviews about visualizations for single trees, pairs of trees and multiple tree collections. For the visual analysis of cell lineages in particular, only few work has been done.

Cedilnik et al. [CBI+07] propose a visualization framework, called *Titan project*, for the visualization and validation of 2D lineage data from the roundworm *Caenorhabditis elegans* (left image in Figure 2.5). They combine linked cell lineage visualizations and volume rendering of the microscopy data for a concurrent investigation of both data modalities. However, their system is not designed for the visual analysis of thousands of cell lineages and they do not support any analysis methods to compare different features of cell tracks. In a similar approach, Boyle et al. [BBM+06] present a visual analysis system called *AceTree* to track expression of genes. The visualization links annotations and images in cell lineages of the nematode *C. elegans*. In an additional 3D view, nuclei are drawn as spheres and colored by their gene expression. Zhao et al. [ZBB+08] extend the analysis by comparing *C. elegans* with the closely related

worm *C. briggsae*. But also their visualizations do not scale to thousands of cell lineages which prevents a direct comparison of trees. Furthermore, both aforementioned systems are limited to single features that are color-coded in the cell lineages.

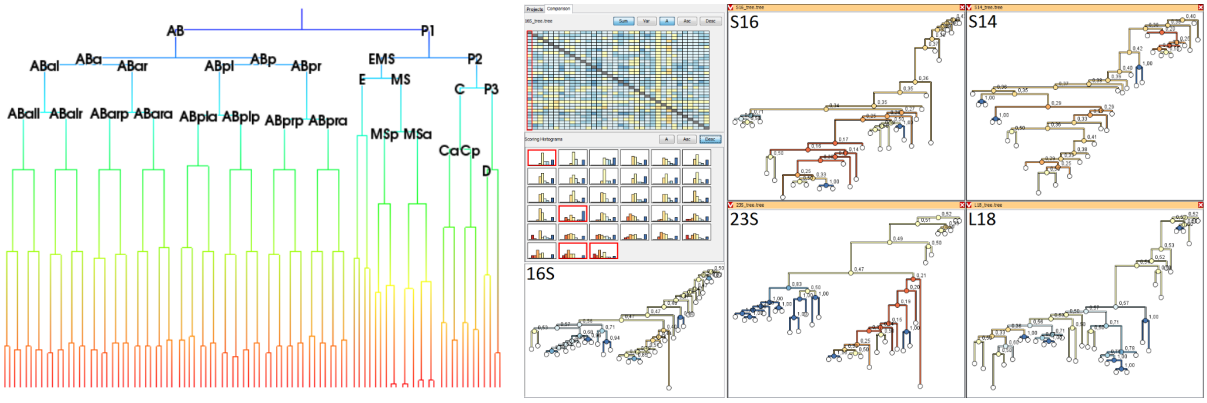


Figure 2.5: Examples of visualizing and comparing tree structures from biology. The left image shows a cell lineage tree of the worm *Caenorhabditis elegans* colored by the temporal development (Figure taken from Cedilnik et al. [CBI+07, p. 8]). The right image shows a visual analysis system for comparing *phylogenetic trees*. The visualization consists of a color-coded matrix, a set of histograms and detailed tree representations of selected trees (Figure taken from Bremm et al. [BvTLH+11, p. 7]).

Further approaches less related to cell lineages but nevertheless interesting are presented for *phylogenetic trees*. These are of high interest for cell biologists [PTL+10]. The leaf-labeled trees illustrate evolutionary relationships among groups of organisms (*taxa*). One example for a visualization system is *TreeJuxtaposer* [MGT+03]. It combines visual analysis of tree differences with interactive leaf similarity highlighting. The similarity measure is realized by associating each node to its most similar (best corresponding) node in another tree. Inspired by the *Robinson-Foulds metric* [RF81], a distance measure between unrooted phylogenetic trees, they compare two internal nodes according to the sets of labeled leaves underneath them. Although their visualization focuses on highlighted differences in trees using colors, the visual representation is not suited for a huge collection of cell lineages. Moreover, their similarity measure is designed for leaf-labeled trees and cannot be applied to unlabeled cell lineages. Bremm et al. [BvTLH+11] present an interactive visual analysis system on multiple levels of detail (right image in Figure 2.5). They use a set of similarity scores that are geared towards phylogenetic trees. A reference tree is selected to visualize its difference to all other trees. These differences are depicted in a color-coded matrix and histograms based on three different similarity measures: Leaf-based, element-based, and edge-based measure. The leaf-based measure is the Robinson-Foulds metric mentioned above. They compute an element-based measure according to the number of partitions in a tree to include information of inner structures. The edge-based variant considers different lengths of edges. Their system is well-suited for a global and local comparison of the computed similarity scores and the latter two similarity measures can also be applied to cell lineages. However, the visual design does not scale to data sets in developmental biology with many thousands of trees and the selection of tree features is limited.

2.4 Summary

In this chapter, I presented an overview of available visual analysis methods applied to traffic and transport data as well as cell trajectories and trees in biology. A lot of research has been done in the area of geographic information science and transportation engineering. However, most of these methods are designed for 2D+t trajectories and cannot be applied to the analysis of cell trajectories in 3D. Far too little research has been done for a visual analysis of 3D+t data. Due to the fast progress of new data acquisition techniques in developmental biology, there is a need for interactive visual analysis methods to analyze this data. This means that interactive visualization methods are required to detect and analyze similarities also for large data sets. In the following chapters, I introduce new visualization approaches that permit an interactive visual analysis of these 3D+t data sets. Prior to their explanation, required preprocessing steps and migration terms are described in the next chapter.

Chapter 3

Data Processing

"The goal is to transform data into information,
and information into insight."
— Carly Fiorina,
in "Information: The currency of the digital age", 2004

The recent available 3D+t data sets of living organisms contain a wealth of biologically relevant and quantifiable information such as cell migrations, cell division orientations, cell growth, and cell rearrangements. Model organisms such as the *zebrafish* or the *Arabidopsis* plant are especially well-suited for experimental manipulation and microscopic observation of these developments. As already mentioned in Section 1.1.2, the fundamental question for developmental biologists is how in multicellular organisms cells proliferate and ensure the reproducible generation of accurate shape. The biologists desire a quantitative analysis of the *morphogenesis*, i.e. the process that causes an organism to develop its shape, of these multicellular organisms to extract general principles of underlying shape formations. The investigation of cell migrations and the analysis of their spatial similarities is an important step towards the understanding of cell behaviors and their identities.

In order to enable a quantitative analysis of cell migrations and cell divisions in both model organisms, several preprocessing steps have to be applied to the raw data. These steps are illustrated and explained in the pipeline in Figure 3.1. The pipeline is novel with regard to the way how *cell paths* and *cell trajectories* are extracted from the data.

3.1 Data Acquisition of 3D+t Biological Data

The biological data sets of both the zebrafish as well as the Arabidopsis are generated using *digital scanned laser light sheet fluorescence microscopy (DSLM)* that is developed by Keller et al. [KSWS08]. The main idea of this recording technique is to excite selectively a small "slice" of the specimen and to detect the light emitted by the excited fluorophores by a second lens located orthogonally to the first one. Figure 3.2 shows the hardware setting of the microscopy. The DSLM features several advantages for recording the specimen. For example, in contrast to confocal or epi-fluorescence microscopes in which the whole specimen is

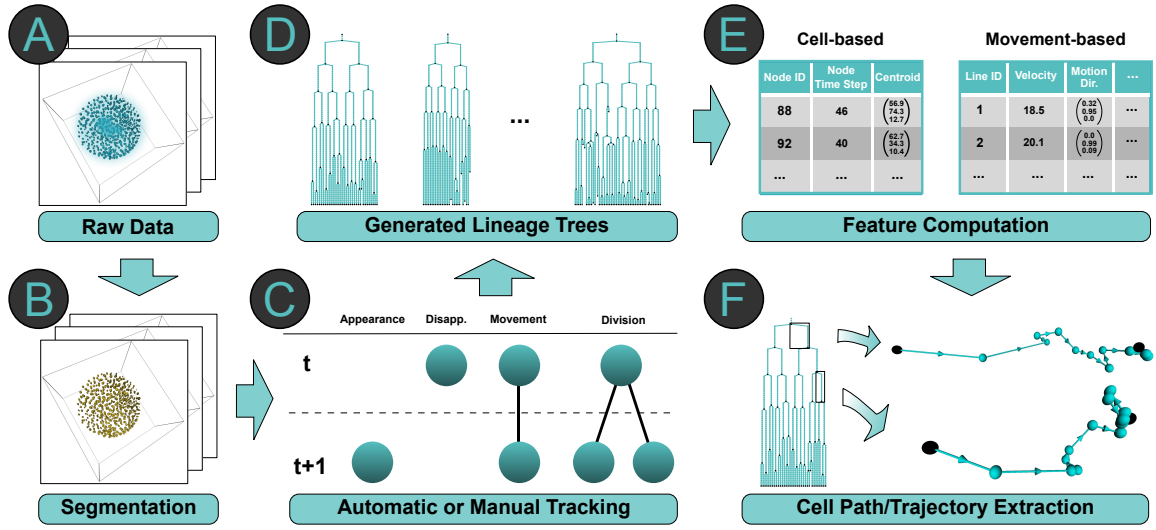


Figure 3.1: Pipeline for data processing. (**A**) The 3D+t raw data is generated by light sheet microscopy (Section 3.1). (**B**) The data is segmented either manually or automatically in order to identify single nuclei (Section 3.2). (**C**) After segmentation, the cells are tracked manually or automatically and cell developments are traced (Section 3.3). (**D**) These cell developments are depicted as *cell lineage trees* [Chi01]. (**E**) The segmentation and tracking results are used to compute cell-based and movement-based features (Section 3.4). (**F**) Based on the lineage tree structure and computed features, *cell paths* and *cell trajectories* are extracted (Section 3.5).

ill-treated, DSLM excites only those fluorophores in the illuminated plane of interest thereby reducing phototoxicity. In addition, all planes are illuminated with the same intensity which supports the overall recording of the whole embryo. DSLM also offers a high recording speed of 63 million voxels per second [KSWS08], enabling high temporal resolution recordings that are essential for cell tracking.

The position of the specimen in the sample chamber is different for zebrafish and Arabidopsis. Keller et al. [KSWS08, Kel13] describe the imaging process of the *digital embryo* using the zebrafish as a model organism. Initially, the embryo is put into a glass capillary filled with microliters of agarose gel. Before the imaging process, the gel containing the embryo is cautiously pushed out of the capillary. This is required in order to avoid the detection lens as well as the light sheet to pass through the glass wall of the capillary. The Arabidopsis seedlings on the other hand, like any other plant type, need sunlight to grow. Thus, they require another type of specimen chamber which fits the requirements of the development. The detailed imaging process is explained by Maizel et al. [MvWF$^+$11] and Wangenheim et al. [vWDL$^+$14]. In a nutshell, the plant grows vertically with leaves in the air and the root on the surface of an organ medium. The plant is held in the microscope chamber from the bottom by a capillary. The plant is germinated on half-strength *MS medium* (named after the invented medium of Murashige and Skoog [MS62]). Seven days after germination, the seedlings are transferred to the holder setting for the specimen. The leaves are supplied with a lighting system from above in order to simulate the sunlight (Figure 3.3C). The specimen chamber also consists of a perfusion system that exchanges the whole volume of the chamber every 15 minutes in order to avoid contaminants and toxic compounds. Analogous to the zebrafish recording, the Phytagel cylinder is ejected from the capillary during the imaging process (Figure 3.3D).

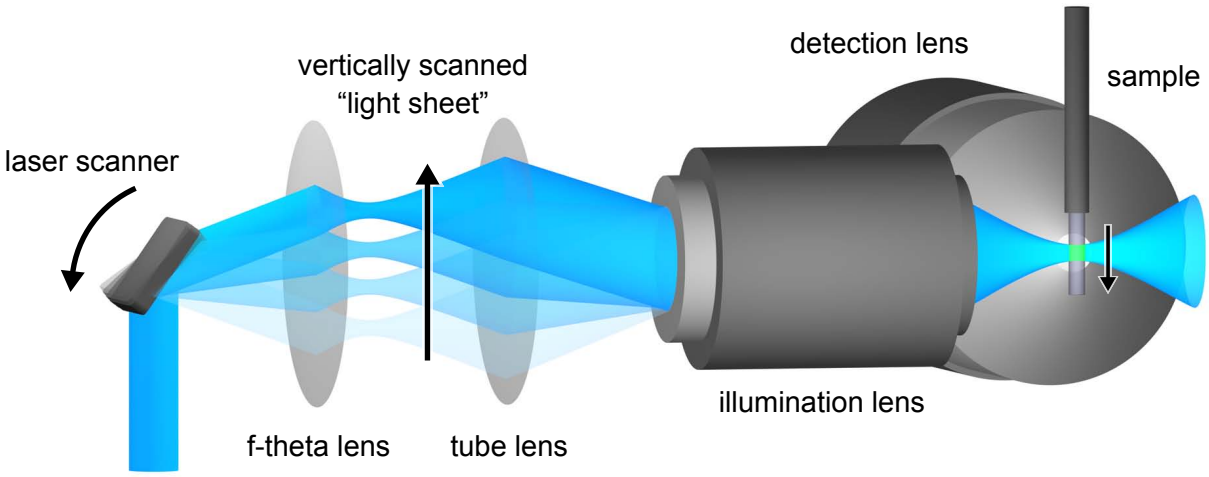


Figure 3.2: Digital scanned laser light sheet microscopy. A μm-thin laser beam scans the sample vertically and excites fluorophores along a single line. The f-theta lens is used to adjust the laser beam vertically while the tube lens and illumination lens focus it on the sample. The detection lens is perpendicular to the illumination and captures the marked nuclei along multiple directions in such a way that hidden parts of the specimen are also recorded (Figure taken from Keller et al. [KSWS08, p. 1]).

The initial nucleus at the one-cell stage (*zygote*) of the zebrafish is labeled by an mRNA injection of H2B-eGFP, a fusion protein of human histone-2B and enhanced green fluorescent protein (GFP) reporter [KSWS08]. This process makes it possible to observe cell positions in the imaging process after a few hours post fertilization (hpf) and injection since the protein does not take effect immediately. In contrast, the plant is injected with three different markers: a pan-nuclear marker (pUBI::H2B-RFP), a plasma membrane marker (Wave131Y) and a lateral root primordium specific marker (pGATA23::nGFP-GUS). Through this, the cell nuclei as well as the cell contours can be recorded simultaneously.

The recording time can range from several hours to a few days. The volume of the specimen is captured along two opposing directions in equidistant time distances. The resulting raw microscopy data of the zebrafish is stored in a *HDF5* (http://www.hdfgroup.org/HDF5) file format for each time step. This file format is designed for managing large and complex data sets and can be easily extended by additional information such as segmentation and tracking information. The Arabidopsis volume data is stored in several TIFF images with z-stacks for the third dimension for each time step. The following two subsections provide more detailed information about the two recorded model organisms.

Zebrafish - *Danio rerio*

Figure 3.4 illustrates the different stages of the zebrafish development. Two time-delayed periods of the vertebrate zebrafish growth are recorded. The two periods differ significantly in their cell behaviors. The first period covers the embryogenesis in the early *epiboly* stages (time step range indicated by green line in Figure 3.4). The term *epiboly* denotes the cell migration of an outer cell layer above an inner cell layer. Epiboly refers to the first coordinated cell migration event in the zebrafish, in frogs, and many other vertebrate species. Here, numerous cell

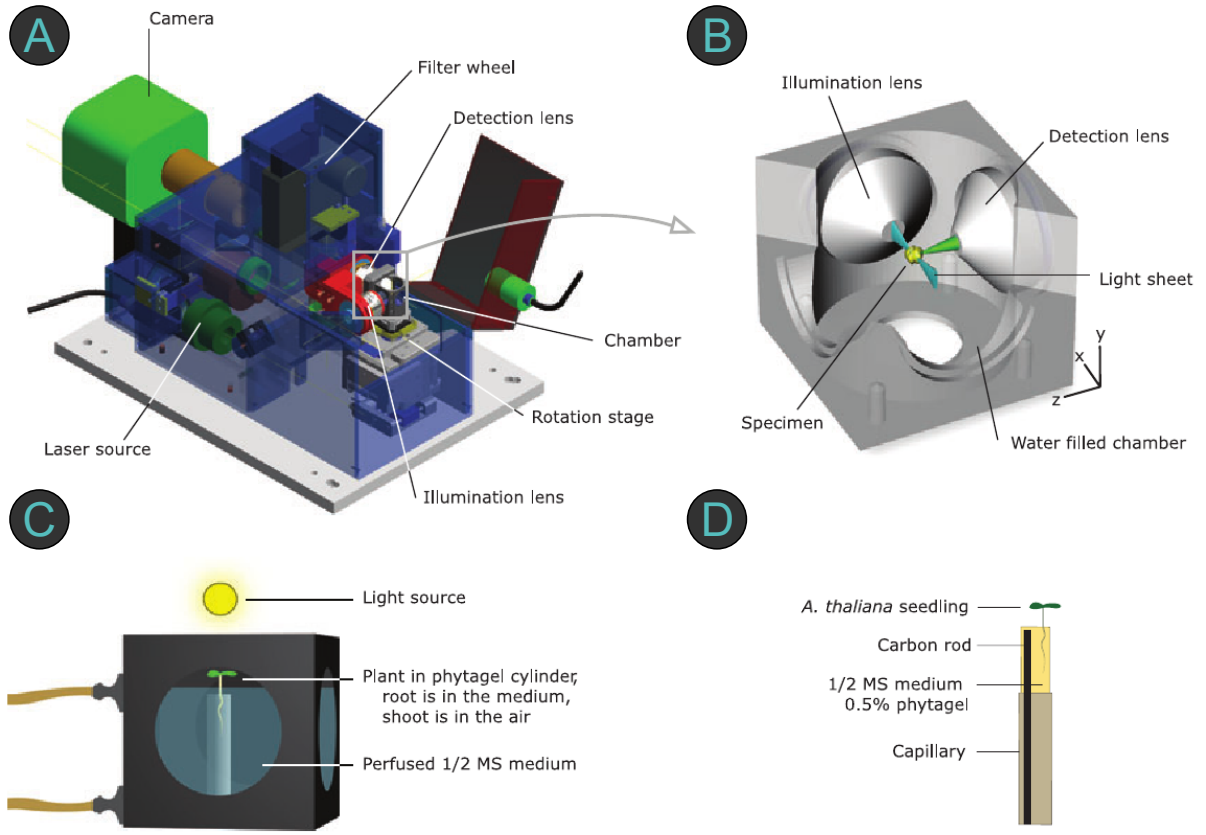


Figure 3.3: Setting of light sheet microscopy for imaging Arabidopsis plants. Image (**A**) shows the setting of the required hardware elements of the microscope. (**B**) shows the specimen chamber that is filled with water. The fluorophores inside a thin planar volume in the center are excited by sending a μm-thin Gaussian laser beam inside the specimen. The volume detection is realized perpendicular to the illumination direction. Image (**C**) illustrates the configuration of the Arabidopsis seedling within the chamber. The *basal* (root) is growing in a *Phytagel* cylinder located in the water while the *apical* (shoot) is situated in the air. A light source is coming from the top to simulate sunlight. Image (**D**) illustrates the holder setting used in the imaging process (Figure modified from Maizel et al. [MvWF+11, p. 2]).

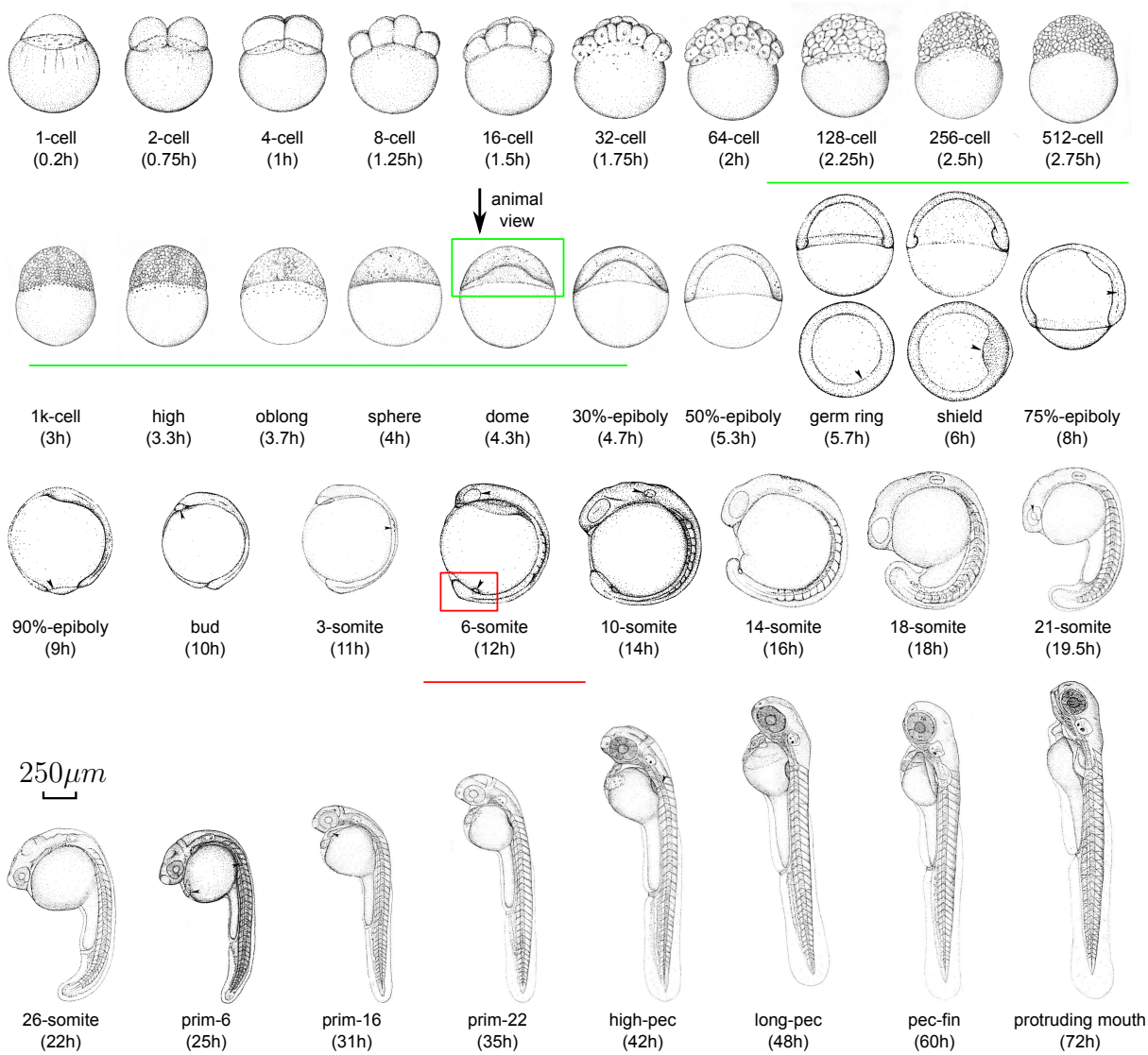


Figure 3.4: Stages of embryonic development of the zebrafish. The fertilized egg is in the *zygote period* $(0 - 0.75\text{h})$ until the first cleavage has occurred after 45 minutes. In the next *cleavage period* $(0.75 - 2.25\text{h})$, cells divide at about 15-minutes intervals until the 64-cell stage is reached. The next stage is called the *blastula period* $(2.25 - 5.25\text{h})$ in which the *blastodisc* starts to form ball-like until the beginning of the *gastrulation*. The *gastrula period* $(5.25 - 10\text{h})$ begins at *50%-epiboly* and at this stage cell movements start to generate the *primary germ layers* and the embryonic axis. The *somites*, i.e. the divisions of the body of the animal, develop based on a variety of cell movements followed by the *segmentation period* $(10 - 24\text{h})$. Here, the tailbud evolves and the embryo elongates. The *pharyngula period* $(24 - 48\text{h})$ refers to the stage in which the embryo evolves similar to other vertebrates. During the *hatching period* $(48 - 72\text{h})$, the embryo continues to develop into the early larval stage. The green line illustrates the recorded time step range of the epiboly data set while the green rectangle shows an example of the perspective viewing from the top, referred to as *animal view*. In contrast, the red line and rectangle indicate the time and region of record of the tailbud data set (Figure modified from Kimmel et al. [KBK+95, p. 4–7]).

divisions and less cell migrations occur. Thus, homogeneous cell behavior and numerous stem cell divisions are expected. Stem cell divisions are commonly classified as being symmetric or asymmetric. A symmetric division results in two identical stem cells whereas an asymmetric division generates one stem cell and one progenitor cell. While stem cells can replicate indefinitely, progenitor cells can only divide a limited number of times. Progenitor cells also divide faster than stem cells. In contrast, the second data set is recorded in a later stage of the zebrafish development. The initial structure of the fish has already been formed and the tail is growing (time step range indicated by red line in Figure 3.4). Here, long cell migrations dominate over less and slower cell divisions. Note that the time ranges differ significantly between the two data sets. The reason for the short record time of the tailbud data set is that at this stage the quality of data acquisition is significantly reduced for several of thousands of crowded cells.

Epiboly Data: The first experimental data set is called the *epiboly data set*. In Figure 3.5A, I generated maximum intensity projections (MIPs) of the early events from *blastula* to early *epiboly* stages ($\approx$ 3.5–4.5 hpf). The MIPs are seen from the *animal view* of the embryo. In early embryogenesis, the development locations can be distinguished between the *animal pole* and the *vegetal pole*. The former term refers to the area in which the cells develop dominantly

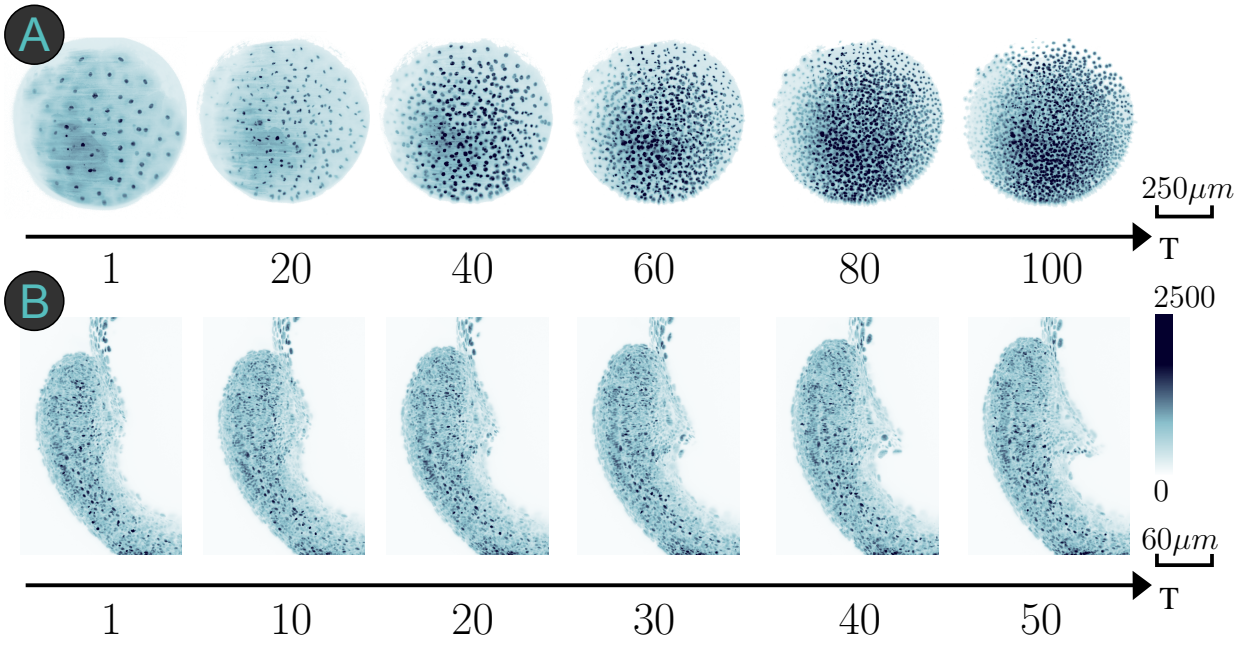


Figure 3.5: Maximum intensity projections (MIPs) of zebrafish development. In (**A**), the images show the temporal development of the epiboly data set from early events in the *blastula* to early *epiboly* stages for six specific time steps. The images are taken from the *animal view* direction. The images illustrate the occurrence of many cell divisions at this stage. In (**B**), the tailbud data set is shown with the tail starting at *5 somite* stage for six specific time steps. Here, less cell divisions occur but with more cell migrations. The embryo is colored based on the intensity level of the nuclei and membrane.

at the beginning embryo and where rapid cell divisions occur. This region is focused in the epiboly data set. The vegetal pole mostly consists of large yolky cells that divide very slowly (embryogenesis in first row of Figure 3.4). The embryo's cells migrate, spread and thin to cover the entire surface of the yolk cell.

Data set	Time steps	Cropped dimension [pixels]	Rec. time [hours]	Cells at start	Cells at end	Cell lineages	Size [GiB]
Epiboly	100	$332 \times 1111 \times 1161$	2.5	90	3,253	4,896	13
Tailbud	50	$1226 \times 834 \times 504$	0.83	9,961	10,173	58,048	26

Table 3.1: Properties of zebrafish data sets.

The data set consists of 100 time steps with a spatial discretization of 90 seconds, resulting in a total captured record time of two and a half hours. The data starts with 90 cells since the injected protein does not take effect earlier for detecting the nuclei. These cells develop into $3{,}253$ cells due to numerous and fast cell divisions while less cell migrations can be observed. Note that the numbers of cells are a result of the segmentation process which is explained later. The microscopy data has a total size of 13 GiB in the compressed HDF5 file format.

Tailbud Data: The second data set illustrated in Figure 3.5B covers the tail extension ($\approx$ 12–13 hpf) of the zebrafish and has a size of 26 GiB in the HDF5 format. This data set, from now on called the *tailbud* data, shows the growing zebrafish tail starting at *5 somite* stage with a temporal resolution of 60 seconds. The expression *tailbud* describes the proliferating mass of cells located at the posterior of an embryo. The margin of the spreading cell mass is called the *blastopore* as a result of the *gastrulation* to form the tailbud. *Gastrulation* patterns the head and trunk regions of the embryo and shapes the main head-to-tail body axis. Subsequent to *gastrulation*, the tailbud continues to elongate this body axis and develops into the hindmost tissues of the body. At the first time step of the recording, $9{,}961$ cells can be identified by segmentation while after 50 time steps, $10{,}173$ cells are detected. The overall recording time is approximately one hour and this relatively slow increase in cell numbers in contrast to the epiboly data set is caused by less cell divisions and more cell migrations. An overview of the different properties of the data sets is given in Table 3.1. Note that the original record is longer than the actual record time given in Table 3.1. Especially in later time steps of the development, the quality of data acquisition sometimes is too low such that segmentation and tracking results are insufficient. Thus, only specific time step ranges are considered that yield satisfactory results.

Arabidopsis thaliana

Figure 3.6 on page 26 illustrates the structure and tissues of the Arabidopsis root. The primary root has been developed during morphogenesis; from this main root, lateral roots are regularly initiated under the action of the hormone *Auxin* [OFB10]. The cell layers of one lateral root (yellow) denote different tissues of cells that form a dome-like structure like a set of stacked caps on top of the lateral root. I investigate five data sets (named after their date of genesis) of the lateral root growth of different plants in order to find similarities in division patterns among different data sets. During less than six cell cycles, different cell layers are generated, gradually forming the dome shape structure of the lateral root. All data sets share the same biological event of the lateral root growth. Except for the data set 121211, the initial cells of all data sets start in one layer. For this data set, in contrast to the other ones, the first division is not captured in the segmentation. In Figure 3.7, I created several MIPs for five time steps of the data set

130508. They illustrate the lateral root development in a side and radial view. The resolution of the recorded volume is $696 \times 520 \times 233$ for all data sets, where 233 is the number of planes of the stack during the imaging process. After data acquisition, subsequent time steps of the volumes

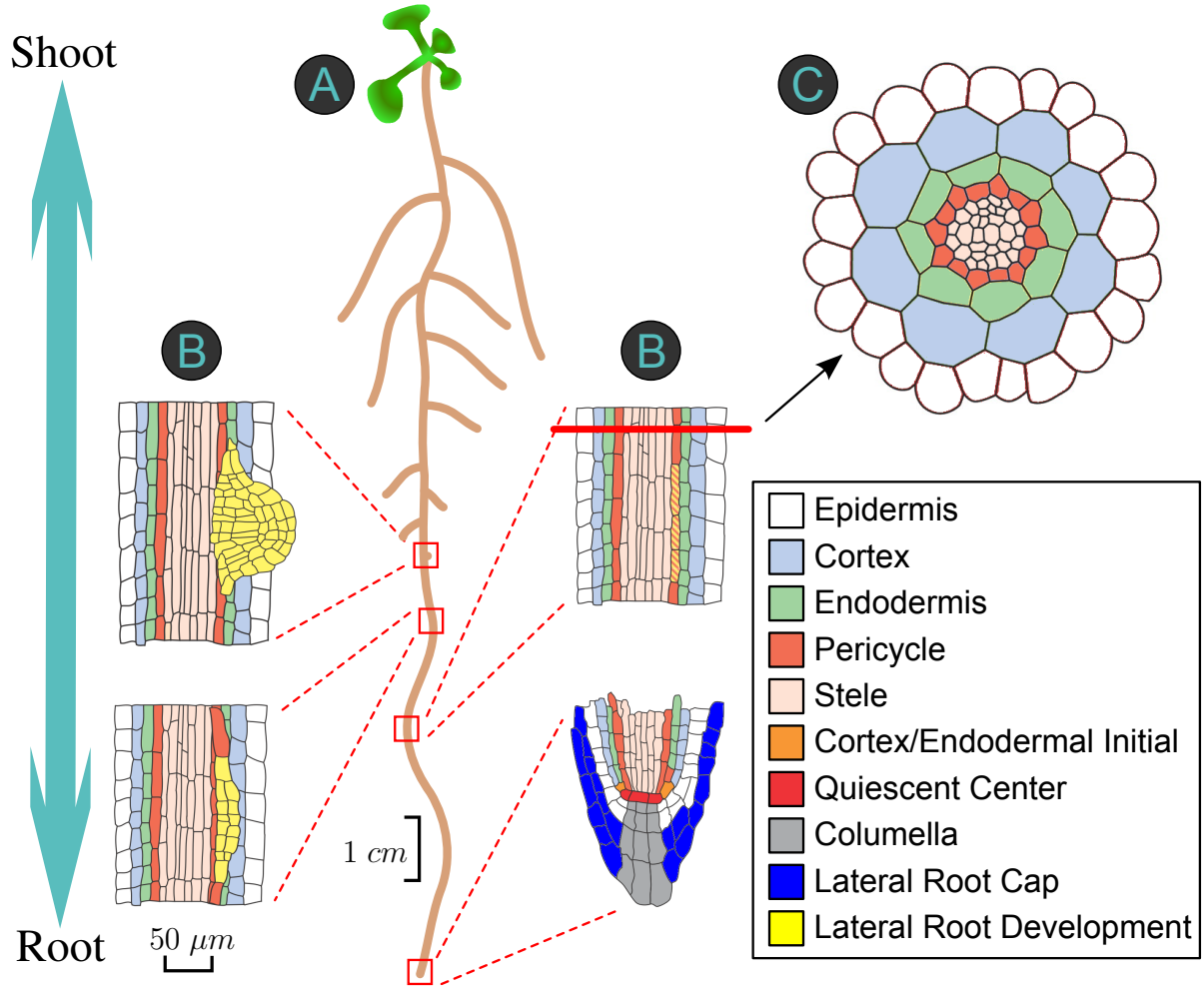


Figure 3.6: Organization of Arabidopsis and the lateral root. Depending on the growth process of the Arabidopsis (**A**), the lateral root (yellow) develops horizontally from the primary root into a branch. It forms a dome-like structure (**B**) and serves to anchor the plant into the soil. The lateral branch also has the purpose to supply the plant with water and nutrients required for its growth. In the stem (brown), the main tissues are named (starting from the outside) *epidermis*, *cortex*, *endodermis*, *pericycle* and *stele* [MB97] as the central part of the stem (**C**).

suffer from *sample drifts*, i.e. the event by samples moving outside of the focused x-y position. Without correction, sample drifts lead to blurred images with decreased resolution, and even to misinterpretations of relevant structures [MSC+11]. The drift is corrected with the *Correct 3D Drift* plugin [CBR10] of the open-source biological-image analysis software Fiji [SACF+12]. For each data set, the empty space beyond the bounding box of the volume is cropped, resulting in the final dimensions given in Table 3.2.

The volume of the specimen is recorded and stored as TIFF stack every 5 minutes. The biological growth event is illustrated by MIPs in Figure 3.7. Analogous to the zebrafish data, the recording time is much longer (see values in Table 3.2) than the segmented and tracked time

Data set	Time steps	Cropped dimension [pixels]	Rec. time [hours]	Cells at start	Cells at end	Cell lineages	Size [GiB]
120830	300	$555 \times 221 \times 147$	47	10	176	10	100
121204	300	$689 \times 393 \times 200$	45	15	160	15	167
121211	300	$736 \times 376 \times 170$	39.5	18	260	18	152
130508	350	$682 \times 406 \times 130$	50.5	9	143	9	432
130607	300	$666 \times 404 \times 130$	64	15	267	15	243

Table 3.2: Properties of Arabidopsis data sets.

range of 25 hours $\left(\frac{5 \cdot 300}{60} = 25\right)$ and $29^1/_6$ hours $\left(\frac{5 \cdot 350}{60} = 29^1/_6\right)$, respectively. The data sets are captured for 300 or 350 time steps because these time steps cover the event of interest and in later time steps the image quality becomes very blurred which complicates the segmentation and tracking.

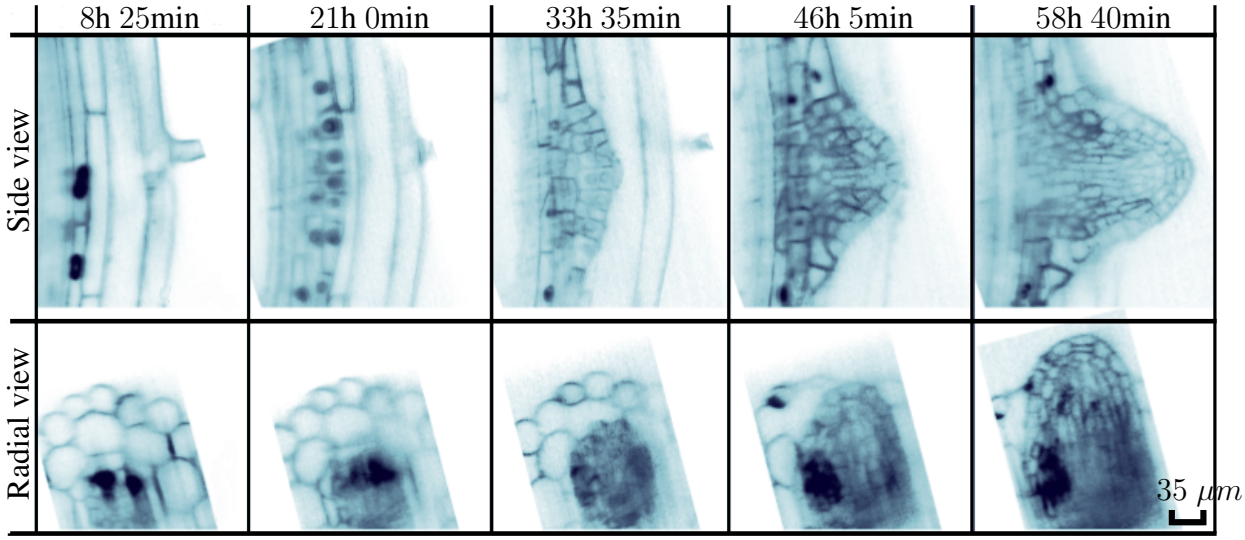


Figure 3.7: MIPs of lateral root development for five time steps. The images show the temporal development of the dome-like structure of data set 130508 in side and radial view.

3.2 Segmentation

The segmentation process describes the detection of single objects of interest in a volume or image, in this case, cell nuclei. Here, this process is realized both manually and automatically. The zebrafish data, consisting of several thousands of cells, requires an automatic approach due to the large number of cells while the segmentation of the Arabidopsis data is handled manually due to their relatively small number of cells (< 300) which results in a more accurate identification of cell developments. Note that in the zebrafish data the cells are migrating while there are no cell migrations in plants. For both data sets, only the single nuclei displacements

are detected.

Automatic Segmentation

The zebrafish data set is segmented using an automatic segmentation approach introduced by Lou et al. [LKL+11]. The segmentation problem is modeled using a *Markov Random Field* (MRF) [Li09] model. The main idea is the minimization of an energy function, consisting of several weighted terms. These terms are weighted differently to satisfy certain constraints such as spatial, shape and length regularizations of voxel data [LKL+11]. The solution to this minimization problem is obtained by applying the *max-flow min-cut theorem*. The output of the segmentation is a binary image that differentiates between the background and cell nuclei. This image is further processed with the *Rosenfeld-Pfaltz Labeling* algorithm [RP66] in order to get a list of individual nucleus objects. Table 3.1 on page 25 lists the number of segmented cells at start and end of recording. Since this process is not central to the thesis, I refer to Lou et al. [LKL+11] for further details. The automatic approach for the zebrafish data yields many erroneous cell lineage trees for which single cell events are missing. Yet, I am able to extract a set of lineage trees (≈ 40) that contain enough cell developments to apply an adequate similarity analysis.

Manual Segmentation

For the Arabidopsis data set, a manual segmentation is applied using the computational software program *Mathematica* (http://www.wolfram.com/mathematica/). A manual approach is chosen because the number of cells is relatively small to be processed manually and it minimizes the segmentation error. When viewed directly by domain experts, the highest detection rate can be achieved. Furthermore, the signal-to-noise ratio depends on the development stage of the lateral root. In advanced stages, the imaging quality is poor and would complicate an automatic detection of cells deep in the primary root of the plant. To minimize the effort of manually segmenting each cell nucleus for each time step, only dividing cells as well as their daughter cells are segmented. The three-dimensional positions in-between are then interpolated linearly. For my purpose, this simplification is valid for the Arabidopsis data because I want to analyze cell division patterns and trends of nuclei displacements.

3.3 Tracking

The segmentation of the data sets yields a set of individual cell nuclei for each time step. However, the temporal information of cell developments is still missing. This information is obtained by applying a *tracking* on the detected cells. Through this, single cells and all their subsequent cell divisions are traced over time, e.g. cell nuclei in each time step are assigned to IDs to be able to track different cell events. Lou et al. [LKL+11] consider four different cell events that can occur between two subsequent time steps: *Cell appearance*, *disappearance*, *movement* and *division*. In order to generate cell tracks, each of these cell events, except cell movement, is assigned a cost constant. The event types plus the associated costs are used to define an integer linear programming (ILP) problem for finding the optimum joint association [LKL+11]. The constants for cell appearance and disappearance are chosen in such a way that both events are heavily penalized in contrast to a cell division. This is motivated by the fact that appearing without prior cell division or disappearing cells are not biologically plausible. Thus, the penalty

serves to support the generation of cell tracks. In fact, a cell appearance in a subsequent time step would not occur in a perfectly segmented data set. However, in the automatic tracking results of the zebrafish data, such events happen due to errors in segmentation. The biological equivalence for a disappearance would be *apoptosis*, i.e. programmed cell death which is not plausible in these data sets. The manual tracking is realized in combination with the segmentation in *Mathematica* because both events can be processed manually at the same time. Note that in a perfect data set, the number of cells in the first time step defines the number of total cell lineages. Yet, since appearances and disappearances are allowed, the tracking delivers more lineages for the zebrafish data than the number of cells at the initial time of recording. Because of the manual segmentation and tracking of the Arabidopsis data, the number of cells at the start and the number of cell lineages are identical. Thus, the quality of the plant data is higher and more accurate than the zebrafish data that is segmented and tracked automatically.

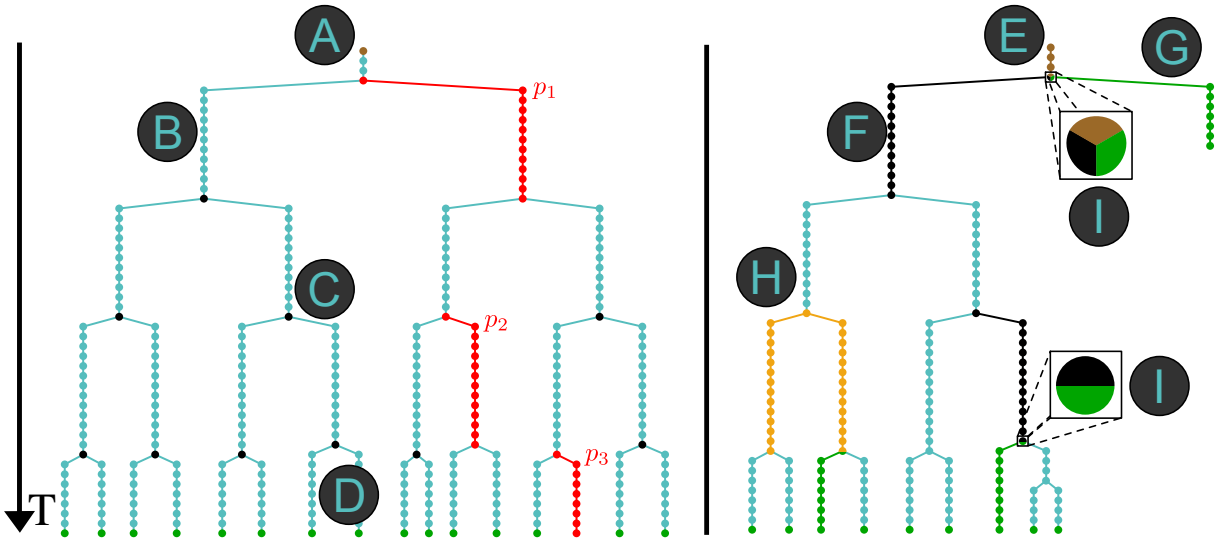


Figure 3.8: Cell lineage trees with examples of cell migrations. The left image shows a cell lineage tree which is a binary tree that represents a single cell development evolving in top-bottom direction. The aforementioned cell events correspond to four different node types: root node (**A**, brown), movement nodes (**B**, cyan), division nodes (**C**, black), and leaf nodes (**D**, green). p_1, p_2 and p_3 are examples for cell paths. The right side illustrates examples for a cell root path (**E**, brown), cell division path (**F**, black) and cell leaf path (**G**, green). (**H**, orange) shows an example of a cell branch. Note that a division node can be shared by two or three different paths marked by a node with multiple colors (**I**).

Acquisition, segmentation, and tracking of the data are realized by collaborating domain experts. In the following, I describe my further processing steps that are required for the similarity analysis methods. Each generated cell track is depicted as a cell lineage as shown in Figure 3.8. This tree is drawn using the Reingold-Tilford algorithm [RT81]. From this point on, the words *node* and *cell* will be used interchangeably. The root node is the first segmented cell tracked in the initial time step. This time step does not necessarily have to be the first time step of the recorded data; it can also start at a later time point because the appearance of a cell is allowed. A short cell cycle at the beginning of a lineage tree is caused by the fixed start time of recording. This behavior can occur for several lineage trees because it cannot be guaranteed that the initial cell cycles are fully captured.

3.4 Feature Computation

Based on the results of segmentation and tracking, I compute for each cell several feature values relevant for the similarity analysis. I distinguish between *cell-based* and *movement-based* features. A cell-based feature describes a property focused at a single cell (e.g. cell position). Movement-based features in contrast are defined by the cell migration or division behavior between at least two subsequent time steps. The position information is given by the segmentation result while for the movement-based features, I employ the tracking in order to get specific cell development information over time. Note that the data sets are stored as volumes, thus each cell is identified by a set of $n \in \mathbb{N}$ voxels. For the similarity analysis, I consider a single cell-based feature which is the centroid of a cell:

- **Centroid:** The centroid $\vec{C} \in \mathbb{R}^3$ of a cell is computed by the center of the voxel positions of the segmented cell nuclei:
$$\vec{C} = \frac{1}{n} \sum_{i=1}^{n} \vec{p}_i, \tag{3.1}$$
where $\vec{p}_i \in \mathbb{R}^3$ denotes the coordinate of the i-th cell voxel.

With the additional information of tracking, the following movement-based features are computed:

- **Motion direction:** The motion direction vector $\vec{m} \in \mathbb{R}^3$ of a cell migration or division between two subsequent time steps is given by the direction vector from one cell centroid at time step t to its subsequent cell centroid at time step $t + 1$:
$$\vec{m} = \vec{C}_{t+1} - \vec{C}_t, \qquad \vec{C}_t, \vec{C}_{t+1} \in \mathbb{R}^3. \tag{3.2}$$
- **Velocity:** The velocity $v \in \mathbb{R}$ between two subsequent cell centroids is computed using the Euclidean norm of the direction vector with $v = \|\vec{m}\|_2$.
- **Delta time:** For domain experts, the temporal difference of a cell migration between two time steps yields important insights about specific durations of cell phases within its cell cycle, for example. Thus, I define a delta time $\Delta t \in \mathbb{N}$:
$$\Delta t = t_{N+1} - t_1 = N. \tag{3.3}$$
Δt is defined by the temporal difference between the last and first time step of a cell migration. Note that $\Delta t = 1$ between two subsequent time steps t_i and t_{i+1} but N is usually greater than 1 because of longer cell migrations.

3.5 Cell Migration Definitions

In this thesis, a cell migration is defined by the choice of two visual representations. I distinguish between migrations that are displayed in a 2D cell lineage tree representation (*cell paths*) and migrations that are visualized in the 3D space (*cell trajectories*). Both designs are applied in the

similarity analysis. Here, the definition of cell paths is a modification of paths in graph theory with the following convention:

Definition 3.1 (Cell Path) *A cell path p is a sequence of ordered connected pairs (n_i, t_i):*

$$p = \{(n_1, t_1), \ldots, (n_{|p|}, t_{|p|})\},$$

with nodes n_i and time steps t_i. The length is $l_p = |p| - 1, (l_p > 0)$ where $|p|$ is the number of pairs. Here, the first and the last nodes are only allowed to be a root, leaf, or division node, i.e. a node of degree 1 or 3.

The left image in Figure 3.8 on page 29 shows some examples of cell paths. In order to distinguish different cell paths, three types are defined:

Definition 3.2 (Cell Root Path) *A cell root path p_r is any cell path that starts at the root node of a lineage tree. It ends in the first division node of the tree.*

Definition 3.3 (Cell Division Path) *A cell division path p_d is a cell path that starts and ends in a division node. The path must contain only movement nodes.*

Definition 3.4 (Cell Leaf Path) *A cell leaf path p_l is a cell path that starts in a division node and ends in a leaf node of the lineage tree.*

Note that if a node n describes a cell division, then two cell paths originate from this node. This pair is called a cell branch b_n:

Definition 3.5 (Cell Branch) *A branch b_n is defined by a starting node n with two successors, i.e. a cell division node, and by the length of its left path l_n and its right path r_n. These cell paths contain all nodes between two cell division nodes. A branch is called* symmetric *if the lengths of l_n and r_n are equal.*

The right lineage tree in Figure 3.8 on page 29 illustrate examples of all four types. A disadvantage of the cell lineage tree layout is that the centroid information of cell paths cannot be analyzed directly. Thus, I define a cell trajectory to represent this additional spatial property.

Definition 3.6 (Cell Trajectory) *A cell trajectory tr with length $|tr|$ is a cell path for which each cell is represented by its centroid $\vec{C}_i \in \mathbb{R}^3$ at time step t_i:*

$$tr = \{(\vec{C}_1, t_1), \ldots, (\vec{C}_{|tr|}, t_{|tr|})\}.$$

Analogous to the definition of the different types of cell paths above, a cell trajectory is also further distinguished into a *cell root trajectory*, a *cell division trajectory*, and a *cell leaf trajectory*. Figure 3.9 shows some examples of these cell trajectory types. In summary, I define a cell migration in this thesis as the following:

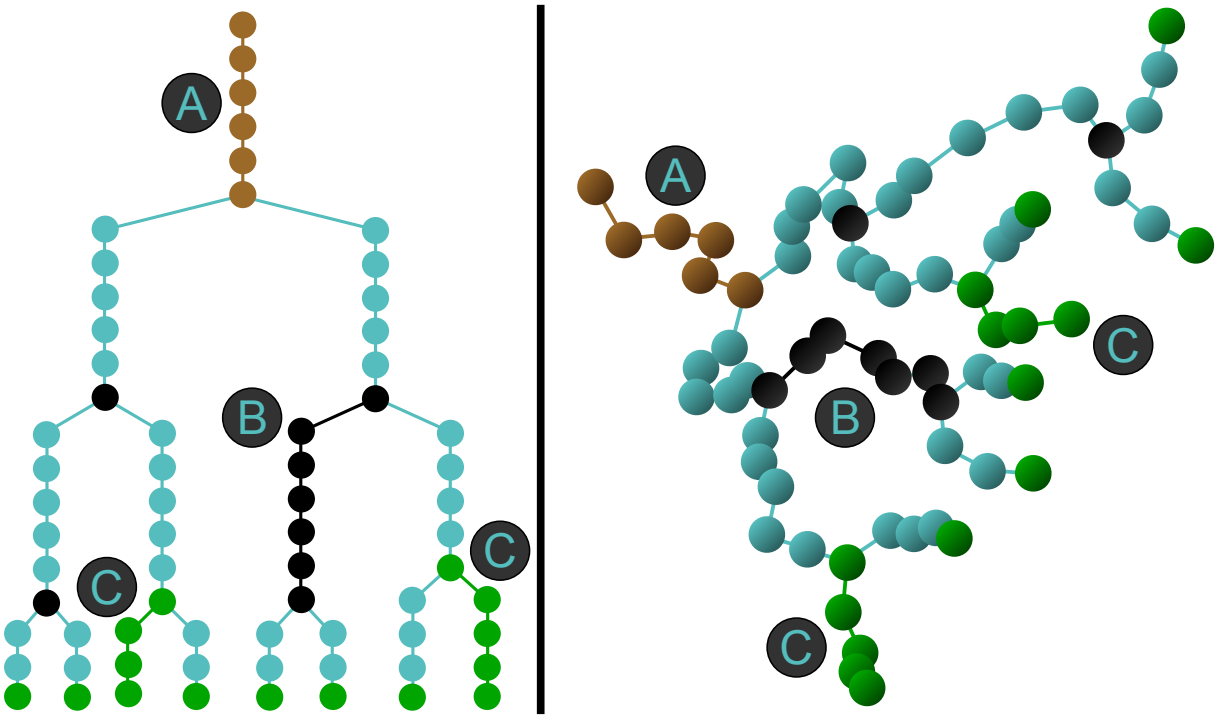


Figure 3.9: Examples of cell trajectory types. On the left side, a small lineage tree is displayed with one root cell path (**A**, brown), one division cell path (**B**, black), and two leaf cell paths (**C**, green). The right side shows the corresponding trajectories in 3D space.

Definition 3.7 (Cell Migration) *If it is visualized in a 2D cell lineage tree a cell migration is called a cell path and if the visual representation is given in the 3D space, a cell migration is called a cell trajectory.*

The separation between these two definitions is motivated by the visual representation of information. If shape, length, spread, and orientation of cell developments are of interest, then cell migrations should be visualized in the 3D space. The depiction of cell paths in 2D cell lineages is well-suited for the analysis and identification of structural properties of divisions (e.g. cell branches) and temporal patterns (delta time). The combined similarity analysis in both visual representations provide users more interpretation possibilities.

3.6 Summary

In this chapter, I described the data processing steps required to perform my visual similarity methods. I explained the data acquisition process of the different data types from the zebrafish embryo and the Arabidopsis seedlings together with their biological background and properties. The raw data is segmented and tracked manually or automatically in order to identify single nuclei and their cell tracks. While an automatic approach for the huge zebrafish data is required but results in erroneous cell lineages, the manual processing of the Arabidopsis data is tedious but yields higher quality of few lineage trees. These trees permit a noise-free similarity analysis among several data sets with a focus on division types. With respect to the data, I defined several features relevant for the analysis followed by a definition of cell migrations either depicted as cell paths or cell trajectories.

In the next chapter, I describe the first visual analysis method, a novel classification algorithm to determine the division types during the growth of lateral roots in plant data sets. The results of the division schemes are used later as additional features for the other two analysis methods.

Chapter 4

Automatic Classification of Cell Divisions in Plant Data Sets

"It is not so much that the cells make the plant;
it is rather that the plant makes the cells."
— Heinrich Anton de Bary,
quoted in "The New Statesman", 1920

In developmental biology, cell migrations and cell divisions fundamentally affect the generation of tissues and structures of any complex organism. A characteristic of developments of multicellular organisms is the robustness of shape formations. For example, plant cells are bounded by rigid cell walls precluding any cell migration. Thus, they solely rely on oriented divisions and cell growth in order to form the shape of their organs. While the plant embryonic development is highly stereotypical and only the basic blueprints of the adult organism are laid out, most of the plant organs are produced post-embryonically. One example of post embryonic organ formation in plants is the generation of new lateral roots from the main root. This development is caused by specific oriented division types forming a dome-like structure of the initial lateral root. Although there is a variation in the number of founder cells in the lateral root, the eventual formation of the dome structure is always similar in shape. Biologists are interested in the detailed visual analysis of this recent 3D+t developments and the detection of similar structures and division patterns of such growth at organ and cellular scales. Through this, both the investigation and comparison of single as well as multiple lateral root growth in several plants enable finding similar patterns. The cell-based analysis of such developments is a challenging task. As already emphasized in the introduction, cell developments in plants used to be only accessible in 2D which severely limited previous investigations such as the analysis of Malamy and Benfey [MB97], for example. Without consideration of the third dimension false interpretations about growth and cell divisions might occur. A processing in 3D is fundamental for a correct analysis but at the same time it complicates the visual analysis. The interactive visualization within an individual plant or the comparison between multiple ones permits new observations and interpretations.

In this chapter, I provide a visual analysis method to determine, classify and visualize cell division events in plant data sets. For this purpose, I introduce a novel automatic classification algorithm for determining three division types (*anticlinal, periclinal, radial*) during the lateral root growth. This classification is based on the generation of colored stereotyped *cell isosurfaces* for which the isovalue represents the number of periclinal divisions. These allow a visual and geometrical 3D comparison of such divisions among several data sets. An additional visualization illustrates the resulting division scheme in color-coded lineage trees with compact information about the division order and type. I apply the method to all five data sets of the Arabidopsis plants. The resulting division types are used as an additional feature for the similarity analysis methods explained later in chapters 5 and 6. I first explain the cellular organization of the lateral root using *cell files* (Section 4.1). In Section 4.2, the automatic classification algorithm based on *cell isosurfaces* is described. Afterwards, I examine its performance and parameter stability followed by application results (Section 4.3).

4.1 Cellular Organization of the Lateral Root

In order to specify a common coordinate system of cells in all lateral roots, *cell files* are introduced. The cell files are defined by the initial positions and radial development directions of founder cells observed in the *radial view* [MB97]. The first row in Figure 4.1 shows the founder

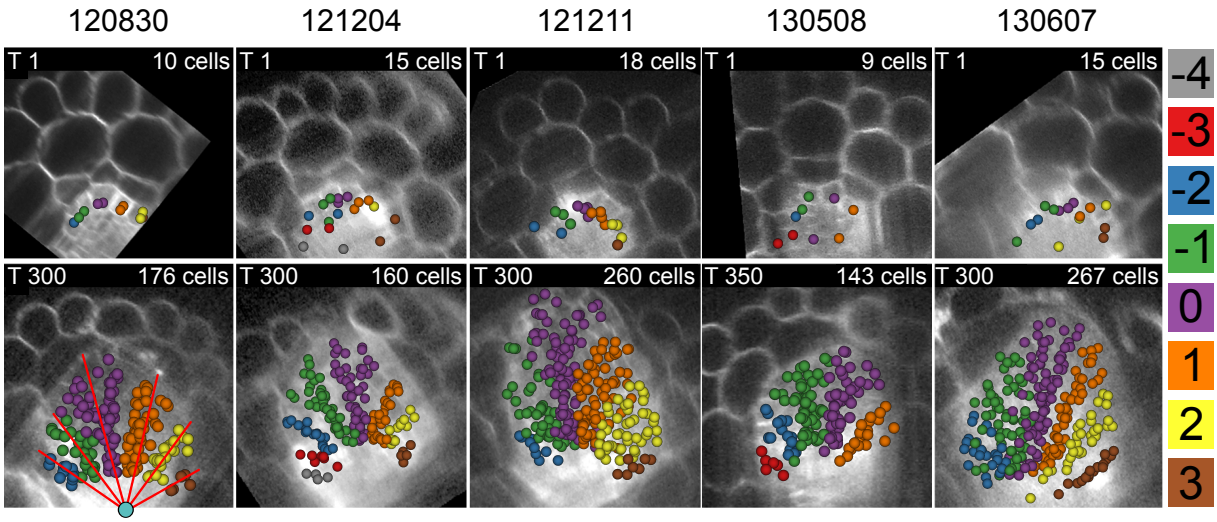


Figure 4.1: Colored cell file assignments on top of raw data MIPs for the Arabidopsis data sets: The first row shows the cell files on top of the MIPs of the raw data for the first time step. The second row illustrates the cell file development at the last time step of the data record. The specific time step and current number of cells are given at the top of each MIP. The labels at the right side indicate the cell file colors in such a way that the master cell file, i.e. the cell file that swells the most, is always assigned the value 0 with color violet. For the last time step of 120830, it is illustrated how the center of the root (cyan disk) is determined manually by intersecting all linear principal components of the cell files. Although the number of founder cells is different among the data sets, the similar structure of the dome is developed. Furthermore, the cells located in the master cell file are not necessarily situated at the center of the generated dome.

cells at the first time step for each plant data in radial view. The cells (spheres) are colored by their cell file memberships. For example, the data set 120830 has initially 10 cells that are

assigned to certain cell file values radially based on their positions. For 120830, these values range from -2 to 3. Note that a cell file can include cells from several lineage trees but a cell lineage is only assigned to exactly one cell file. For the classification algorithm, the position of the center of root is required. This position can be determined by the intersection point of the radial principal directions of each cell file (cyan disk in Figure 4.1 for the last time step of 120830). The *master cell file* is defined as the file that swells the most in comparison to all other cell files. The swell value is given by the Euclidean distance between the highest position of a cell within its file and the root position. Note that this does not necessarily mean that the master cell file features the highest number of cells in the file. In contrast to the *periphery* files, i. e. the outer cell files, the master cell file is of particular interest because it is mainly responsible for the forming of the dome-like structure of the lateral root. The cell files are labeled with fixed integers beginning from left to right with respect to the master cell file. These values are assigned in such a way that the master cell file has always a value of zero and cells colored in violet.

4.2 Automatic Classification of Division Types

The lateral root development in Arabidopsis is formed by a combination of cell divisions and cell growth while no cell movement is taking place (only nuclei displacements). In general, the division types are classified into three orientations depending on the division position with respect to the lateral root: *anticlinal, periclinal* and *radial* divisions. The colored arrows in Figure 4.3B show the different division directions that influence the growth and total size of the lateral root, i.e. both the number of cells as well as the volumetric size. I present a novel automatic classification algorithm to determine these cell division types. For this purpose, I introduce 3D *cell isosurfaces* and explain how they are generated.

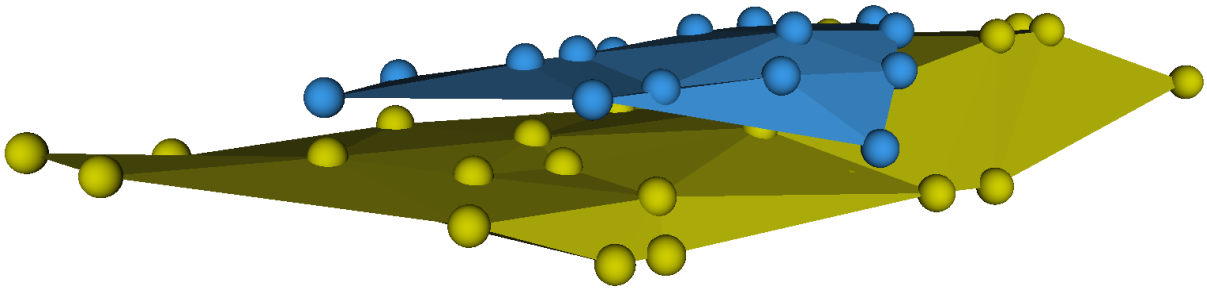


Figure 4.2: Example of two cell isosurfaces. The initial nuclei positions form a 3D triangulation which is called an isosurface with an isovalue of zero (yellow color). The nuclei positions may change over time and if a cell of this surface divides anticlinally or radially than the two daughter cells remain in this surface with equal isovalue of zero. However, a periclinal division results in the creation of a new isosurface (blue color) or the augmentation of an existing one with an increased isovalue by one.

4.2.1 Cell Isosurfaces

In this thesis, a cell isosurface is a 3D triangulation of a set of cells that share the same number of periclinal divisions. Its color-coded isovalue represents this number. This means that existing isosurfaces change in each time step and new isosurfaces are generated or augmented if cells

divide periclinally. Figure 4.2 illustrates an example of two isosurfaces. They allow two biological interpretations: First, the colored isosurfaces depict the number of periclinal divisions and allow a direct investigation of these divisions that mainly contribute to the height of the dome structure. Second, the isosurfaces serve as a visualization of the spatial development of

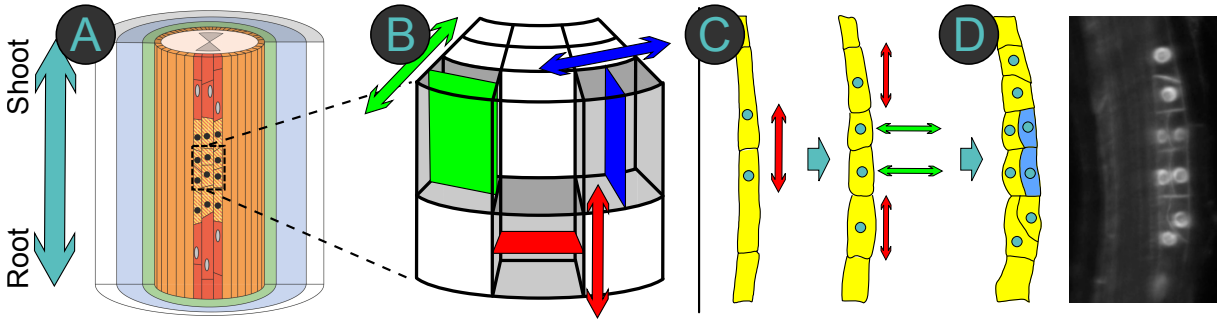


Figure 4.3: Division types in the lateral root of Arabidopsis. (**A**) shows a section of the Arabidopsis root. The enhanced part within the section illustrates the different division types (**B**): *anticlinal* (red arrow), *periclinal* (green arrow) and *radial* (blue arrow). (**C**) illustrates the behavior of anticlinal and periclinal divisions for cells in the master cell file. A periclinal division results in a new isosurface (**D**, orange cell tissues) while for an anticlinal division, the cells remain in the same isosurface. Note the change of *cell walls* in the raw microscopy image for each cell indicating the new isosurface and arrangement of cells.

this division type. Thus, users can observe where periclinal divisions occur and analyze their contributions. The latter one can also be expressed quantitatively by the volume or the number of cells of the isosurfaces. These isosurfaces are used to generate vertex/surface normals that are compared with division orientations. This comparison then yield a classification of division types explained below.

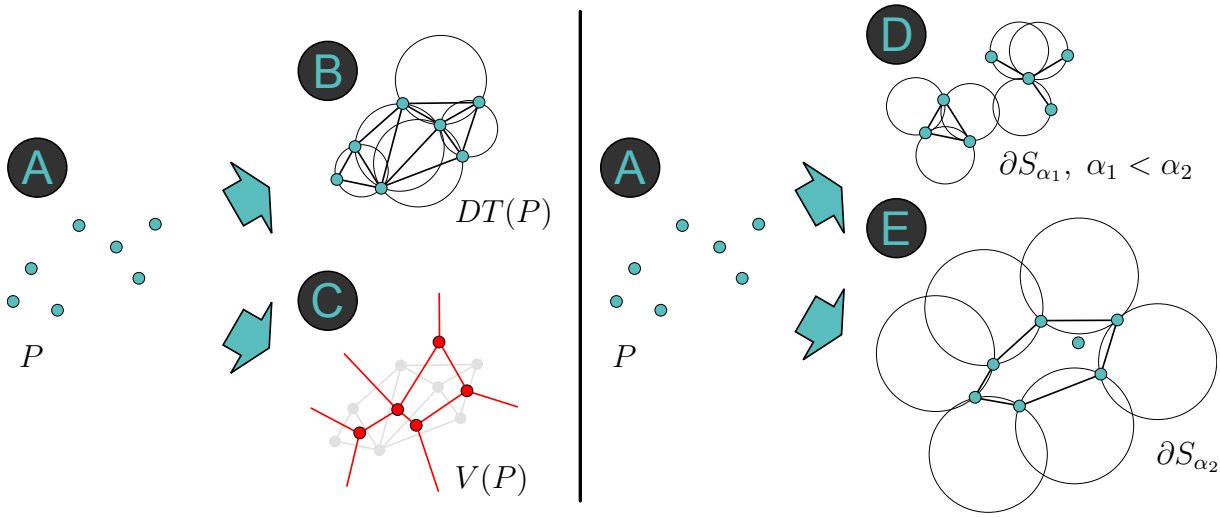


Figure 4.4: 2D examples of Delaunay triangulation and α-shapes. The left image shows the generated Delaunay triangulation (**B**) and the corresponding Voronoi diagram (**C**) based on the point set P (**A**). The right image gives two examples of the resulting α-shapes when using two different α parameters α_1 (**D**) and α_2 (**E**) with $\alpha_1 < \alpha_2$.

For generating the surface of a set of cells P with three-dimensional points, the most common method is the *Delaunay triangulation* [Del34] DT. This triangulation has the property that no point in P is inside the circumsphere of any tetrahedron in $DT(P)$ (Figure 4.4B for an example). Another property of the Delaunay triangulation is that the union of all simplexes in DT is the convex hull of all cell positions. A simplex is a generalized description for an n-dimensional polytope, i.e. a polygon of arbitrary dimensions. For example, 0-simplex is a point, 1-simplex is a line, 2-simplex is a triangle, and 3-simplex is a tetrahedron. A *dual graph* of a plane graph G is a graph that has a vertex for each corresponding facet of G and an edge connecting two adjacent facets for each edge in G. Figure 4.4C shows the dual graph of the Delaunay triangulation: the *Voronoi diagram* [Vor08]. However, as illustrated in Figure 4.5A, the surface, or more precisely the curvature of the convex hull is an inappropriate approximation of the evolving dome-like structure of the primordium according to domain experts. This will lead to incorrect results in the subsequent comparison of surface/vertex normals and division directions. Because of these reasons, I choose an α-*shape* [BB97] for generating the surface (Figure 4.5B). This shape is a family of piecewise linear simple curves associated with the shape of a point set P based on the Delaunay triangulation. More precisely, the α-complex of P is a subcomplex of this triangulation that contains α-exposed k-simplexes ($0 \leq k \leq 3$). A simplex

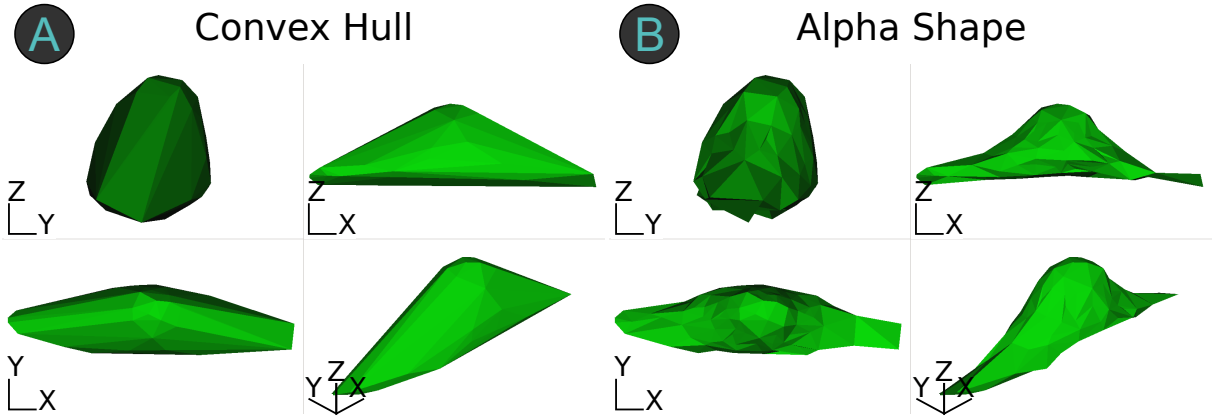


Figure 4.5: Convex hull and alpha shape of the data set 130607. Biologically motivated, four different viewing types (*Radial (y, z), side (x, z), top (x, y), and 3D*) are used to represent the lateral root. The images show different surfaces of the last time step of data 130607. (**A**) illustrates the convex hull of the primordium based on the delaunay triangulation while (**B**) shows the α-shape. α is selected in such a way that only one connected component is generated. The α-shape is better suited as an approximation of the lateral root than the Delaunay triangulation.

is α-exposed if there is a sphere with the squared radius of α in which all points of the simplex lie at its boundary and do not contain any other points of P:

Definition 4.1 *Let P be the set of points in general position, i.e. they do not satisfy a special or coincidental relation to each other, and $T \subset P$ with $|T| = k + 1 \leq 4$. Let further be $M_T \in \mathbb{R}^k$ a polytope denoting the convex hull of T. Then M_T is a k-simplex.*

Let S_α be the α-shape. Then the boundary ∂S_α consists of all k-simplexes of P that are α-exposed:

$$\partial S_\alpha = \{M_T \ : T \subset P, |T| \leq 3 \text{ and } M_T \text{ is } \alpha\text{-exposed}\}. \tag{4.1}$$

Then, S_α is the resulting triangulation. Figures 4.4D and E illustrate two examples of α-shapes for two different alpha values $\alpha_1 < \alpha_2$. Note that an α-complex can be a non-connected polytope (Figure 4.4D) but the α value with $0 \leq \alpha \leq \infty$ can be selected optimally in such a way that only one connected component is generated. Further note that the α-shape degenerates to the point set P if $\alpha \to 0$. If $\alpha \to \infty$ then the α-shape is the convex hull (Figure 4.4E). The α value in Figure 4.5B is chosen in such a way that only one connected component is created. For all α-shapes in the algorithm, I will use appropriate α values that always result in exactly one connected component. This is realized by a binary search on the α values and has a time complexity of $\mathcal{O}(|P| \log |P|)$ with $|P|$ denoting the number of points.

In order to determine the division types, I generate and update the cell isosurfaces for each time step because the shape of the primordium varies over time. Based on these surfaces, the algorithm is able to classify the different types of divisions using information of surface and vertex normals. Each division direction is then compared to this surface or vertex normal by computing the angle between them. This angle then designates the kind of occurring division type.

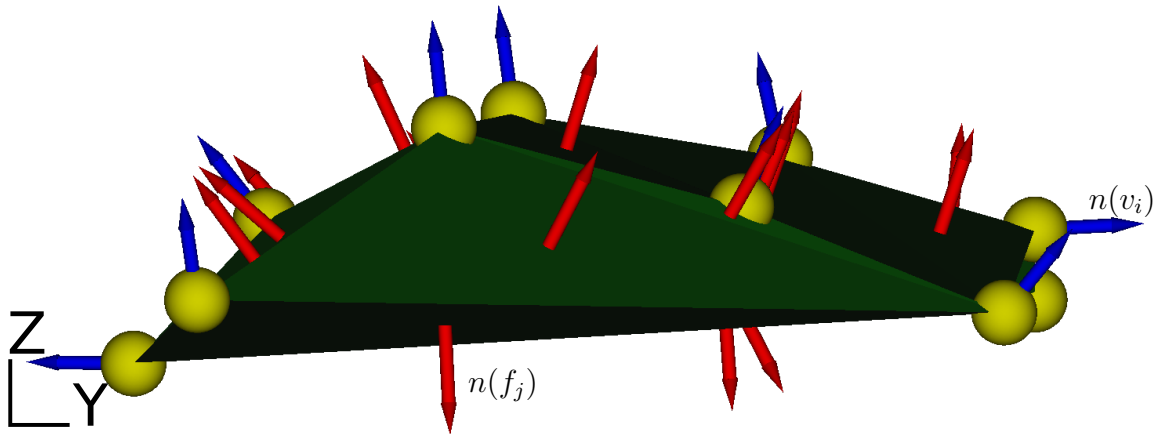


Figure 4.6: Visualization of isosurface and vertex normals. The image shows an example of an α-shape of the primordium in radial view. The red arrows indicate the surface normals pointing outwards and the blue ones show the vertex normals.

4.2.2 Classification Algorithm

In each time step and for each dividing cell, the algorithm performs two angle comparisons between division orientations and vertex/isosurface normals. In the first angle comparison, the algorithm performs a fundamental check for an anticlinal or periclinal division. Note that for the former division, the cells remain in the same isosurface while for the latter division, a new isosurface is generated or augmented by a cell. However, at this time of the algorithm, the anticlinal division could also be a radial one (for both types, the cells remain in the isosurface). This is caused by the fact that a single angle check in 3D allows no unique separation between an anticlinal or radial division. For this reason, I apply a second angle comparison to distinguish between an anticlinal and radial division. For both angle checks, user-selected angle thresholds (δ and ρ) are introduced that mainly influence the classified type of division. I apply a stability check for these parameters in Section 4.2.4.

For each isosurface, the surface and vertex normals are determined (Figure 4.6). Let v_i be a vertex of the cell set forming the current α-shape and $n(f_j)$ the surface normal of the facet (triangle) f_j pointing outwards. The vertex normal $n(v_i)$ of vertex v_i is computed as the mean of all adjacent facet normals:

$$n(v_i) = \frac{1}{|F_i|} \sum_{j \in F_i} n(f_j). \tag{4.2}$$

F_i represents the index set of all adjacent facets to vertex v_i and $|F_i|$ denotes the number of adjacent facets. Note that only cells that belong to the boundary ∂S_α of the α-shape have a vertex normal. For interior cells, the normal is determined by the closest facet normal in terms of the closest Euclidean distance between the cell position and a facet. A vertex or surface normal, from now on denoted as n, is compared with the division direction. In the following, the value l will refer to both the isosurface and its isovalue. The algorithm is designed based on the following biologically motivated constraints:

- When a dividing cell at time step t with isovalue l performs a periclinal division, then only one of the daughter cells is assigned a new isovalue $l + 1$. The other daughter cell remains in the previous isosurface l.
- For each division, it is assumed that the cells divide almost collinearly. This means that the division orientation is given by the vector between the two daughter cells. This does not necessarily mean that both daughter cells at $t + 1$ and the dividing cell at t lie on the same line.

The algorithm is designed in such a way that the division types are determined for the current time step t but the isovalues are set for the next time step $t + 1$. This is required because the algorithm performs angle checks based on the position of a dividing cell at time step t and the direction vector of its subsequent daughter cells. With these constraints the automatic classification algorithm is realized as follows (Figure 4.7):

Shape generation: For each time step t, the α-shape of the cell point set P_t that share the same isovalue l is generated (line 6). If $t = t_0$ and the first recorded time step, then each cell is assigned an initial isovalue of 0 (yellow cells and shape in Figure 4.8A or Figure 4.9A).

Cell division check: For each cell with position p_k at time step t, the corresponding isovalue l is determined (line 8) and checked if it is a dividing cell. If so then the positions of its daughter cells are stored. Afterwards, the next step of the normal determination is applied. If the cell is not dividing, then its successor (based on tracking) in time step $t + 1$ is assigned the same isovalue l as in time step t (line 33). If the cell has no successor at all then nothing is done and the next cell is checked.

Normal determination: In line 12, the normal is computed. If an isosurface l only consists of one or two cells, the normal for the angle check is given by their vertex normals. For one cell, this is the normalized direction vector pointing from the center c to the current cell position p_k:

$$n = \frac{p_k - c}{|p_k - c|}. \tag{4.3}$$

Note that the center c is defined manually by the intersection of the principal components of

Input : Point set of cells P_t for all time steps $t \in [1, T]$, center of root c, angle thresholds δ and ρ, view rotation matrix M_{rot}, tracking information of cells.
Output: Classified division types.

```
1  D(i, j) : map with i ∈ [1..T − 1], j ∈ ℕ of {anticlinal, periclinal, radial};
2  begin
3      numSurfaces ← 1;
4      for t ← 1 to T − 1 do
5          for l ← 0 to numSurfaces−1 do
6              generateAlphaShape(P_t);
7          for k ← 1 to |P_t| do
8              l ← determineIsoValue(p_k);
9              if dividingCell(p_k) == true then
10                 d_1 ← daughterCell(1, p_k);
11                 d_2 ← daughterCell(2, p_k);
12                 n ← normalDetermination(p_k, c, l);
13                 dir ← determineDivisionDirection(d_1, d_2);
14                 β_1 ← computeAngle(dir, n);
15                 β_2 ← computeAngle(−dir, n);
16                 divType ← divisionAngleCheck(β_1, β_2, n, δ);
17                 if divType == periclinal then
18                     if β_1 < β_2 then
19                         assignIsoValue(d_1, l + 1);
20                         assignIsoValue(d_2, l);
21                     else
22                         assignIsoValue(d_1, l);
23                         assignIsoValue(d_2, l + 1);
24                     if l + 1 > numSurfaces then
25                         numSurfaces ← numSurfaces+1;
26                 else
27                     assignIsoValue(d_1, l);
28                     assignIsoValue(d_2, l);
29                     divType ← radialDivisionCheck(dir, M_rot, ρ);
30                 D(t, getID(p_k)) ← divType;
31             else
32                 d ← daughterCell(1, p_k);
33                 assignIsoValue(d, l);
```

Figure 4.7: Automatic classification algorithm for determining division types.

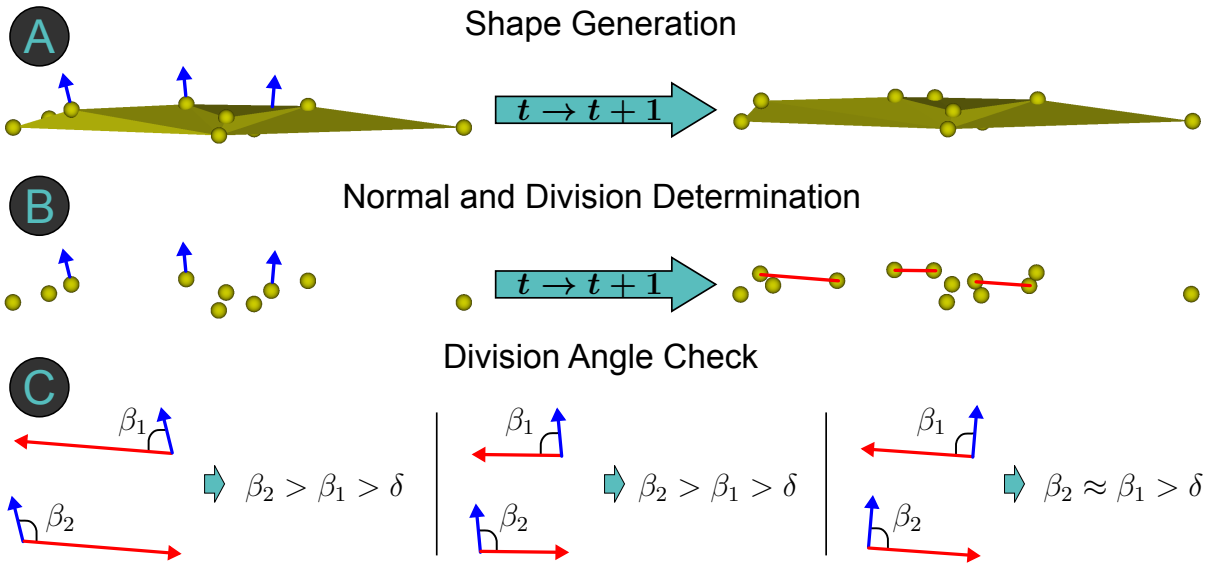


Figure 4.8: Determining the division type that results in anticlinal divisions. In (**A**), the shape is generated for all cells within the same isosurface. Blue arrows indicate the normals of those cells that divide in time step $t+1$. The divisions are highlighted by red lines in (**B**, right). For both possible division directions, the angles between the vertex normals are computed and compared with a user-selected angle threshold δ (here $\delta = 45$). In (**C**), all three divisions result in an anticlinal division due to the larger angles β_1 and β_2 compared to δ.

the cell files. For two cells, the mean position $\overline{p} = \frac{p_1+p_2}{2}$ is computed and afterwards the same computation as above is used to get the normalized direction vector pointing from the center to $\overline{p}$:

$$n = \frac{\overline{p} - c}{|\overline{p} - c|}. \tag{4.4}$$

For three cells, I create a triangle and select the facet normal pointing away from the center c. With at least four cells the isosurface l is generated using an α-shape. An additional check is required to determine if the current cell position p_k belongs to the boundary or if it is located in the interior of the surface. In the former case, n is given by the vertex normal of the dividing cell at time step t (blue arrows in Figure 4.8A,B or Figure 4.9A,B) while in the latter case, n is selected by the normal of the isosurface nearest to the cell position p_k.

Division angle check: In the next step, it is determined if the division is an anticlinal/radial or a periclinal division. The division direction is given by the vector between the two daughter cells at time step $t + 1$ (line 13 and red arrows in Figure 4.8C or Figure 4.9C). Let $\delta = 45$ be the first selected angle threshold. It has to be decided which of the two possible directions of the division should be considered for the comparison with the normal n. Because in the case of a periclinal division, this defines which of the two daughter cells is assigned to the next isosurface. The two angles β_1 and β_2 are computed for both possible directions (line 15) and the daughter cell associated with the smaller angle is assigned a new isovalue increased by one. Let β_1 be the smaller one. If $\beta_1 > \delta$, the division is an anticlinal/radial (all three divisions in Figure 4.8) and both daughter cells are assigned the same isovalue l. Otherwise, if $\beta_1 \leq \delta$ the division is a periclinal one and a new isosurface is created (blue cell in Figure 4.9).

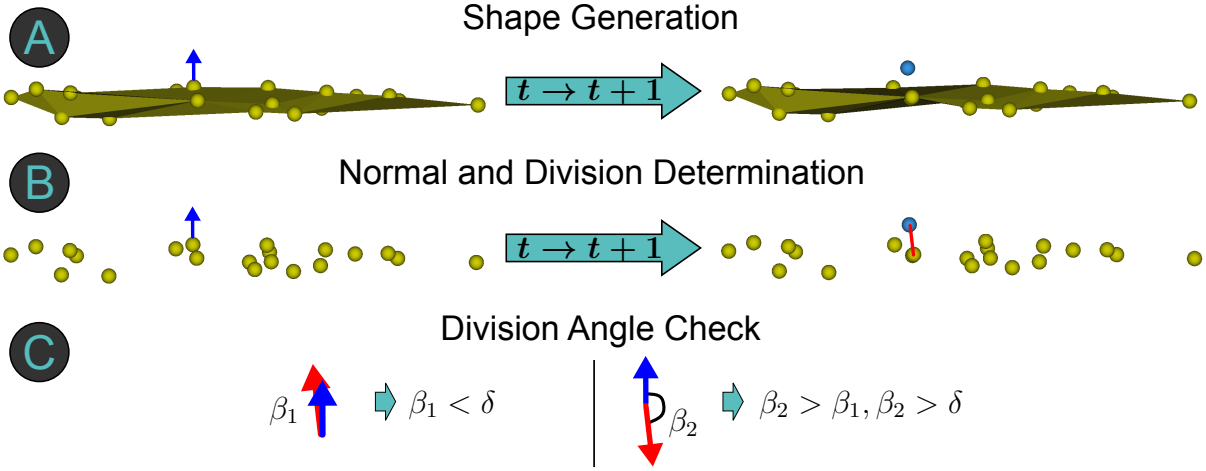


Figure 4.9: Determining the division type that results in periclinal divisions. For all cells within an isosurface, the α-shape is generated (**A**). The blue arrow in (**B**) indicates the normal of a cell that divides at time step $t+1$. The red line on the right image in (**B**) denotes the division direction. This is checked for both possible directions. In (**C**), $\beta_1 < \beta_2$ and $\beta_1 < \delta = 45$. This results in a periclinal division and the "upper" daughter cell that corresponds to the angle β_1 is assigned to a new isosurface colored in blue.

Radial division check: After the first division check, it is still unclear if an anticlinal division could not also be a radial one. This is realized in line 29. Each data set is rotated by a rotation matrix M_{rot} such that the four viewing types (radial, side, top, and 3D view) of the lateral root in Figure 4.5 are satisfied. With respect to the side view, a perfect radial division would correspond to the normal n_s of the x-z plane with $n_s = (0, -1, 0)^T$. Thus, the direction vector dir between the two daughter cell positions d_1 and d_2 is chosen in such a way that the vector always points from the larger rotated y value to the smaller one:

$$\text{dir}(d_1, d_2) = \begin{cases} d_1 - d_2 & \text{if } d_1{}'_y < d_2{}'_y \\ d_2 - d_1 & \text{else} \end{cases} \tag{4.5}$$

with $d_1{}' = M_{rot} \cdot d_1$ and $d_2{}' = M_{rot} \cdot d_2$, respectively. Let $n_{tr} = M_{rot}^{-1} \cdot n_s$, then the division type of dir is determined by using the second user-selected angle threshold ρ:

$$\text{divType}(\text{dir}) = \begin{cases} \text{radial} & \text{if } \sphericalangle(n_{tr}, \text{dir}) < \rho \\ \text{anticlinal} & \text{else.} \end{cases} \tag{4.6}$$

Afterwards, a map D with values of type `enum` for anticlinal, periclinal, and radial stores the division types for each dividing cell (line 30). A pair of the current time step t and the cell ID is an unique identifier and used as a key for the map. After processing all cells at all time steps except the last one, all division types are classified.

4.2.3 Performance Analysis

The algorithm is tested on a standard desktop computer with an Intel Core i7, 3.20 GHz, 12 GB of memory and an NVidia GTX 480. I use the *Computational Geometry Algorithms Library (CGAL)* (https://www.cgal.org/) to generate the isosurfaces realized by α-shapes. Table 4.1 lists

Data set	Time steps	Divisions	Cells at end	Computation times [s]	
				Isosurface generation	Angle checks
120830	300	166	176	1.05	0.12
121204	300	156	160	0.98	0.12
121211	300	242	260	1.29	0.20
130508	350	134	143	0.91	0.09
130607	300	252	267	1.41	0.25

Table 4.1: Computation times for automatic classification of division type in Arabidopsis data sets.

the computation times in seconds for the isosurface generation and the pair of angle checks of the automatic algorithm. In total, the algorithm takes less than a second to finish the classification. In contrast to the angle checks, the shape generation needs more time by a factor of approximately $6-10$. Note that in the inner for-loop over all cells in the current time step, only dividing cells are further processed. Thus, the total number of divisions in the data set is identical to the number of the pair of angle checks. The time complexity is at most $\mathcal{O}(S(T-1)|P_S|^2 + |D|)$ with S as the maximal number of generated isosurfaces, T the number of time steps, $|P_S|$ the number of points of the corresponding surface, and $|D|$ the number of total divisions in a data set. The term $S(T-1)|P_S|^2$ refers to the required time of the isosurface generation. This means that the more cells are considered the longer the surface generation takes. This can be observed for the data sets 121211 and 130607 that both have more than 260 cells at the end. A result in Section 4.3 is that the number of maximal generated isosurfaces are for all data sets only $S = 4$ so this is a constant. While for the angle checks, only simple computations are required such as using the cosine to compute the angle between two vectors, the worst time complexity for generating the α-shapes is quadratic in the number of points $|P|$. However, the α-shapes are based on the Delaunay triangulation for which the upper bound is $\mathcal{O}(|P|^2)$ which is usually not reached. Thus a smaller time complexity of $\mathcal{O}(|P|(\log|P|)^2)$ is more common [EM94a]. This also includes the time complexity of $\mathcal{O}(|P|\log|P|)$ for the binary search finding the optimal α value in such a way that only one connected component is generated. Another explanation for the large difference in the computation times between surface generation and angle check is the fact that the α-shapes are created for all cells and for all time steps while the angle check is only performed for dividing cells. Also note that in earlier time steps, the surface generation is realized much faster with a few cells in contrast to later time steps with increasing and more than hundreds of cells. Even though the computation times are low, I store the results of the resulting division types as well as the isosurfaces on the disk for faster reprocessing.

The data acquisition process of the Arabidopsis plants results in recorded 3D+t volumes of several hundreds of GiBs. However, the algorithm only needs the extracted information of the manual segmentation and tracking results. This means that the three-dimensional position information of all cells over all time steps as well as their tracking information realized by pointers in a binary tree structure are required in the algorithm. Consequently, in the worst case, the space complexity of the algorithm is $\mathcal{O}(Q + |D|)$ with $Q = \sum_{t=1}^{T} |P_t|$ as the sum of all cells over all time steps and $|D|$ as the size of the map in which the results are stored. This

means that the space usage increases linearly in the number of total cells for all time steps plus the number of total divisions. For example, the space usage of the algorithm is smaller than one MiB (64 bits double precision) for the data set 120830.

4.2.4 Parameter Stability

The choice of the two user-selected angle thresholds δ and ρ fundamentally influences the results of the classification algorithm. In order to examine the stability of these parameters and how they influence the results, different distributions of division types for slightly changed threshold values are investigated. For this purpose, only the data set 120830 is considered. Figure 4.10 lists four line charts of the different division distributions when changing the thresholds in steps of 5 degrees in a total range of $[0, 100]$. A degree value higher than 100 does not further change the distribution in this data set. Four cases of varying thresholds are distinguished: δ changes with fixed ρ (Figure 4.10A), ρ changes with fixed δ (Figure 4.10B), δ and ρ change with identical values (Figure 4.10C), and δ and ρ change in such a way that their sum is always 100 (Figure 4.10D). These variations allow an investigation of possible threshold settings. In a perfect model of the lateral root, the angle difference between a pair of an anticlinal, periclinal, or radial division would be exactly 90 degrees. For this reason, a fixed value of 45 is chosen

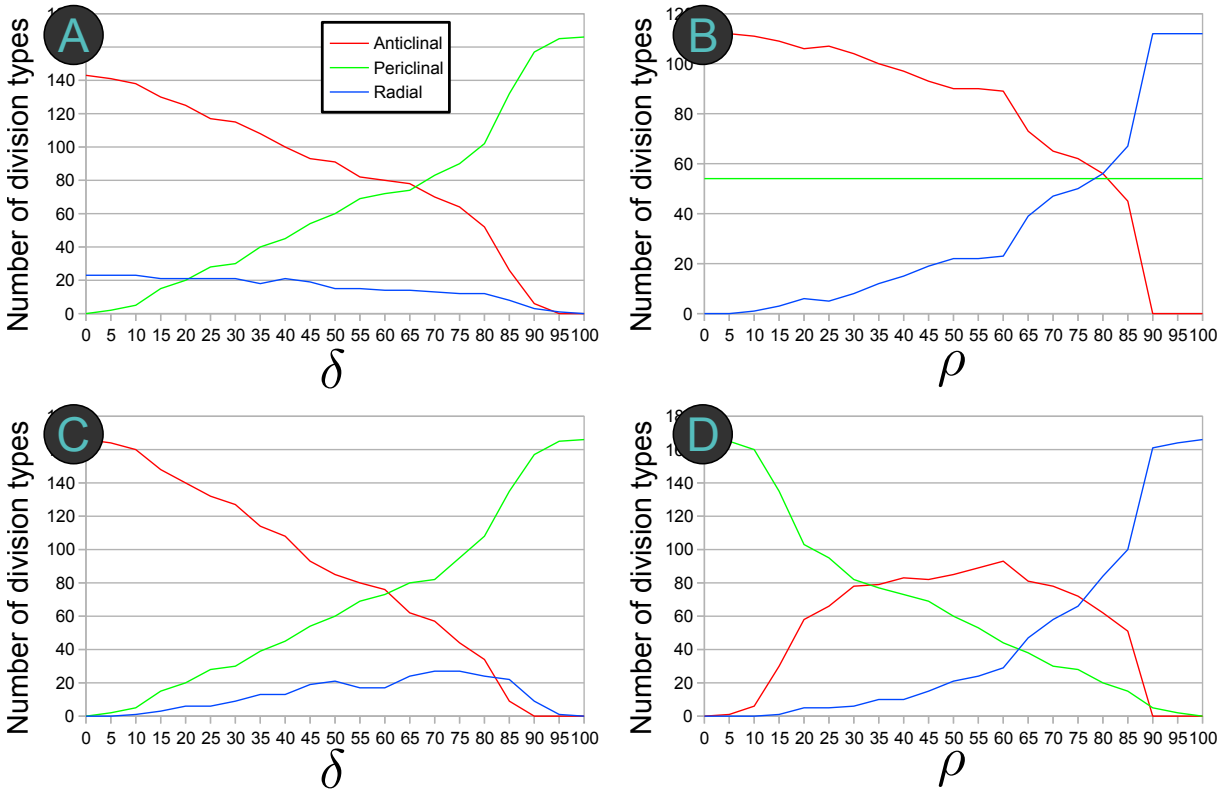


Figure 4.10: Stability check for angle thresholds of classification algorithm. All line charts show the division distributions by three curves representing the three types in the data set 120830 with 166 divisions in total. For the pair of thresholds (δ, ρ), the following sequences are represented in each chart: (**A**) $(0, 45), (5, 45), \ldots$, (**B**) $(45, 0), (45, 5), \ldots$, (**C**) $(0, 0), (5, 5), \ldots$, (**D**) $(100, 0), (95, 5), \ldots$. Note that the x-axis in (**D**) only shows the value for ρ, although both thresholds are changed and have different values.

in two cases in such a way that this parameter does not prefer a certain division type. A first observation is that the number of division types varies even for small changes of the angle values. Another common property in all four charts is that for angle values higher than 80 degrees a pair of curves has always an abrupt change in its behavior (red and green in Figure 4.10A and Figure 4.10C, red and blue in Figure 4.10B, blue and green in Figure 4.10D). This means that for such high thresholds, the data sets contain only a few division events that satisfy such an extreme behavior. Furthermore, based on the design of the algorithm, there is a contrast between the division types. This means that if a division is not an anticlinal one then it has to be a periclinal division. Consequently, if a division is classified as an anticlinal one, it could also be a radial division but not a periclinal one. This dependency explains the symmetric behavior for such pair of curves. Another observation is that the curves in Figure 4.10A (only changing δ) and Figure 4.10C (changing both thresholds simultaneously) are similar which means that the influence of the δ parameter is stronger than the influence of the ρ value. In Figure 4.10B, the periclinal curve is constant for all ρ thresholds because this value affects only the choice between an anticlinal or radial division. The result in Figure 4.10D shows an ambivalent behavior. For $\rho < 60$, more divisions are classified as anticlinal and radial while for larger angle values, the anticlinal divisions tend to zero because larger ρ values yield more radial divisions.

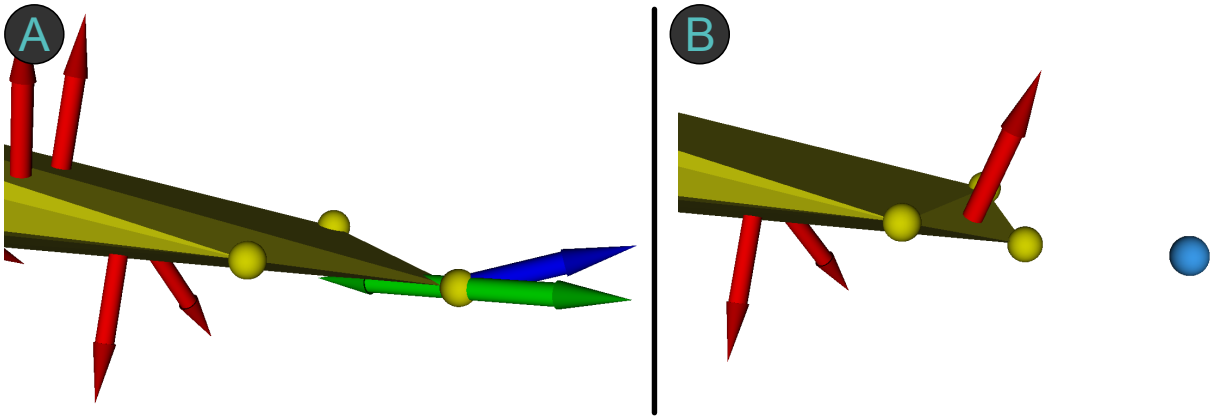


Figure 4.11: Example for an implausible isosurface assignment of a cell. The images show two subsequent time steps in which a boundary cell at time step t (**A**) is assigned to the next isosurface at time step $t + 1$ (**B**). However, although the algorithm works correctly, the assignment is biologically incorrect. Based on the normals of the four adjacent surfaces (red arrows) the vertex normal (blue arrow) of the dividing cell is determined. Its division directions are indicated by green arrows. In order to change the resulting periclinal division into an anticlinal one, a manual reassignment of isovalue information is possible.

Note that values of $\delta = 45, \rho = 45$ correspond to an equally weighted distribution such that no division type is preferred. For the data sets under investigation (see Section 4.3), I choose a slight variation with parameters of $\delta = 50, \rho = 45$ as it yields more biologically plausible results according to domain experts. For parameters with a larger divergence (> 10), the isosurfaces become more chaotic with more implausible division types. But even for this parameter setting, few divisions are assigned to a cell isosurface by mistake. This can occur for cells located at the periphery, for example. They have a vertex normal for which the division angle check yields a result that is biologically implausible. Figure 4.11 demonstrates such a case. The yellow boundary cell in Figure 4.11A features a vertex normal (blue arrow) which is determined by

its four adjacent surface normals (red arrows). In the next time step in Figure 4.11B, the cell divides (green arrows) and the algorithm identifies this division as a periclinal one. However, the analysis in 3D reveals that this division should be classified as an anticlinal one. Because of such possible errors, a manual editing of the assignments to an isosurface is permitted. In each Arabidopsis data set, on average 5% of the cell divisions are assigned to isosurfaces implausibly. These errors are corrected manually and saved for further analysis steps.

As a result, small changes of the degrees also result in small changes of the number of division types. This behavior is caused by the fact that sometimes the cells divide in such a way that their division types cannot be identified without any doubt even by domain experts. For such cases, the manual editing can be used to select the assignment that seems most plausible.

Visual Analysis

In this subsection, I briefly present the visual analysis techniques for investigating the resulting division types and isosurfaces.

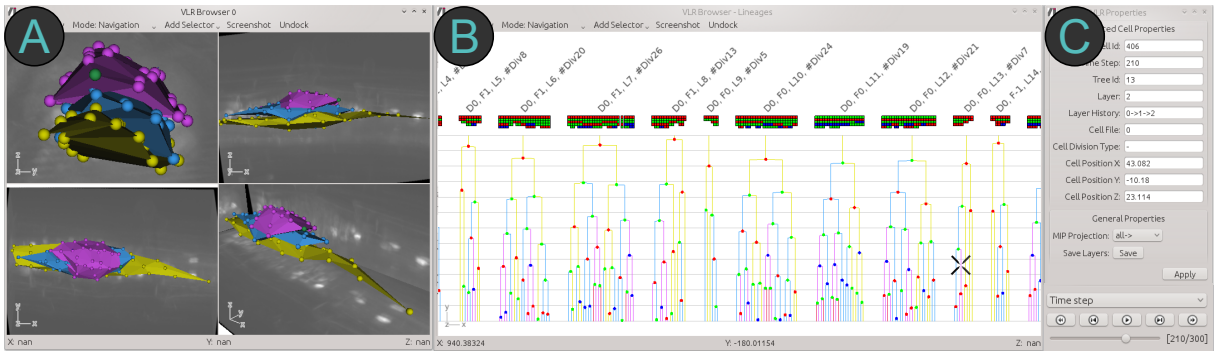


Figure 4.12: Visual analysis of division types in Arabidopsis data. The cells and isosurfaces are presented in a 2D/3D composite view (**A**) while a set of color-coded cell lineages illustrate the cell developments in 2D (**B**). All visualizations are interlinked and interactive. Detailed properties of selected cells are given on demand (**C**).

Cells and isosurfaces (Figure 4.12A) In a 2D/3D view, the cells are rendered by spheres with constant radii that are colored according to their isosurface assignments. The surfaces are represented as α-shapes in the same color scheme. Additionally, I include the maximum intensity projections of the current time step and data set in the background of each view. Through this, biologists can compare the relative location of cells to their cell walls in the raw microscopy data and the virtual representation of the cell. Note that the cell positions between two subsequent cell divisions are interpolated linearly. Thus, the positions match for dividing cells and their daughter cells but not necessarily for the displaced nuclei. The manual editing of isovalues is realized in this representation.

Lineage trees (Figure 4.12B) A 2D lineage tree is used to represent the isosurface memberships of migrating nuclei as well as the division types of dividing cells. This information is encoded in the tree using different colors for lines (nuclei displacements) and circles (dividing cells). Figure 4.13B shows an example of such a cell lineage. On top of each lineage structure, I create a compact grid of small colored squares to allow a faster investigation of the division

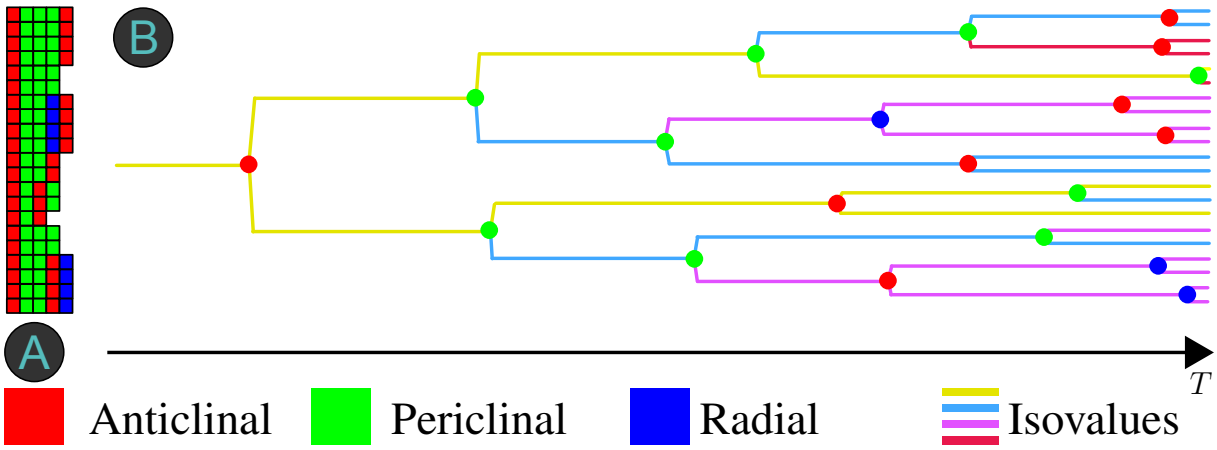


Figure 4.13: Visualization of ordered division sequences in lineage trees. I color the lines of the lineage tree in (**B**) based on the results of the isosurface assignment in yellow for isovalue 0, light-blue for isovalue 1, magenta for isovalue 2 and red for isovalue 3. The cell divisions are colored according to an anticlinal (red), periclinal (green), and radial (blue) division. In order to better compare these division results to other trees, I add a compact grid of squares colored according to the order and occurrence of division types (**A**). Each row, read from left to right, indicates the ordered appearance of divisions between the root and each leaf of the tree.

order (Figure 4.13A). In this rotated view, the number of rows of the grid denotes the number of leaves of the tree while the number of columns indicates the number of precursor divisions for the cell located at a leaf. For example, consider the last row with five squares colored from left to right by red, green, green, red and blue. This order of divisions is associated with the lowermost leaf of the tree developing from one anticlinal division, two periclinal ones, again one anticlinal and finally one radial division. By the use of this compact visualization, users can immediately compare the division orders among several trees.

4.3 Application Results and Data Comparison

In order to demonstrate the usefulness of the algorithm it is applied to all five Arabidopsis data sets using angle thresholds of $\delta = 50$ and $\rho = 45$. Table 4.2 lists properties of the data sets and results of the generated division types. For each plant data, the algorithm generates at most four isosurfaces at the last time step. Consequently, only four different colors (yellow, blue, magenta, red) are required to distinguish between them. This result can be explained by two observations: Although the same growth event of the lateral root is captured in all raw data, the record beginnings and endings are differing. For example, in 121211, the first anticlinal divisions are missing, which causes the peak of the number of cells (18) at the beginning. Thus, the lateral root is in a later stage of development for which a fourth isosurface is identified. Second, the cell cycle durations vary significantly among the data. This yields a high variance of occurring cell divisions and therefore the faster or slower generation of isosurfaces. 121211 and 130607, for example, both feature a high amount of divisions among all time steps. Consequently, these two data sets also develop the furthest into four isosurfaces in contrast to the other data sets.

In order to find similarities in the division behaviors, the lineage trees located in the master

Data set	Start time	End time	Divisions	Division types			Cells at start	Cells at end	Isovalue
				A	P	R			
120830	1	300	166	72	69	25	10	176	3
121204	1	300	156	83	54	19	15	160	3
121211	1	300	242	86	107	49	18	260	4
130508	1	350	134	51	56	27	9	143	3
130607	1	300	252	110	94	48	15	267	4
Sample standard deviation	0	22.36	53.42	21.50	23.55	13.92	3.78	58.11	0.55

Table 4.2: Division properties of Arabidopsis data sets considering all time steps. The classification algorithm for all recorded time steps yields high sample standard deviation values for the different division types because of the diversity of the temporal development of the plant data. This variety is also illustrated in the total number of divisions as well as the number of cells at start and end.

cell file are analyzed. Note that the master cell file contributes most to the complete tissue of the lateral root. Thus the cells in this file are of high interest for domain experts. Also note that the isovalue of an isosurface reflects the number of periclinal divisions of a cell. Figure 4.14 illustrates the color-coded lineage trees and division schemes of all plants. One observation is that all trees start with an anticlinal division except for 121211 because the record of this data set starts at a later development stage. Furthermore, the trees often feature an alternating order of their divisions. This order switches back and forth between an anticlinal/radial and a periclinal division. In other words, the lateral root switches between a growth in width and girth and a growth in height. Note that the plant data is varying according to the record time and that sometimes division types are ambiguous even after manual examination. Thus, two subsequent anticlinal divisions at the beginning for 121204 and 130508, respectively, are identified as well as some outliers in the alternating division order. The last row in Table 4.2 shows the sample standard deviations of all column properties of the data sets. Although, there are only samples of five data points, all values within a column have nearly a symmetric distribution and therefore the sample standard deviation can be measured. Because of the diversity of the recorded plant developments, the occurrences of the different division types differ significantly, also indicated by the high sample standard deviation values. In order to be able to compare the different data in a reasonable way, I register the five plant data sets based on the total number of cells. For this purpose, I determine the maximum number of cells at start ($= 18$) as well as the minimum number of cells at the end ($= 143$) in each data set. Through this, arbitrary values of cell numbers in between can be selected and it is guaranteed that all plants have approximately the same number of cells for a specific time step.

Table 4.3 lists the changed division properties for the registered data sets. Note that it is not always guaranteed that a plant has exactly 18 or 143 cells at a certain recorded time step because of the discrete data acquisition. Hence, the next nearest cell number is chosen which is 19 for 121204, for example. The data sets 120830 and 130508 feature more than 18 cells

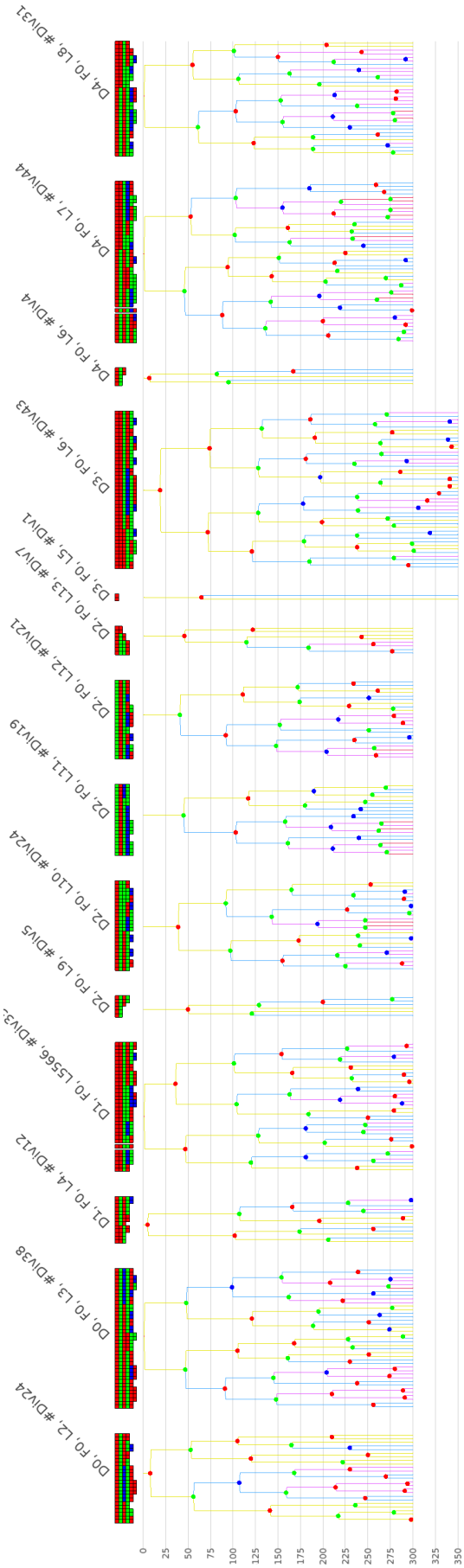


Figure 4.14: Colored lineage trees located in master cell file of all Arabidopsis data sets. The lineage diagram consists of all lineage trees located in the master cell file. On top of each tree, the data IDs (starting with "D"), the cell file (starting with "F"), the lineage ID (starting with "L"), and the number of divisions are given (data ID 0 for 120830, 1 for 121204, 2 for 121211, 3 for 130508, and 4 for 130607). All trees start with an anticlinal division except 121211 because the recording of this data started later compared to the other ones. The trees feature an alternating order of their divisions switching back and forth between growing in width and girth (anticlinal/radial divisions) and growing in height (periclinal divisions).

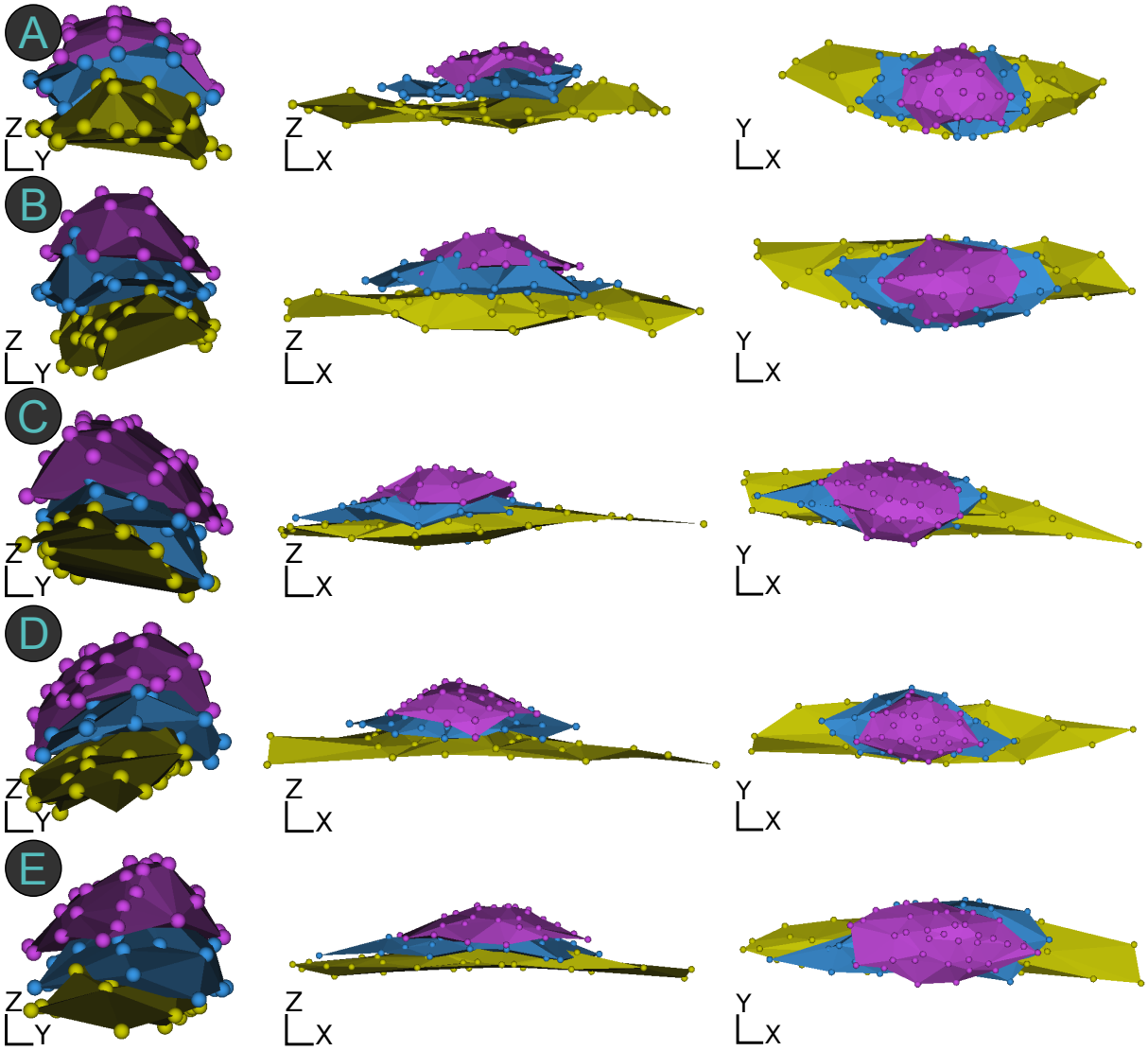


Figure 4.15: Isosurfaces of all registered Arabidopsis data sets. The figure shows three projections of the isosurfaces at the last registered 143-cell state for each data set (**A:** 120830 at $t = 269$, **B:** 121204 at $t = 277$, **C:** 121211 at $t = 230$, **D:** 130508 at $t = 344$, and **E:** 130607 at $t = 213$). At this stage, always three isosurfaces that are similar among the different plant data are generated. For all data sets, the highest number of periclinal divisions always occur at the center of the dome structure.

Data set	Start time	End time	Registered divisions	Division types			Cells at start	Cells at end	Isovalue
				A	P	R			
120830	36	269	127	52	60	15	18	143	3
121204	2	277	130	66	51	13	19	143	3
121211	1	230	126	50	57	19	18	143	3
130508	73	344	126	44	55	27	17	143	3
130607	2	213	124	56	59	9	20	144	3
Sample standard deviation	31.76	50.79	2.19	8.17	3.58	6.84	1.14	0.45	0

Table 4.3: Division properties of Arabidopsis data sets considering registered number of cells. When registering the different plant data based on the number of cells, an adequate comparison of the division types is possible. The sample standard deviation values for the types are much smaller compared to the total temporal analysis.

after 36 and 73 time steps, respectively, while the other remaining ones have the same number of cells right at the beginning of the record. 121211 and 130607 develop much earlier to at least 143 cells. These differences illustrate again the high diversity of the plant data sets. The registration allows a feasible comparison of the data sets and their isovalue results despite this high diversity. For each data set, at most three isosurfaces are generated until the 143-cell stage. The sample standard deviation values for the different division types are reduced by a factor between 2 and 7 in contrast to the generation among all time steps. Consequently, the different data sets share similar numbers of divisions based on the registration depending on the number of cells.

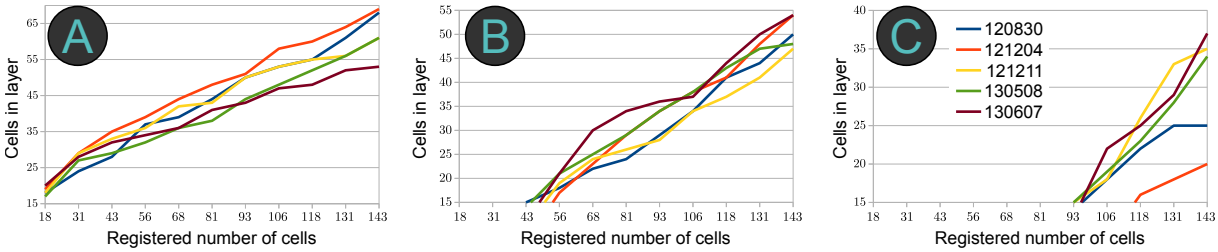


Figure 4.16: Registered comparison of cells in isosurfaces among all Arabidopsis data sets. The three line charts show the almost linearly increasing number of cells in each of the three isosurfaces. For each registered step, they also share approximately the same number of cells. Another observation is that the second isosurface (**B**) starts developing at a number of approximately 50 cells and the third one (**C**) at roughly 100 cells for each data set.

Figure 4.15 shows the visualization of the cells and isosurfaces for each last registered state with approximately 143 cells. All isosurfaces share the same visual appearance. Furthermore, almost all periclinal divisions occur near the center of the developing dome structure. Thus,

these divisions mainly contribute to the height and shape of the dome. In order to quantify these observations, I analyze the number of cells in each isosurface and the corresponding volumes of the α-shapes. For this purpose, eleven registered steps are considered in such a way that the range $[18, 143]$ is subdivided into ten equidistant intervals. The number of cells and the volumes are then determined for each isosurface. Figure 4.16 shows the number of cells in three line charts for each isosurface. In all three images, the number of cells is increasing almost linearly. The second isosurface in Figure 4.16B starts to evolve at approximately 50 cells while the third isosurface in Figure 4.16C is generated at a cell number of around 100. A similar observation is made by analyzing the volumes of the enclosed isosurfaces in Figure 4.17. Although the volumes are increasing almost linearly (except for 130607 in Figure 4.17B), their magnitudes are varying significantly. This indicates that the positions of cell divisions are not fixed among different plant data. Consequently, the order of periclinal divisions influences the volume computations and vice versa. The volumes indicate that these divisions are not distributed equally.

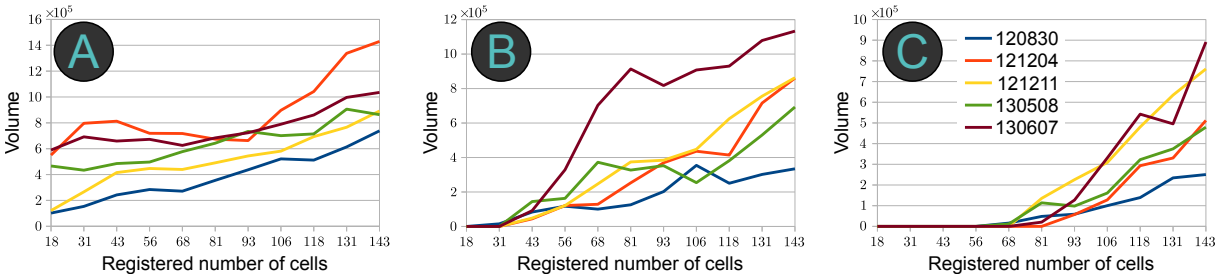


Figure 4.17: Registered comparison of volumes of isosurfaces among all Arabidopsis data sets. For each of the three isosurfaces, the line charts share similar growth patterns except for 130607 in (**B**). However, the individual magnitudes differ significantly among the registered plant data. This is caused by different orders of periclinal divisions resulting in different sizes of the α-shapes for each isosurface.

4.4 Summary

In this chapter, I introduced a novel automatic classification algorithm to determine the division types (anticlinal, periclinal, and radial) of cells within the lateral root of plants. The algorithm is based on cell isosurfaces that represent numbers and locations of periclinal divisions of cells. The normal information of these surfaces using α-shapes are compared to division directions to determine the different division types. The performance analysis shows a quadratic time complexity in the number of total points times the number of time steps. The space usage grows as a function of the number of divisions and total cells for all time steps. The results of the algorithm are influenced by two user-selected thresholds δ and ρ. The outcome of the stability analysis is that small changes of these parameters also result in small changes of the number of division types.

The usefulness of the algorithm was demonstrated by its application to five data sets of the Arabidopsis plant. These are registered based on the number of cells in order to compare them although they have different starting and ending times of the record. After registration, it can be observed that all data sets share similar distributions of divisions, i.e. similar growth behaviors. At the last registered step, at most three isosurfaces are generated that look visually

alike. Considering all data sets, these isosurfaces evolve similarly in the number of cells and volumes and also are similar in the spatial development of periclinal divisions. The visualization of the color-coded lineage trees with the compact information of ordered division schemes reveals an alternating order of anticlinal/radial and periclinal divisions. These division types are an important property of the lateral root growth and I use this information in the next similarity analysis methods explained in the chapters 5 and 6.

After investigating the division types, an analysis of movement patterns is still missing. For this reason, a new visualization method is presented in the next chapter. It is a visual analysis method for finding similar migration patterns in plenty of 3D cell trajectories.

Chapter 5

Similarity Analysis of Cell Trajectories

"All things are the same except for the differences, and different except for the similarities."
— Thomas Sowell, "Penetrating the Rhetoric", The Vision of the Anointed, 1996

Biologists studying animal embryonic development aim to understand how a single cell (*zygote*) develops into well-organized tissues, organs, and eventually a fully-formed viable organism. This development follows a regular behavior in its formation of the tissues. As motivated in the introduction, biologists assume that the migration of cells is a consequence of specific cell fates. More precisely, it is believed that a cell migrates and divides because it has a certain identity and somehow knows how it contributes to organs or tissues. This cell fate commitment could be even influenced by spatial dependencies like the relative position of cells within the developing embryo, for example. The research of cell identities leads to a better understanding of how proliferating cells are able to maintain a regular tissue and organ development. Domain experts seek to find evidence that confirms their hypotheses and reveal insights on the cause of cell migrations and fate decisions. However, the detection of such cell fates and behaviors is a challenge for the analysis process because of the large diversity of the 3D+t data of thousands of cells. This analysis process greatly benefits from interactive visualization methods that automatically extract and classify plenty of cell developments to group similar trajectories.

In this chapter, I introduce a visual analysis method that permits a similarity analysis for 3D+t cell trajectories [FHWL12]. More precisely, I establish a similarity measure for comparing cell trajectories based on a combination of migratory and geometrical features. The results are stored in a similarity matrix that is processed in a hierarchical clustering approach. Figure 5.1 illustrates the generation steps of the presented method. After explaining the clustering algorithm, its performance (Section 5.3.1) and validity (Section 5.3.2) are examined in detail. The usefulness of the method is illustrated by applications to zebrafish and Arabidopsis data sets in Section 5.4. For the zebrafish data, the visual analysis helps in the detection of collective cell migrations and similar tendencies in their development. The visualization of the Arabidopsis data reveals a hitherto undiscovered correlation between the division orientations and subsequent nuclei displacements.

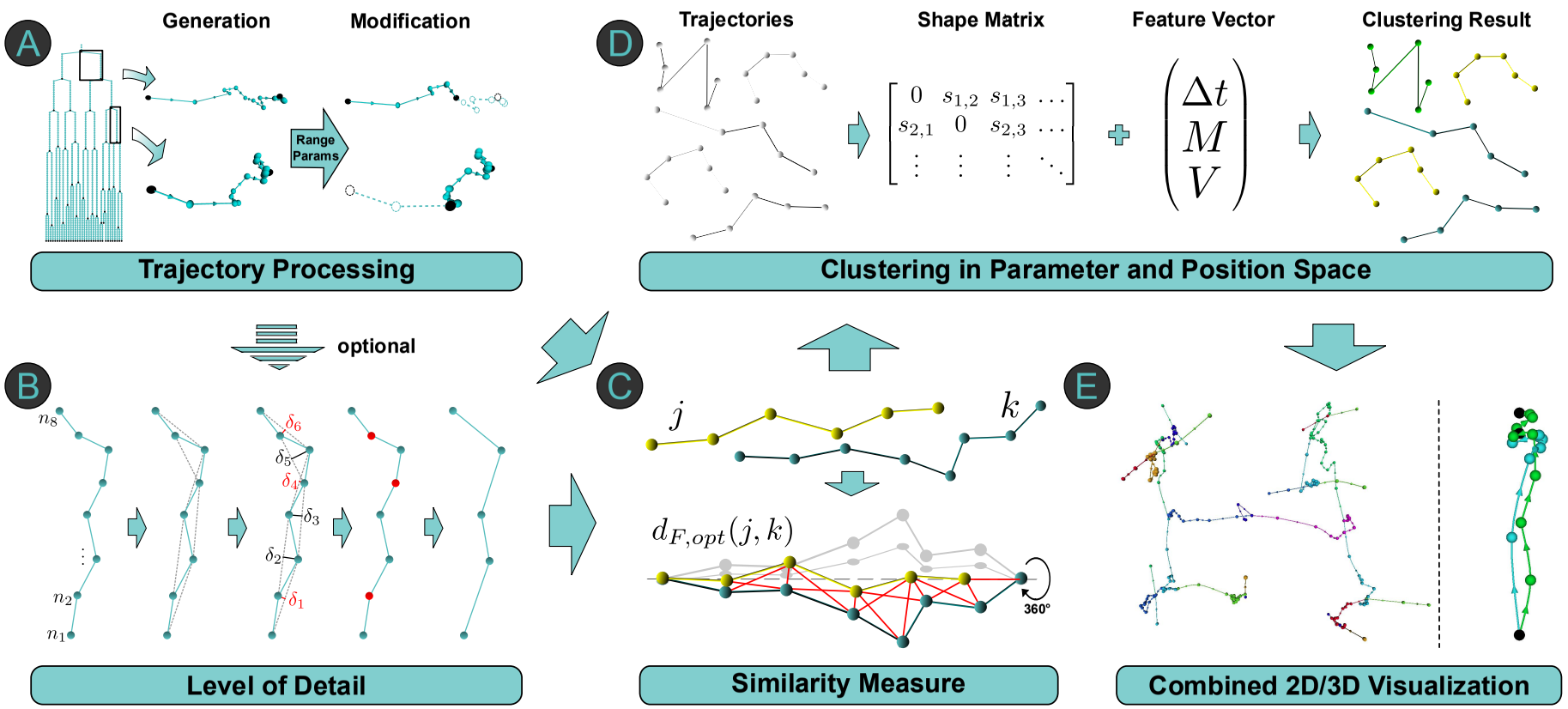


Figure 5.1: Workflow chart of the similarity analysis of cell trajectories. (**A**) The cell trajectories are generated based on the input data (Section 3.5) and modified according to user-selected range parameters such as time, cell cycle length, and the number of divisions. (**B**) Optionally, a level of detail can be set for all trajectories using the edge criterion (Section 5.1). After these geometry-based modifications, the extracted cell-based and movement-based features from each sub-trajectory are stored in a feature vector $\vec{f}$ (Section 5.2). (**C**) For each pair (j, k) of trajectories, k is transformed with suitable rotations around the x-axis in order to find the best similarity measure using the *coupling distance* [EM94b] (Section 5.2). The results are stored in a shape matrix S (Section 5.2). (**D**) The extracted feature vectors $\vec{f}$ and the shape matrix S are given as input data for the clustering algorithm (Section 5.3) resulting in a cluster assignment. (**E**) For validating the clustering results, a combination of several 2D/3D visualizations is provided (Section 5.3.2).

5.1 Cell Trajectory Modifications

Domain experts have the possibility to focus on specific cell trajectories. This is realized by the setting of certain biologically motivated range parameters for time, cell cycle length, and the number of divisions. The choice of parameters may affect the geometry of trajectories in such a way that the range constraints are fulfilled. Furthermore, to focus on the analysis of collective cell migrations and to identify trends, a method for adapting the level of detail is introduced. This modification reduces the complexity of data and simplifies the visual analysis.

Level of Detail for Cell Trajectories

The visualization of thousands of 3D cell trajectories is a challenge for the visual analysis. These trajectories often suffer from little cell position changes in time resulting in overplotting of cell sets in the visualization (Figure 5.2A gives an example). This is caused by small cell

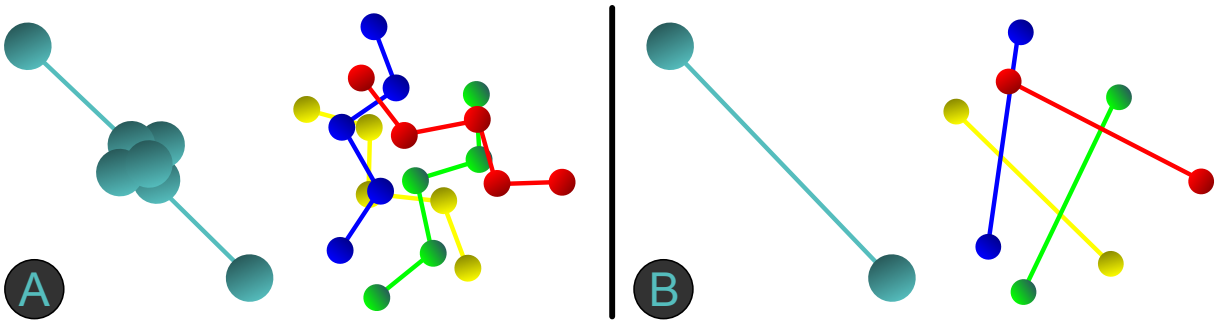


Figure 5.2: Visualization of cell trajectories using lowest level of detail: The examples in (**A**) highlight the visual problems in the visualization of cell trajectories with full level of detail. The left trajectory shows that the centroid computations may result in small cell position changes leading to overplotting. Overlapping cell trajectories within a dense region may also complicate the visual analysis. Both visual issues are addressed in (**B**) by applying the lowest level of detail that simplifies the visualization and consequently the analysis.

migrations between subsequent time steps and the temporal resolution of data acquisition. The overplotting also affects the visualization of complete cell trajectories within a dense region as illustrated in Figure 5.2A. This phenomenon impedes the analysis in both the 2D and the 3D visualizations. In order to avoid overplotting, a *level of detail (LOD)* technique is applied to all cell trajectories. I use the *edge criterion* [Jen89] that is explained in Figure 5.3 in five steps for a two-dimensional example. The procedure works similarly in 3D and can be directly applied to cell trajectories. The main idea is that for each triple of subsequent nodes, a triangle is generated. For each triangle, the shortest distance δ_i between the second node and its opposing line is considered. If δ_i is smaller than some user-selected threshold γ then the second node is erased from the trajectory. This means that the value of γ affects the LOD applied to the trajectories. The edge criterion tends to remove almost collinear nodes prior to those that feature an abrupt change of the trajectory direction. This is intuitively a correct behavior because collinear nodes share a similar movement direction while a variety of different movement orientations should be coarsened only at a lower LOD.

Figure 5.2B shows how the edge criterion can be used to address the visual problems described above. If the lowest LOD is chosen only the start and end points of each trajectory

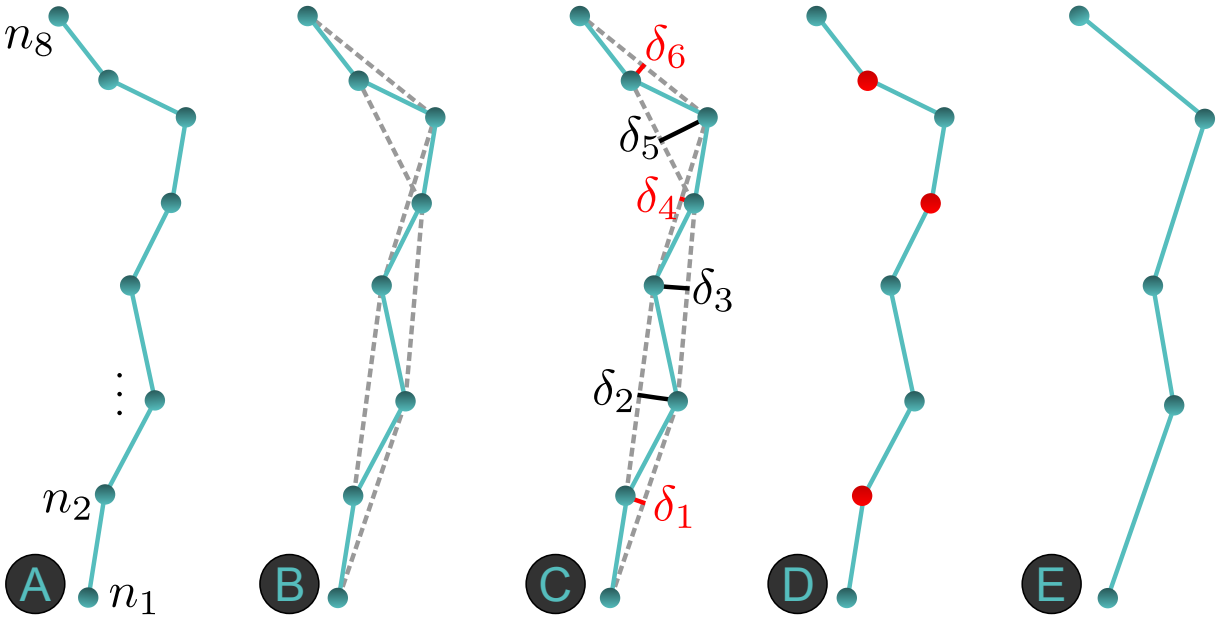


Figure 5.3: Explanation of edge criterion. (**A**) Trajectory with eight nodes $n_1, \ldots, n_8$. (**B**) Each triple of subsequent nodes defines a triangle with a dashed line connecting the first and last nodes. (**C**) For each triple, the shortest length δ_i between the inner node and its opposed dashed line is computed. If this length is smaller than some user-selected threshold γ then the inner node of the triple is erased (red δ_j), else the node (black δ_j) remains in the trajectory. (**D**) The three nodes marked in red are erased and result in the new coarsened version of the cell trajectory (**E**).

remain. Note that the feature computation explained in the next section yields different results if the LOD is altered. For example, if a trajectory with 10 nodes is coarsened to the lowest LOD such that only two nodes remain, only the information provided by these two nodes influence the feature computation. Although different levels of detail could be generated, only two levels are considered in this thesis. Either no LOD is applied at all ($\gamma > 2.5$ in data sets) or the lowest one is chosen in such a way that a single line remains for describing a cell migration ($\gamma < 0.1$ in data sets). This consideration might seem to be a rough simplification of the complex data but it serves as a focused analysis of migration trends. Without a reduction of the LOD, all migration properties and shape structures of a trajectory can be analyzed while for the lowest LOD, a better analysis of main tendencies of single or collective cell migrations is permitted. It furthermore satisfies a requirement for the clustering method explained later.

5.2 Similarity Measure

In the similarity analysis, I use the movement-based features introduced in Section 3.4. There are a lot of distance functions available determining the similarity between trajectories j and k. The most commonly used measure is the *Minkowski distance*:

$$d_M(j,k) = \sqrt[p]{\sum_{i=1}^{n} (j_i - k_i)^p}, \quad l_j = l_k = n, \tag{5.1}$$

or more precisely the *Manhattan distance* $(p = 1)$ and *Euclidean distance* $(p = 2)$ which are metrics and can be computed in linear time. I compute the absolute distance of the delta time information (cell cycle length) and the Euclidean distance of the velocities for each sub-trajectory because they provide a fast computation and compare the actual rating of these features. I additionally include the direction vectors for each sub-trajectory and compute the *geodesic distance* between the spherical coordinates of the vectors. I choose this distance because it represents an adequate similarity of direction vectors in the 3D space. The required features for each trajectory are stored in a feature vector $\vec{f} \in \mathbb{R}^{4L+1}$:

$$\vec{f} = (\Delta t, M, V)^T . \tag{5.2}$$

The first entry is the time value $\Delta t \in \mathbb{N}$ followed by a tuple $M = (\vec{m}_1, \ldots, \vec{m}_L)^T$ of the L single direction vectors $\vec{m}_i \in \mathbb{R}^3$ for each sub-trajectory. The tuple $V = (v_1, \ldots, v_L)^T$ denotes the corresponding velocities $v_i \in \mathbb{R}$. Within any time step range, the lengths of the trajectories and thus the dimension of the feature vectors may vary. Clustering, however, only works with data of the same dimension, i.e. trajectories with the same number of sub-trajectories. After discussion with domain experts, for each individual comparison between a pair of trajectories, the back of the longer one is pruned in such a way that both trajectories share the same length (Figure 5.4). I choose this operation because an extension of the shorter trajectory yields false information that distorts the analysis. Furthermore, a pruning at different positions is not allowed because the biological property of cell trajectories starting at a dividing cell should be preserved. Note that the geometrical structure of a trajectory is not changed, only its representing feature vector is pruned. Also note that this pruning is only required when not reducing the LOD because using the lowest LOD guarantees that a trajectory is represented as one line and has a feature vector $\vec{f}$ of dimension 5.

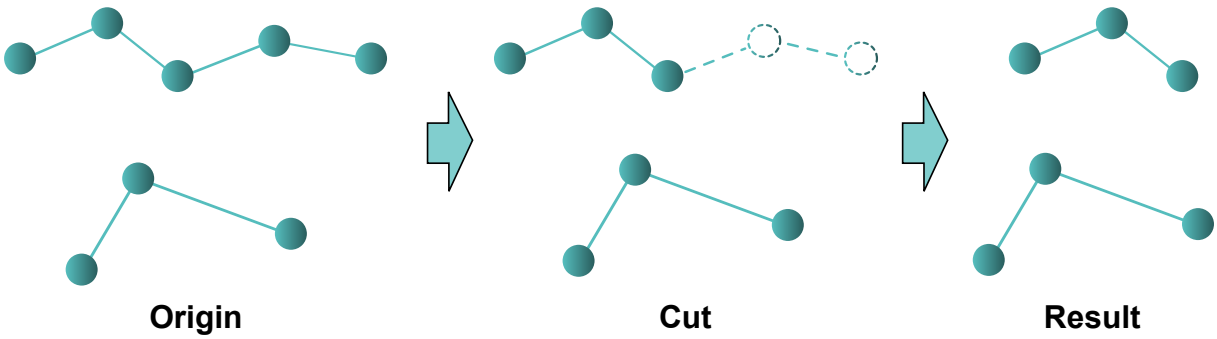


Figure 5.4: Pruning of sub-trajectories. In order to assure the same dimensionality of the feature vectors, trajectories are pruned at the back.

Due to the versatile nature of the input data, a similarity measure only based on single features would not capture all biological events in the data set. Thus, I generate a similarity measure based on combined feature values that covers similarity in cell cycle lengths, velocities, local motions, as well as shapes of entire trajectories. For this purpose, when comparing two trajectories j and k of different lengths, let $L := \min(l_j, l_k)$ be the smaller trajectory length with respect to the pruning approach.

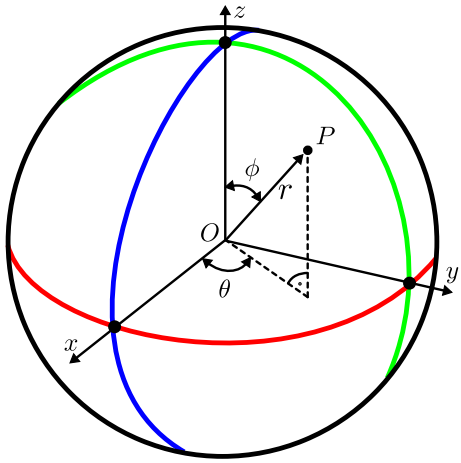
Migration Orientation

I compute the spherical coordinates $P = (r, \phi, \theta)$ of a normalized direction vector $\vec{m} = (m_1, m_2, m_3)^T$ in order to capture and compare the local motions of cells:

$$r = \sqrt{m_1^2 + m_2^2 + m_3^2},$$

$$\theta = \arccos\left(\frac{m_3}{r}\right),$$

$$\phi = \begin{cases} \arctan\left(\frac{m_2}{m_1}\right) & \text{if} \quad m_1 > 0, \\ \operatorname{sgn}(m_2) \cdot \frac{\pi}{2} & \text{if} \quad m_1 = 0, \\ \arctan\left(\frac{m_2}{m_1}\right) + \pi & \text{if} \quad m_1 < 0 \wedge m_2 \geq 0, \\ \arctan\left(\frac{m_2}{m_1}\right) - \pi & \text{if} \quad m_1 < 0 \wedge m_2 < 0. \end{cases}$$



Due to the normalization, r is always one. $\theta \in [0, \pi]$ is called the polar angle while $\phi \in (-\pi, \pi]$ is called the azimuthal angle and sgn is the sign function. If a trajectory has more than one sub-trajectory I compute the mean of all direction vectors in order to average the complete migration of a cell:

$$\vec{m}_a = \frac{1}{L} \sum_{i=1}^{L} \vec{m}_i. \tag{5.3}$$

The resulting mean vector $\vec{m}_a$ is normalized and its spherical coordinates are computed. To measure the similarity between two spherical coordinates $P_j = (1, \phi_j, \theta_j)$ and $P_k = (1, \phi_k, \theta_k)$ the geodesic distance (*great-circle*) is computed which is the shortest distance between two points on the surface of a sphere:

$$d_D(P_j, P_k) = 2 \arcsin\left(\sqrt{\sin^2\left(\frac{|\theta_j - \theta_k|}{2}\right) + \cos\theta_j \cos\theta_k \sin^2\left(\frac{|\phi_j - \phi_k|}{2}\right)}\right). \tag{5.4}$$

In contrast to the Euclidean distance that calculates the length of the straight line between two points, the geodesic distance is measured along the surface of the sphere. I choose this method because it is well-suited for capturing and comparing the motion directions of trajectories.

Cell Cycle

The similarity of cell cycle durations is computed as the absolute difference between two delta time values:

$$d_T(\Delta t_j, \Delta t_k) = |\Delta t_j - \Delta t_k|. \tag{5.5}$$

Note that using the pruning approach and no LOD reduction yields $d_T(\Delta t_j, \Delta t_k) = 0$. But using the lowest LOD, only one line represents each trajectory and all feature vectors share the same number of elements. In this case, the feature vectors are recomputed based on the first and last cell centroid. $d_T(\Delta t_j, \Delta t_k)$ then yields the difference between two cycle durations.

Velocity

I further define the similarity between velocities of trajectories by the Euclidean distance of the tupels V:

$$d_V\left(V_j, V_k\right) = \|V_j - V_k\|_2. \tag{5.6}$$

Using only the pruning approach, all trajectories originate from a dividing cell and share the same temporal order. This distance function then computes the individual differences of velocities starting at the same biological event which makes it a feasible measure to compare cell migration speeds.

Shape

In addition to the distance functions described above, I employ a geometric similarity measure in order to compare the shape of different trajectories. One example of such a similarity mea-

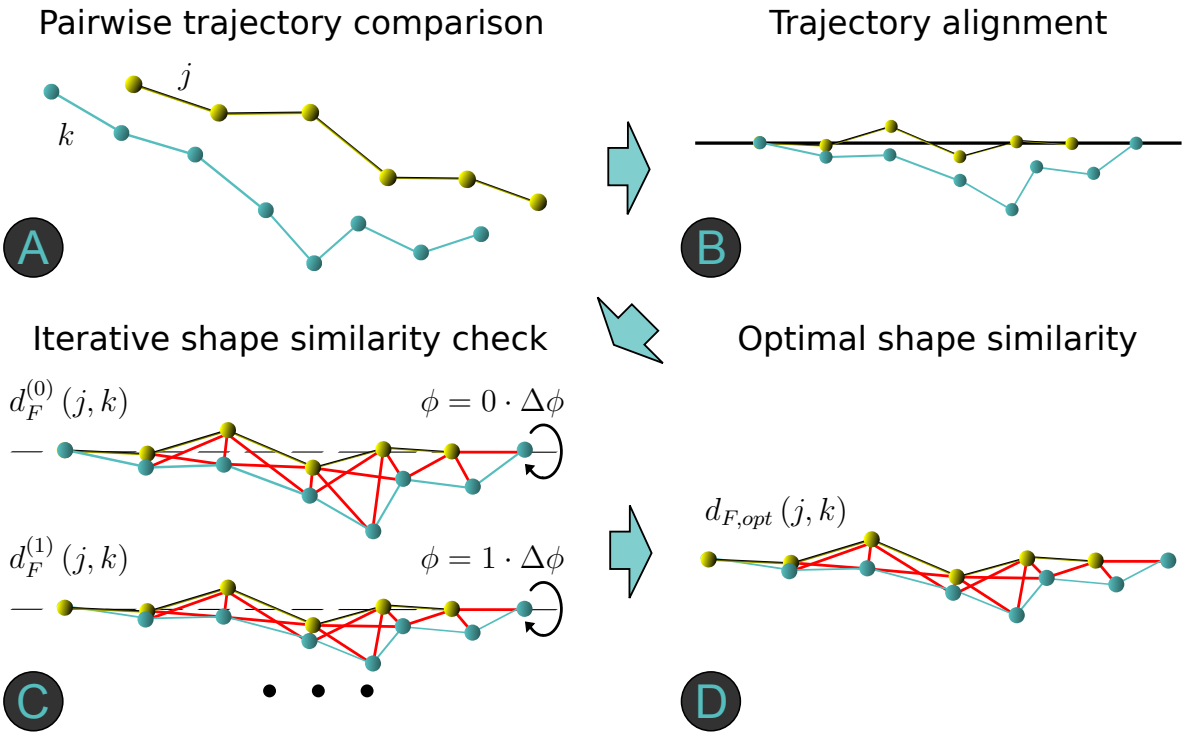


Figure 5.5: Shape similarity computation of trajectories. Each pair of trajectories is considered to compare their shapes (**A**). Prior to computing the coupling distance, both trajectories are aligned on the x-axis according to their direction vector between the first and last cell centroid (**B**) . In each iteration, the second trajectory is rotated along the x-axis and the coupling distance is computed for this constellation (**C**). According to all rotations, the coupling distance is selected in such a way that it maximizes the shape similarity (**D**).

sure is the (bidirectional) *Hausdorff distance*. Intuitively, it computes for each position of one trajectory j the distance to its closest position of another trajectory k and returns the maximum over all these values. However, this measure is prone to noise and it neglects the order of the points on the trajectory. Unlike the Hausdorff distance, the *Fréchet distance* includes the order of traversal of the trajectory which makes it more suitable for the similarity comparison of

shapes [Alt09]:

$$d_F(j,k) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \|\sigma(\alpha(t)) - \tau(\beta(t))\|. \tag{5.7}$$

$\sigma, \tau : [0,1] \to \mathbb{R}^3$ are parameterizations of the two trajectories j and k while $\alpha, \beta : [0,1] \to [0,1]$ range over all continuous and monotone increasing functions. The continuous Fréchet distance can be informally explained by the commonly known connection between a person and a dog by a leash. Imagine the person and the dog walking along a path (curve) from its starting point to its end point. Both are allowed to control their speed, but they cannot backtrack. The Fréchet distance between the two trajectories is the minimal length of a leash that is sufficient for traversing both curves in this manner. Because of the fact that these trajectories correspond to cell developments, the order in which a cell traverses different positions over time is an important biological property for the similarity check. Since the trajectories are defined by discrete points, I apply the *coupling distance* $d_F(j,k)$ of Eiter and Mannila [EM94b] which is a good approximation of the Fréchet distance for discrete curves. The basic idea is to look at all possible couplings between the end positions of all line segments of both trajectories. The distance is then computed in polynomial time using a dynamic programming algorithm.

The coupling distance is sensitive to the spatial location of the trajectories that are being compared and thus they require an alignment. Goodrich et al. [GMO99] as well as Alt and Guibas [AG96] analyze and apply rigid-body transformations for the purpose of matching point sets in 3D space. However, the focus here is on cell trajectories that start with a dividing cell and these are transformed in such a way that both share the same first cell centroid. For this reason, the first cell of each trajectory is translated into the origin. The direction vectors of the first and last cell centroids are calculated for both trajectories. Both are aligned in such a way that their direction vectors lay on the x-axis. One degree of freedom remains, namely the rotation of the trajectory around the x-axis. The angle ϕ for that rotation is determined by an iterative check of both trajectories based on their shape similarity. More precisely, the second trajectory is rotated iteratively around the x-axis by the angle ϕ that maximizes the shape similarity

$$d_{F,opt}(j,k) = \min_{0\le i<n} \left\{ d_F^{(i)}(j,k) \,\middle|_{\phi}\; \phi = i\cdot\Delta\phi,\ \phi\in[0,360) \right\}, \tag{5.8}$$

with i denoting the iteration index and n as the total number of iterations. $\Delta\phi \in (0,360)$ is a user-selected offset angle and has to be a divider of 360. The angle ϕ is increased in each iteration by $\Delta\phi$ until all rotations along the x-axis are performed. In each step i, the coupling distance $d_F^{(i)}(j,k)$ is computed. ϕ is chosen in such a way that it minimizes the coupling distance $d_{F,opt}(j,k)$, thus maximizing the shape similarity. Figure 5.5 illustrates the explained steps. Note that a smaller angle $\Delta\phi$ means a higher computational effort for the similarity analysis. While smaller offsets require more iterations n, the shape comparison also becomes more accurate. I choose an angular discretization of 15 degrees (24 comparisons for each pair of trajectories) because this selection yields an optimal compromise between efficient computation and exact results for the applied data sets. This costly optimization step is done only once for M trajectories and the normalized optimal coupling distances $d_{F,opt}(j,k) = s_{j,k}$ are stored in a

shape matrix $S \in \mathbb{R}^{M \times M}$:

$$S = \begin{bmatrix} 0 & s_{1,2} & s_{1,3} & \dots & s_{1,M} \\ s_{2,1} & 0 & s_{2,3} & \dots & s_{2,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{M,1} & s_{M,2} & \dots & s_{M,M-1} & 0 \end{bmatrix}. \tag{5.9}$$

Note that the coupling distance also takes the length of two trajectories into account. This means that if the lowest LOD is applied, trajectories only based on their lengths between the first and last cell centroid can also be compared because each cell migration is represented by a single line. In conclusion, the similarity of two trajectories j, k based on their feature vectors $\vec{f_j}, \vec{f_k}$ is given by a weighted combination of the individual feature values:

$$\begin{aligned} d_{similarity}(\vec{f_j}, \vec{f_k}) = \lambda_1 \cdot \widehat{d}_D(P_j, P_k) + \\ \lambda_2 \cdot \widehat{d}_V(V_j, V_k) + \\ \lambda_3 \cdot \widehat{d}_T(\Delta t_j, \Delta t_k) + \\ \lambda_4 \cdot s_{j,k}. \end{aligned} \tag{5.10}$$

The weights should satisfy $\sum_{i=1}^{4} \lambda_i = 1$ and $\lambda_i \in [0, 1]$. Since the features manipulated by $\lambda_1, \dots, \lambda_3$ may vary considerably, the three components are normalized to $[0, 1]$, indicated by a hat symbol, to provide a balanced comparison of local and global features. Note that the entries of S are already normalized.

5.3 Hierarchical Clustering of Feature Vectors

By means of the feature vectors that describe trajectory properties and shape-based characteristics, the trajectories are clustered in order to group thousands of cells with similar motion patterns. Clustering is a technique to group objects based on similar features. It is a large area of research and several survey articles exist with focus on general clustering techniques [ELLS11, Rok10] or with focus on clustering of time series and trajectories [WL05, KMNR10]. Common methods are classified into *partitioning, hierarchical, density-based, grid-based,* and *model-based* clustering techniques. However, for domain experts in biology, it is important to retrace the hierarchy of clustered trajectories because it enables them to differentiate between the similarity level of cell migrations. In other words, trajectories that are subsequently merged are more similar in comparison to all other trajectories added later to the same cluster. The clustering process does not need any a priori information about the number of clusters and can also be visualized in a tree diagram called *dendrogram* that serves as a hierarchy representation for detailed cluster analysis. For these reasons, I apply a *hierarchical clustering* approach which is a distance-based unsupervised learning method (i.e. no a priori labeling of the data is given). It merges clusters if they are close to each other and this can be realized using an *agglomerative* "bottom up" approach. Here, each element is assigned to exactly one cluster at the beginning and several clusters are merged during the clustering process. In the contrasting approach, called *divisive* type, all elements start in one big cluster and clusters are split in a "top down" fashion. The divisive type can be computationally expen-

sive if all $2^{k-1} - 1$ possible divisions for k objects are considered [ELLS11, p. 84]. Hence, I choose an agglomerative approach for which the performance is analyzed later in Section 5.3.1. More precisely, each element starts in its own cluster and similar trajectories are joined until a predefined number of clusters or a similarity threshold is attained. The clustering requires the following input: (i) The feature vectors $\vec{f}$ for each trajectory (Equation 5.2), (ii) a similarity measure for the feature vectors (Equation 5.10), (iii) a linkage criterion that defines the way to merge clusters and, (iv) a stopping criterion for the clustering.

The use of different linkage algorithms can result in completely different clustering results. In Section 5.3.2, I apply and compare several linkage types described here and give a detailed analysis of the resulting clusters for the biological data sets. The linkage criterion *single-link* [Sne57] (nearest neighbor) results in the merging of clusters with the shortest distance between any two elements in the two clusters. Here, a small initial cluster can attract locally the other elements one by one leading to a *chain effect*. In the analysis, this linkage algorithm tends to generate one large cluster with other clusters containing only a few elements. While this behavior is suitable for outlier detection, i.e. elements with largest cluster distances, it is not appropriate for the presented analysis. In contrast, the linkage criterion *complete-link* [Sor48] (furthest neighbor) avoids the *chaining* phenomenon because it merges clusters globally based on the longest distance between two elements in two clusters. However, it is strongly affected by outliers and the merging of elements with large distances significantly changes the clustering. Therefore is it also not suitable for the cell data. Other types are *centroid linkage (unweighted pair-group method using centroids - UPGMC)* [Gow67] and *median linkage (weighted pair-group method using centroids - WPGMC)* [Gow67]. For the centroid linkage the geometric center (centroid) of each cluster is computed. The distance between two clusters is then given by the (Euclidean) distance between these two centroids. However, newly formed clusters may significantly change the cluster hierarchy. For the latter type the distances based on the median of each cluster are computed. This is useful when for the distance calculation each element should be equally weighted which is not suited for clustering the cell trajectories. Two criteria yield the best clustering results with this algorithm applied to the zebrafish and Arabidopsis data: *Group average (unweighted pair-group method with arithmetic mean - UPGMA)* [SM58] and *Ward's method* [War63]. The first one defines the distance between two clusters as the average (Euclidean) distance between all pairs of the elements x, y in two clusters X, Y:

$$d_{Average}(X, Y) = \frac{1}{|X||Y|} \sum_{x \in X, y \in Y} d(x, y). \tag{5.11}$$

$|X|, |Y|$ are the cardinalities of each cluster. In contrast, the *weighted average (weighted pair-group method with arithmetic mean - WPGMA)* [McQ57] linkage performs the same computation as the average linkage but weighted distances based on the number of elements in a cluster are considered. Ward's method minimizes the total within-cluster variance, i.e. how far the objects are spread out within a cluster:

$$d_{Ward}(X, Y) = \frac{|X||Y|}{|X| + |Y|} \left\| \overline{X} - \overline{Y} \right\|^2 \tag{5.12}$$

with $\overline{X}, \overline{Y}$ as the centroids. Instead of merging the two most similar elements successively it tends to combine those elements whose merge increases the within-cluster variance to the

Input : Feature vectors $\vec{f}$, shape matrix S, similarity measure, linkage criterion, metric weights, stop criterion.
Output: Cluster assignment

```
1  D = d(j,k) : matrix [1..M, 1..M] of real;
2  begin
3      getMinMax();
4      for j ← 1 to M do
5          for k ← j to M do
6              d(j,k) ← computeSimilarity(x_j, x_k);
7              d(k,j) ← d(j,k);
8              if d(j,k) ≤ minValue then
9                  minValue ← d(j,k);
10                 minRow ← j;
11                 minColumn ← k;
12     clusterSize ← M;
13     matrixReduction ← 0;
14     while clusterSize ≥ clusterStop or !distStop do
15         merge(X_minRow, X_minColumn);
16         clusterSize ← clusterSize−1;
17         matrixReduction ← matrixReduction+1;
18         for j ← 1 to M−matrixReduction do
19             for k ← j to M−matrixReduction do
20                 if j =minRow and k =minColumn then
21                     d(j,k) ← LanceWilliams(X_j, X_k);
22                     d(k,j) ← d(j,k);
23                 if d(j,k) ≤ minValue then
24                     minValue ← d(j,k);
25                     minRow ← j;
26                     minColumn ← k;
27                 foreach d(j,k) do
28                     distStop ← stopCriterion(d(j,k));
```

Figure 5.6: Agglomerative hierarchical clustering algorithm based on trajectory feature and shape similarities.

Linkage	δ_1	δ_2	δ_3	δ_4
Average	0.5	0.5	0	0
Ward	$\frac{\lvert X\rvert+\lvert Z\rvert}{N}$	$\frac{\lvert Y\rvert+\lvert Z\rvert}{N}$	$-\frac{\lvert Z\rvert}{N}$	0

Table 5.1: Parameters for Lance-Williams algorithm [LW67]. $N = |X| + |Y| + |Z|$.

smallest possible degree. Ward's method is well-suited if almost equally sized clusters can be assumed which is the case for the investigated data sets. The implementation of the linkages are realized using the recursive *Lance-Williams algorithm* [LW67]:

$$\begin{aligned} d_{LW}(X \cup Y, Z) = \delta_1 \cdot d(X, Z) + \delta_2 \cdot d(Y, Z) + \delta_3 \cdot d(X, Y) + \\ \delta_4 \cdot |d(X, Z) - d(Y, Z)| , \end{aligned} \tag{5.13}$$

with $X \cup Y$ being the clusters to be merged and $d(\cdot, \cdot)$ being the pairwise distances between clusters X, Y, Z. Different values for δ_i correspond to different linkages criteria. The parameters for the applied methods are given in Table 5.1.

The current distances between single elements and clusters are stored in a symmetrical similarity matrix $D = d(j, k) \in \mathbb{R}^{M \times M}$, where M is the number of trajectories. In each clustering step, this matrix is updated and the dimension of the row and column is decreased. The clustering algorithm proceeds as follows (pseudo-code in Figure 5.6):

- The minimum and maximum values of the similarity measure results are computed for all feature vectors $\vec{f}$ (line 3). This information is required for the normalization of the first three components of the similarity measure in equation 5.10.

- In line 6, the similarity between each pair of trajectories is computed using equation 5.10. Subsequently, the minimum value `minValue` with indexes `minRow` and `minColumn` of the current matrix D are determined and stored (lines 9–11).

- The initial number of clusters is set to M (line 12) and the variable `matrixReduction` indicates the dimension reduction of the matrix D during the clustering (set to zero at the beginning in line 13).

- The while-loop runs until either a certain number of clusters has been reached or if all cluster distances are smaller than a pre-defined threshold. In line 15, the cluster X_{minRow} is merged with the cluster $X_{minColumn}$. As a consequence, the cluster size is decreased by one and the similarity matrix D is reduced by one row and one column, i.e. the row and column with index `minColumn` is removed.

- For each pair of trajectories, the similarities are recomputed using the recursive Lance-Williams algorithm and the Ward linkage criterion in line 21. Note that a recomputation of similarities only affects clusters that are linked to the clusters with index `minRow` and `minColumn`, respectively. In the loop, the new local minimum is determined for all entries in the matrix D (lines 24–26). In line 28, in case the threshold stop criterion is selected, it is checked whether the cluster distances are all smaller than the threshold.

5.3.1 Performance Analysis

Table 5.2 on page 70 shows a summary of the number of sub-trajectories and computation times in seconds for each main task in the pipeline: Trajectory generation, trajectory modification, similarity measure, and clustering. The algorithm is applied to all zebrafish and Arabidopsis data sets with the same hardware setting as in chapter 4 (Intel Core i7, 3.20 GHz, 12 GB of memory and an NVidia GTX 480). The number of sub-trajectories and the other three columns are based on different parameter settings for each data set (Epiboly: $t \in [0, 65]$, all trees with division range in $[8, 64]$, length > 5; Tailbud: all trees, all times steps, length > 30; Arabidopsis: all trees and all time steps). These settings are identical to the ones used to present the application results and are motivated later in Section 5.4.

The initial generation of the trajectories has a time complexity of at most $\mathcal{O}(N + ML_{\text{max}})$ with $N = \sum_{l=1}^{L} n_l$ as the sum of all tree nodes n_l in all L cell lineages because the tracking information is stored in a binary tree and its traversal is realized in $\mathcal{O}(n_l)$. M is the number of trajectories and L_{max} is the maximum length of all trajectories. The term ML_{max} refers to the feature computation for all sub-trajectories. The trajectory generation is done only once for each data set. This result is stored and modified based on the input range parameters and the chosen LOD. The second task in modifying all trajectories is realized in linear time in the number of trajectories M.

The clustering algorithm has a time complexity of at most $\mathcal{O}(M^3)$. More precisely, it is $\mathcal{O}(\frac{M^2-M}{2} \cdot (M - C))$ with $C \leq M$ as the desired number of clusters. The term $\frac{M^2-M}{2}$ illustrates the time of scanning the symmetric $M \times M$ matrix D in which the diagonal only contains zeros. The value of $M - C$ refers to the number of iterations and on the number of clusters that should be generated. The more clusters are chosen the faster the algorithm is finished. Rafsanjani et al. [RVC12] present a survey of recent agglomerative hierarchical clustering techniques with a comparison of space and time complexities. While there are other algorithms with improved performance under certain assumptions ($\mathcal{O}(M^2 \log M)$ [GRS98, GRS00], $\mathcal{O}(M \log^3 M)$ [KBXS12], $\mathcal{O}(M \log^2 M)$ [EDSN11]), it is still sufficient for these data sets because only a subset of all cell trajectories is considered.

In comparison with the cubic time complexity of the clustering method, the similarity measure needs more time to be calculated. The computation of the coupling distance requires that each pair of elements of the sub-trajectories for two trajectories is compared with each other. This has an upper bound time complexity of $\mathcal{O}(I \cdot \frac{M^2-M}{2} \cdot L_{\text{max}}^2)$ with I as the number of similarity checks to maximize the shape similarity between two trajectories. Although, $I = 24$ is a constant, the similarity measure has a bad time complexity depending on the number of trajectories and their lengths. This fact also explains the long computation time for the similarity measure of the tailbud data (approximately 20 minutes) in contrast to the epiboly data set (approximately $2^1/_2$ minutes) because of their large difference in the number of sub-trajectories. The same observation holds for the different Arabidopsis plants with computation times between approximately 3 and 8 minutes for the similarity measure while all other tasks take less than a second. However, this huge computation time may be strongly reduced depending on the analysis purpose: The coupling distance needs only to be computed when the shape between trajectories should be investigated. This means that for $\lambda_4 = 0$ only the migration-based features are compared and clustered. Furthermore, when only investigating the trends of migrations and collective movement behaviors, then the lowest LOD can be used. This means that

Data set w/o level of detail	Number of sub-trajectories	Computation times [s]			
		Trajectory generation	Trajectory modification	Similarity measure	Clustering
Epiboly	15,897	16.16	0.14	160.74	8.94
Epiboly (LOD)	1,132	16.16	0.20	10.8	7.43
Tailbud	51,432	100.96	1.07	1275.34	13.43
Tailbud (LOD)	1,331	100.96	1.18	15.73	12.53
120830	21,084	0.19	0.01	205.4	0.47
120830 (LOD)	339	0.19	0.04	1.13	0.35
121204	19,849	0.27	0.02	180.29	0.39
121204 (LOD)	323	0.19	0.03	1.04	0.33
121211	28,636	0.23	0.02	380.13	1.15
121211 (LOD)	502	0.23	0.04	2.45	0.90
130508	19,102	0.16	0.02	166.12	0.30
130508 (LOD)	277	0.16	0.03	0.8	0.24
130607	32,084	0.26	0.02	476.5	1.10
130607 (LOD)	514	0.26	0.05	2.44	0.92

Table 5.2: Computation times for similarity analysis of all data sets. The table lists the number of sub-trajectories and the individual computation times in seconds for each main task in the similarity analysis using Ward's linkage. Note that the number of sub-trajectories is identical to the number of cell trajectories in the data sets for which the lowest LOD is applied.

all trajectories are only presented by their first and last centroids. Consequently, $L_{\text{max}} = 1$ and the time complexity is quadratic. The reduced computation times are listed in Table 5.2. The generated shape matrix S based on the coupling distance is generated only once and stored on the disk for fast reprocessing in later sessions. However, if the structure of any trajectory is changed by varying the input parameters or by using another LOD, then S has to be recomputed. Nevertheless, by using preprocessed results, the data is loaded in a few seconds and the visual analysis is realized in real-time.

With respect to space usage, the clustering algorithm requires all features vectors $\vec{f} \in \mathbb{R}^{4L+1}$, the shape matrix $S \in \mathbb{R}^{M \times M}$ and a temporary similarity matrix D with an initial dimension of $M \times M$ that is reduced in each iteration. The clustering results are stored in a vector of trajectory IDs for each cluster. Consequently, the algorithm has a space complexity of at most $\mathcal{O}(ML_{\text{max}} + M^2)$. For example, for the Arabidopsis data 130607, the space usage of the algorithm is approximately 5 MiBs while for the tailbud data it is approximately 30 MiBs (64 bits double precision).

5.3.2 Cluster validity

In order to evaluate the presented clustering approach, a *cluster validity* check is performed. In general, there are three strategies how this can be realized [TK99, p. 596–608]. *External criteria* are based on pre-specified structural assumptions (e.g. labels) imposed on the data set but not regarded in the clustering itself. This information is often created by human experts and considered as a gold standard used for evaluation. *Internal criteria* consider the information of the similarity matrix, for example, and validate how well the cluster approach preserves the pairwise distances. This means that the compactness and goodness of a clustering structure are measured. The third group, called the *relative criteria*, evaluates the clustering structure by comparing it to other clustering results using the same algorithm but different parameters. In this section, the internal and relative properties are investigated because no labeled a priori knowledge of the biological data sets is available. I first validate which linkage methods are suited for clustering the data sets. Afterwards, I analyze the parameter stability and the cluster robustness with respect to the weight parameters λ_i of Equation 5.10. The analysis of the internal quality of the clustering permits a way to measure the *goodness-of-fit* of linkage types applied to all data sets. I realize this measurement using the *cophenetic correlation coefficient* [SR62]:

$$c = \frac{\sum_{j<k}(d_{jk} - \overline{d})(y_{jk} - \overline{y})}{\sqrt{\sum_{j<k}(d_{jk} - \overline{d})^2 \sum_{j<k}(y_{jk} - \overline{y})^2}}. \tag{5.14}$$

d_{jk} are the similarity and y_{jk} are the dendrogrammatic distances between trajectories $j, k \in [1, M]$. $c \in [0, 1]$ and $\overline{d}, \overline{y}$ are the averages of the d_{jk} and y_{jk}, respectively. This measurement describes how faithfully the cluster hierarchy preserves the pairwise distances between the original data. The closer the coefficient is to 1, the more accurately the clustering result reflects the original data. Table 5.3 on page 72 lists the different coefficients for the linkage types explained in Section 5.3. The clustering is applied to all data sets w/o reduction of LOD based on the orientation of trajectories. The numbers in bold type illustrate the highest score for each data set. The average linkage features most of the time the highest values with similar results for the centroid and Ward's method. However, the remaining linkage types have $10 - 20\%$ smaller values with even $25 - 50\%$ for the single type. The same linkage behavior is also concluded by Saraçli et al. [SDD13] applied to simulation results w/o outliers. In the visual analysis of the clusters, the average linkage and Ward's method yield similarly good results in contrast to the other linkages. Therefore, I use these two methods for analyzing the biological data sets in Section 5.4. As already mentioned before, using the single linkage results in one big cluster and many clusters with few elements. This behavior of inappropriate clustering is also confirmed by the small cophenetic coefficient.

The choice of the weight parameters λ_i defines the ratio how migratory and geometrical features of cell developments are taken into account in the similarity analysis. Consequently, it is important to check how robust the clustered trajectories are when the weights are changed. A certain cluster setting is selected as a reference hierarchy H to be compared with the other clustering result C to analyze the relative differences when changing a specific pair of λ_i. This is realized with the same data set, identical number of trajectories M, and the same number of clusters k. The weights are slightly steered by small steps in such a way that $\lambda_a + \lambda_b = 1(a \neq b \wedge a, b \in [1, 4])$ is always satisfied. There are many measurements to evaluate the similarity between cluster results [WW07] but I use the *Jaccard index (J)*, the *Fowlkes-Mallows*

	Average	Centroid	Complete	Median	Single	Ward	Weighted average
Epiboly	**0.62**	0.61	0.30	0.31	0.32	0.60	0.52
Epiboly (LOD)	**0.70**	0.68	0.61	0.61	0.44	0.67	0.61
Tailbud	**0.71**	0.71	0.52	0.61	0.32	0.64	0.64
Tailbud (LOD)	**0.69**	0.68	0.54	0.59	0.47	0.58	0.54
120830	**0.78**	0.78	0.67	0.61	0.52	0.71	0.55
120830 (LOD)	**0.78**	0.77	0.68	0.68	0.58	0.76	0.74
121204	0.70	**0.72**	0.67	0.63	0.32	0.67	0.66
121204 (LOD)	**0.75**	0.74	0.69	0.64	0.54	0.73	0.62
121211	**0.78**	0.75	0.60	0.58	0.55	0.68	0.67
121211 (LOD)	**0.76**	0.75	0.68	0.58	0.49	0.72	0.70
130508	**0.81**	0.81	0.73	0.55	0.58	0.70	0.64
130508 (LOD)	0.77	**0.79**	0.72	0.66	0.48	0.77	0.70
130607	**0.74**	0.74	0.64	0.61	0.31	0.71	0.60
130607 (LOD)	**0.74**	0.74	0.66	0.64	0.38	0.71	0.65

Table 5.3: Cophenetic correlation coefficient for different linkage types applied to all data sets. Each cell entry shows the coefficient for clustering based on orientations only.

index (FM) [FM83], and the *F-Measure (F)* [vR79] because they are commonly used for the comparison of clustering results. All these measurements yield values in $[0, 1]$ for which one refers to identical cluster structures and zero refers to no common elements in both hierarchies. All consider the number of points that are common or uncommon to two hierarchy structures:

- TP (true positives) is the number of points that occur in the same cluster in both H and C.
- FP (false positives) is the number of points that occur in the same cluster in H but not in C.
- FN (false negatives) is the number of points that occur in the same cluster not in H but in C.
- TN (true negatives) is the number of points that are in different clusters in both H and C.

Jaccard Index (J): This index measures the number of objects common to both hierarchies H and C divided by the total number of elements in both clusters:

$$\mathrm{J} = \frac{|H \cap C|}{|H \cup C|} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \tag{5.15}$$

Fowlkes-Mallows Index (FM): This index computes the geometric mean of the *precision* rate P and the *recall* rate R. Precision is the fraction of correctly out of all retrieved instances while recall is the fraction of correctly retrieved instances out of all existing matching instances. In order to compute the FM index, each cluster hierarchy of H and C is cut to produce $k = 2, \ldots, M-1$ clusters for each dendrogram. This means that the results can be visualized in a plot of FM_k against k to compare them with other cluster hierarchies. The computation of a specific cluster k is realized by the following formula [FM83]:

$$\mathrm{FM} = \sqrt{\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \cdot \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}} = \sqrt{\mathrm{P} \cdot \mathrm{R}}. \tag{5.16}$$

F-Measure (F): The F_δ-Measure can be used to balance the contribution of FN using a weight parameter δ but I consider here $\mathrm{F} = F_1$ which is interpreted as the harmonic mean of P and R [vR79]:

$$\mathrm{F} = \frac{2\mathrm{P} \cdot \mathrm{R}}{\mathrm{P}+\mathrm{R}}. \tag{5.17}$$

I apply the evaluation to one Arabidopsis plant (because the analysis of the other plants yields similar outcomes) and the two zebrafish data sets using reference settings that are motivated by the application results in Section 5.4. The clustering is performed using the average linkage criterion. For the Arabidopsis, a reference parameter setting of $\lambda_a = 0.8, \lambda_b = 0.2$ and $k = 4$ is chosen. For these values, the visual four clusters represent best a correlation between the division types and the subsequent orientations of nuclei migrations. More details about this are given later in Section 5.4. For the zebrafish data, a reference setting of $\lambda_a = 0.5, \lambda_b = 0.5$ is selected ($k = 5$ for epiboly and $k = 6$ for tailbud) in such a way that both represented feature values are weighted equally. The number of clusters k is biologically motivated and chosen based on the outcome of the dendrogram. For example, the Arabidopsis plant features a common growth and division direction, thus the number of clusters is set to four (more details in Section 5.4). Regarding the dendrogram, a link whose height differs significantly from the height of the links below is an indicator for an inconsistent link. This means that the trajectories joined at this stage are much farther apart from each other than their previous components and that this link could be cut to form an additional cluster.

Table 5.4 on page 74 shows the computed three indexes for all possible pairs of weight parameters for the Arabidopsis plant 120830. Except for the pair λ_3, λ_4, nearly all indexes have values greater than 0.9 for parameter changes of at most 0.16. This cluster stability is also confirmed in the visual analysis of the four clusters for which the correlation result can still be observed when the changes are smaller than 0.16. In particular, for λ_2, λ_3 and λ_2, λ_4, the values are quite high for the same bandwidth of changing parameters. This means that the velocities correlate with the time durations and the lengths of the trajectories. Consequently, most of the nuclei migrate continuously with constant speed. The variation of values for the pair λ_3, λ_4 is assumed to be caused by a higher variance of the time features in comparison to the shape structure (length when using the lowest LOD) of cell trajectories. The index values are different for the epiboly data set. In Table 5.5 on page 75, the clustering is stable for λ_2, λ_4

	λ_a, λ_b / Results	0.60, 0.40	0.64, 0.36	0.68, 0.32	0.72, 0.28	0.76, 0.24	0.80, 0.20	0.84, 0.16	0.88, 0.12	0.92, 0.08	0.96, 0.04	1.00, 0.00
$a = 1, b = 2$	J	0.53	0.53	0.90	0.96	0.99	1	0.88	0.97	0.94	0.93	0.84
	FM	0.72	0.72	0.95	0.98	0.99	1	0.94	0.99	0.97	0.97	0.91
	F	0.70	0.70	0.95	0.98	0.99	1	0.94	0.99	0.97	0.97	0.91
$a = 1, b = 3$	J	0.54	0.85	0.93	0.87	0.88	1	0.87	0.96	0.85	0.84	0.88
	FM	0.72	0.92	0.96	0.93	0.94	1	0.93	0.98	0.92	0.91	0.94
	F	0.70	0.92	0.96	0.93	0.94	1	0.93	0.98	0.92	0.91	0.94
$a = 1, b = 4$	J	0.52	0.82	0.88	0.90	0.88	1	0.99	0.87	0.87	0.90	0.90
	FM	0.71	0.90	0.94	0.95	0.94	1	1	0.93	0.93	0.94	0.94
	F	0.69	0.90	0.94	0.95	0.94	1	1	0.93	0.93	0.94	0.94
$a = 2, b = 3$	J	0.80	0.94	0.95	0.84	1	1	1	1	1	1	1
	FM	0.89	0.97	0.98	0.92	1	1	1	1	1	1	1
	F	0.89	0.97	0.98	0.92	1	1	1	1	1	1	1
$a = 2, b = 4$	J	0.96	0.98	0.96	0.96	0.98	1	0.98	0.98	0.98	0.98	0.98
	FM	0.98	0.99	0.98	0.98	0.99	1	0.99	0.99	0.99	0.99	0.99
	F	0.98	0.99	0.98	0.98	0.99	1	0.99	0.99	0.99	0.99	0.99
$a = 3, b = 4$	J	0.83	0.86	0.84	0.83	0.84	1	0.53	0.53	0.89	0.54	0.54
	FM	0.91	0.93	0.92	0.91	0.92	1	0.70	0.70	0.94	0.71	0.71
	F	0.91	0.93	0.91	0.91	0.91	1	0.69	0.69	0.94	0.70	0.70

Table 5.4: Cluster measurements for investigating the parameter stability of the Arabidopsis plant 120830 using the lowest LOD. For each parameter setting with $k = 4$ using the average linkage, and $\lambda_a + \lambda_b = 1, a \neq b \wedge a, b \in [1, 4]$ in steps of 0.04, the Jaccard index (J), the Fowlkes-Mallows index (FM), and the F-Measure (F) are computed to quantify the similarity between the resulting clusters. The cell colored in light blue indicates the parameters of the reference cluster.

	λ_a, λ_b / Results	0.30, 0.70	0.34, 0.66	0.38, 0.62	0.42, 0.58	0.46, 0.54	0.50, 0.50	0.54, 0.46	0.58, 0.42	0.62, 0.38	0.66, 0.34	0.70, 0.30
$a = 1, b = 2$	J	0.41	0.47	0.46	0.62	0.49	1	0.71	0.50	0.45	0.54	0.37
	FM	0.59	0.64	0.63	0.77	0.66	1	0.84	0.68	0.64	0.71	0.57
	F	0.58	0.64	0.63	0.77	0.66	1	0.83	0.67	0.62	0.70	0.54
$a = 1, b = 3$	J	0.33	0.57	0.56	0.67	0.55	1	0.46	0.66	0.52	0.52	0.51
	FM	0.51	0.73	0.72	0.81	0.71	1	0.63	0.79	0.69	0.69	0.68
	F	0.5	0.73	0.71	0.81	0.71	1	0.63	0.79	0.69	0.68	0.67
$a = 1, b = 4$	J	0.46	0.6	0.53	0.55	0.60	1	0.55	0.50	0.43	0.48	0.43
	FM	0.65	0.75	0.69	0.71	0.75	1	0.71	0.68	0.61	0.68	0.61
	F	0.63	0.75	0.69	0.71	0.75	1	0.71	0.67	0.61	0.65	0.60
$a = 2, b = 3$	J	0.59	0.98	0.5	0.51	0.96	1	0.98	0.98	0.98	0.82	0.92
	FM	0.77	0.99	0.71	0.71	0.98	1	0.99	0.99	0.99	0.91	0.96
	F	0.74	0.99	0.67	0.67	0.98	1	0.99	0.99	0.99	0.90	0.96
$a = 2, b = 4$	J	0.95	0.96	0.99	1	0.97	1	0.96	0.96	0.99	0.94	0.95
	FM	0.97	0.98	1	1	0.99	1	0.98	0.98	0.99	0.97	0.98
	F	0.97	0.98	1	1	0.99	1	0.98	0.98	0.99	0.97	0.98
$a = 3, b = 4$	J	0.97	1	1	1	1	1	0.99	1	0.98	0.99	0.98
	FM	0.98	1	1	1	1	1	0.99	1	0.99	0.99	0.99
	F	0.98	1	1	1	1	1	0.99	1	0.99	0.99	0.99

Table 5.5: Cluster measurements for investigating the parameter stability of the epiboly data set using the lowest LOD. For each parameter setting with $k = 5$ using the average linkage, and $\lambda_a + \lambda_b = 1, a \neq b \wedge a, b \in [1, 4]$ in steps of 0.04, the Jaccard index (J), the Fowlkes-Mallows index (FM), and the F-Measure (F) are computed to quantify the similarity between the resulting clusters. The cell colored in light blue indicates the parameters of the reference cluster.

and λ_3, λ_4. However, when the weight parameter for orientation λ_1 is taken into account, the values are much smaller around 0.7. The visual analysis of the five clusters show that they start to differ significantly for changes above 0.1. These observations are explained by the large diversity of arbitrary cell migration orientations such that even small changes of the weight parameters affect the cluster results. For the analysis of the parameter stability of the tailbud data in Table 5.6 on page 77, the values are even smaller for all pairs of weight parameters. This means that for small changes of the parameters the trajectories are assigned to different clusters.

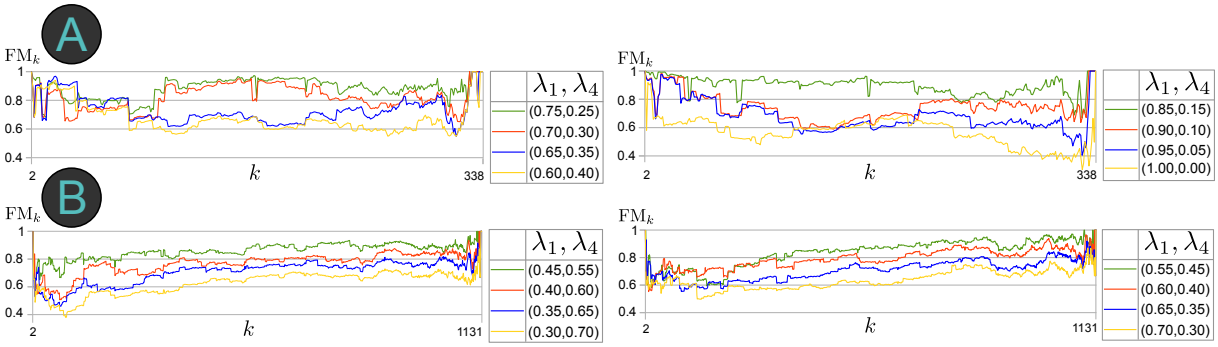


Figure 5.7: Plot of Fowlkes-Mallows index FM$_k$ against clusters k for weight parameters λ_1 and λ_4. The line charts in (**A**) show the relative differences of the hierarchy clusters for the Arabidopsis data set 120830 ($M = 339$) with decreasing (left chart) and increasing λ_1 (right chart). The weight parameters are changed in steps of 0.05 with respect to a reference parameter setting of $\lambda_1 = 0.8, \lambda_4 = 0.2$. In (**B**), the same pair of weight parameters are changed slightly for the epiboly data set ($M = 1132$) with a reference setting of $\lambda_a = 0.5, \lambda_b = 0.5$. For both data sets the lowest LOD is used. Thus, the shape corresponds to the length of the trajectories.

A more detailed analysis of the Fowlkes-Mallows index FM$_k$ plotted against the number of clusters k reveals some more information. Figure 5.7 shows the resulting line charts for the Arabidopsis plant and the epiboly data set using the average linkage criterion. Note that for small and large numbers of clusters k with respect to the amount of trajectories M, FM$_k \to 1$. This is a property of the method that even occurs when the cluster hierarchies are independent [WW07]. In fact, the Fowlkes-Mallows index is based on a strong null hypothesis, i.e. that there is no relationship between two cluster hierarchies. But for each line chart, I focus on the same number of clusters and a fixed linkage type applied to the same data set. Through this, I can solely investigate the relative hierarchy differences according to changing weight parameters. Especially, the middle regions of the plots provide a meaningful interpretation. The pair of line charts in Figure 5.7A shows the resulting plots for the Arabidopsis plant 120830. On the left side, the index is plotted for decreasing λ_1 in steps of 0.05 while on the right side λ_1 increases with respect to the reference parameter setting. A significant change is observed for the blue line ($\lambda_1 = 0.65, \lambda_4 = 0.35$) in comparison to the red one. This means that the cluster memberships of trajectories have changed considerably. The same behavior is shown for increasing λ_1 even earlier according to the red line in the right plot ($\lambda_1 = 0.9, \lambda_4 = 0.1$). This robust cluster stability is caused by the main movement orientations of nuclei in anticlinal and periclinal directions within the lateral root. In contrast, the line plots for the epiboly data set in Figure 5.7B share a common behavior and have similar relative differences. This means that even small changes of the weights affect the clustering results and the trajectory assignments because of the high

	λ_a, λ_b / Results	0.30, 0.70	0.34, 0.66	0.38, 0.62	0.42, 0.58	0.46, 0.54	0.50, 0.50	0.54, 0.46	0.58, 0.42	0.62, 0.38	0.66, 0.34	0.70, 0.30
$a = 1, b = 2$	J	0.44	0.48	0.43	0.44	0.39	1	0.43	0.48	0.47	0.51	0.38
	FM	0.62	0.68	0.61	0.63	0.6	1	0.61	0.67	0.65	0.69	0.58
	F	0.61	0.65	0.6	0.61	0.56	1	0.61	0.65	0.64	0.68	0.55
$a = 1, b = 3$	J	0.6	0.7	0.81	0.85	0.71	1	0.45	0.47	0.46	0.41	0.42
	FM	0.77	0.83	0.89	0.92	0.83	1	0.62	0.64	0.63	0.59	0.6
	F	0.75	0.83	0.89	0.92	0.83	1	0.62	0.64	0.63	0.58	0.6
$a = 1, b = 4$	J	0.45	0.55	0.43	0.49	0.64	1	0.42	0.67	0.37	0.45	0.49
	FM	0.62	0.71	0.61	0.66	0.78	1	0.59	0.8	0.54	0.62	0.65
	F	0.62	0.71	0.6	0.66	0.78	1	0.59	0.8	0.54	0.62	0.65
$a = 2, b = 3$	J	0.38	0.41	0.36	0.36	0.42	1	0.32	0.55	0.4	0.41	0.36
	FM	0.56	0.58	0.53	0.53	0.6	1	0.49	0.72	0.57	0.59	0.53
	F	0.56	0.58	0.53	0.53	0.59	1	0.49	0.71	0.57	0.58	0.53
$a = 2, b = 4$	J	0.51	0.53	0.49	0.81	0.54	1	0.39	0.44	0.51	0.56	0.64
	FM	0.68	0.69	0.66	0.89	0.7	1	0.56	0.61	0.68	0.72	0.78
	F	0.68	0.69	0.66	0.89	0.7	1	0.56	0.61	0.68	0.72	0.78
$a = 3, b = 4$	J	0.4	0.36	0.54	0.66	0.68	1	0.75	0.71	0.69	0.63	0.58
	FM	0.59	0.53	0.71	0.8	0.81	1	0.85	0.83	0.82	0.77	0.74
	F	0.57	0.53	0.7	0.8	0.81	1	0.85	0.83	0.82	0.77	0.73

Table 5.6: Cluster measurements for investigating the parameter stability of the tailbud data set using the lowest LOD. For each parameter setting with $k = 6$ using the average linkage, and $\lambda_a + \lambda_b = 1, a \neq b \wedge a, b \in [1, 4]$ in steps of 0.04, the Jaccard index (J), the Fowlkes-Mallows index (FM), and the F-Measure (F) are computed to quantify the similarity between the resulting clusters. The cell colored in light blue indicates the parameters of the reference cluster.

variance of cell migration orientations. Furthermore, the two images in Figure 5.7B, each with four line plots, are nearly symmetrical with respect to the reference parameter setting which can also be seen in Table 5.5 on page 75. This means that the outcome of the relative differences between the line plots are independent of the arbitrarily chosen reference cluster for this pair of parameters.

Although there are four weights λ_i that can be steered in the similarity measure, only two of them are allowed to be nonzero. This means that I do not consider any setting of three or four λ_i that are nonzero in the analysis of the biological data. I have decided on this because for a concurrent steering of three or four weight parameters, a meaningful parameter stability is required and the cluster results are hard to interpret.

As a result, the parameter setting for the Arabidopsis data is more robust to changes in contrast to the zebrafish data sets. These changes are also confirmed in the visual analysis of the clustered trajectories. The robust behavior for the Arabidopsis plant is explained by the fact that the data has a high quality with no outliers and the cell developments follow a main growth direction in height, width and length. The same observations are confirmed analyzing the other Arabidopsis data sets. In contrast, the zebrafish data has a low quality and high diversity of cell trajectories with arbitrary cell migration directions and lengths. This versatile behavior is the cause for varying cluster assignments of single trajectories when the parameters are changed slightly. Because of this reason, only one $\lambda_i \neq 0$ and thus I consider a single feature of cell trajectories in Section 5.4 when investigating the zebrafish data sets. In this case, the presented clustering approach degenerates to a standard hierarchical clustering algorithm.

Visual Analysis of Cluster Results

In this section, I briefly explain the different visual analysis approaches for investigating the clustering results.

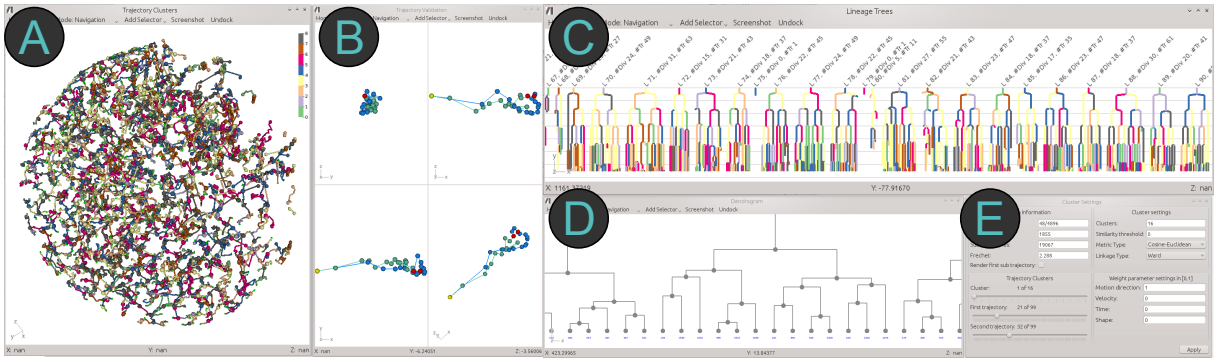


Figure 5.8: Visual analysis of clustered cell trajectories. The clustered trajectories are visualized in a 3D-window (**A**) while a composite view shows the shape and structure comparison between pairs of trajectories within the same cluster (**B**). Additionally, a color-coded lineage diagram (**C**) and a dendrogram (**D**) are generated encoding cluster properties. Next to the visualizations, certain cluster properties and interaction possibilities are given (**E**).

Trajectory Clusters (Figures 5.8A and B): The clustered trajectories are analyzed in a 2D/3D visualization. This is required in order to analyze the spatial developments of cells over time

such as division properties. Such a visualization allows an intuitive comparison of different cell developments in space. The trajectories are realized as cylinders and spherical nodes colored based on their cluster membership. For this, I use a discrete qualitative color map consisting of eights colors that is generated with *ColorBrewer* (http://colorbrewer2.org/). The cluster color assignment depends on the order in which the elements are merged. In order to minimize occlusion problems and visual clutter, the rendering of trajectories can be interactively turned on and off. Individually, the structural comparison of pairs of trajectories (based on their alignments illustrated in Figure 5.5B on page 63) is analyzed. By this, the user can immediately validate the geometrical similarities in more detail.

Color-coded Lineage Trees (Figure 5.8C): The cell developments are depicted in a lineage diagram for which the cell paths of the corresponding cell trajectories are colored based on the clustering results. This allows a direct comparison of clustered similarities between a set of cell paths in lineage trees. Additional cell lineage information are given on top of each tree (ID, number of divisions, number of cell paths).

Dendrogram (Figure 5.8D): A tree diagram called dendrogram illustrates the arrangement and order of the generated cluster hierarchies. As described above, this visualization gives important feedback about the clustering results and about which cluster parameters are suited best for the data sets. Both the correct number of clusters and the cluster stop threshold are unknown when analyzing new data sets. The dendrogram provides a visual representation to verify the selection of both values depending on the data.

5.4 Application Results

I apply the clustering method to the zebrafish data sets and to all Arabidopsis data sets. Next to the quantitative results of the weight parameters in Section 5.3.2, the visual analysis yields new biological insights. These are the detection of similar collective cell migrations in dense regions and the detection of an hitherto unknown correlation between the orientation of trajectories and division types. For the clustering of the Arabidopsis data, the average linkage criterion is used while for the zebrafish data sets, Ward's method is applied.

Epiboly Data

The tracking results in $4{,}896$ lineage trees. However, only few of them contain a complete track of a cell development over the whole recorded time step range due to the low segmentation and tracking quality. Most of the trees are incomplete and do not contain a sufficient amount of tracking information. For this reason, I consider cell lineages that contain most of the tracked cells and therefore almost complete cell developments. In order to filter, only trajectories in trees are considered with at least 8 and at most 64 divisions. Additionally, trajectories with a length smaller than five are ignored and only the first 65 time steps are regarded. In later time steps, the lineage trees are too error-prone. This filtering results in a total of $1{,}132$ trajectories of 37 lineage trees.

I first apply a clustering based on the delta time property in order to investigate the durations of cell cycles. For this purpose, $\lambda_{\{1,2,4\}} = 0$ and $\lambda_3 = 1$, no LOD adaption is used and five clusters are generated because within the first 65 time steps up to five cell cycles after the first division are recognized. In a lineage tree, the cell root path (path that starts at the root node)

Figure 5.9: Lineage diagram colored by clustered cell paths based on delta time for the epiboly data set.

is defined as the zeroth cell cycle which is not further considered in the analysis. Because of the starting time of the data record it cannot be verified if the cell root path is fully captured. Figure 5.9 shows the lineage diagram colored according to the cluster assignments. Nearly all cell division paths (paths between two subsequent divisions) of the first two cell cycles are assigned to the violet cluster. These cell branches correspond to symmetric divisions of stem cells. The two daughter cells have the same stem cell properties for the first and second cell cycle. In the next one, based on the yellow cluster assignment of cell paths, asymmetric divisions occur that result in a stem and a progenitor cell. The latter one divides faster than stem cells and this behavior explains the diversity of the cluster colors (orange and yellow) in cell cycles three and four. Cluster blue includes the longest cell paths for which at least one tracked cell division is missing. The green cluster includes the shortest cell leaf paths (paths that end in a leaf node) that are pruned because of the fixed considered time range. For this cluster setting, the visual analysis of the 3D cluster results does not reveal any spatial pattern and the trajectories are uniformly distributed all over the embryo. It is assumed that this observation is due to the epiboly phase in which the first recognized cell migration occurs, the cells are spread uniformly and divide quickly while forming the embryonic axis.

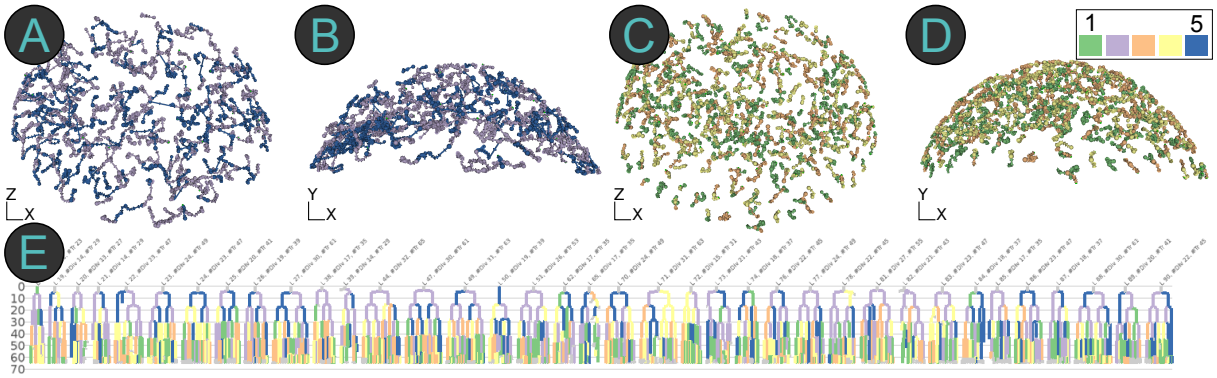


Figure 5.10: Cluster results based on shape for the epiboly data set. The figure shows the trajectories from the animal (**A, C**) and side view (**B, D**) as well as the lineage trees colored based on the cluster assignments (**E**).

In order to investigate the shape structure of cell developments, weight parameters of $\lambda_4 = 1, \lambda_{\{1,2,3\}} = 0$ are chosen. Figure 5.10 shows the 3D clustering results and the lineage diagram. Almost all trajectories of the first and second cell cycle are either assigned to the violet or blue cluster. The visual analysis in 3D reveals that both clusters consist of trajectories that feature similar long lengths and shapes located all over the animal pole of the embryo. This result also confirms the previous observations that cells in the first two cell cycles are similar in

the their temporal development. The remaining trajectories in clusters yellow, green and orange share similar short lengths and structures but no common spatial pattern can be identified in the lineage diagram or 3D visualization. Note that using the lowest LOD, a cluster analysis of the trajectory lengths is enabled when clustering with regard to shape because the coupling distance takes the length into account. The analysis of the lengths could be realized more simply by inserting an additional distance measure in the similarity analysis. However, this would also mean that an additional weight parameter had to be considered but the length information is already integrated in the shape comparison of the coupling distance. Thus, I use this measurement to analyze the lengths. The visual analyses based on lengths and velocities ($\lambda_2 = 1, \lambda_{\{1,3,4\}} = 0$) are alike and indicate that cells migrate with constant and continuous speed. This behavior is already assumed based on the quantitative results in Table 5.5 on page 75. Furthermore, it can be observed that again cell division paths in the first two cell cycles share similar lengths and velocities. However, the investigation of the orientations ($\lambda_1 = 1, \lambda_{\{2,3,4\}} = 0$) delivers no regions of similar migration patterns. The epiboly data features lots of cell divisions with arbitrary directions and this behavior allows no detection of collective migrations.

In summary, in the epiboly data set, the trajectories of the first two cell cycles share similar shape, duration, length and velocity properties. At later time steps more asymmetric divisions occur resulting in stem and progenitor cells. The data becomes more dynamic and a homogeneous migration behavior all over the embryo can be identified. However, the visual analysis does not support finding patterns in later time steps. Because of the high dynamic of several cell divisions, similar cell developments are distributed equally among the embryo. Using clustering can in this context also be used to identify outliers in the data. For example, in exceedingly long cell trajectories, division events are missing while strikingly short trajectories indicate cell root or cell leaf paths. For particularly short cell division paths it is assumed that subsequent cell divisions are tracked incorrectly.

Tailbud Data

The tailbud data set contains $58{,}048$ tracked cell lineages. This data set features more cell migrations and less cell divisions in comparison to the previous data set and it is more challenging due to the higher density of cells than in the early stages of the epiboly. In the raw data set of the tailbud, some cell information is missing due to data loss. This is the reason why in the mid area of the tail a conspicuous gap is observed. Because of the error sensitivity of the tracking algorithm, lots of small lineage trees containing only few nodes are generated. Thus, I consider trajectories that have a length of at least 30 in order to ignore short outliers. By this, the complete data set and all time steps can be analyzed. $1{,}331$ trajectories are generated that feature $51{,}432$ sub-trajectories in $1{,}314$ cell lineages. I choose six clusters to gather different stages of the cell migrations towards the front tail.

I apply the clustering algorithm based on orientation with weight parameters $\lambda_1 = 1$ and $\lambda_{\{2,3,4\}} = 0$. Using the lowest LOD enables the investigation of main trajectory directions and trends. Figure 5.11 on page 82 shows the cluster results from the side (Figures 5.11A and C) and top view (Figures 5.11B and D) of the tail. The cells migrate from right to left differing slightly with regard to their orientations. These are clustered into three groups (blue, yellow and green) and represent the main tail growth. The red and orange trajectories in the mid represent dense regions of similar cells migrating to the front tail (Figure 5.11D) or ingoing side (Figure 5.11C). It is assumed that these cell migrations supply and push new cells into the main growth direction

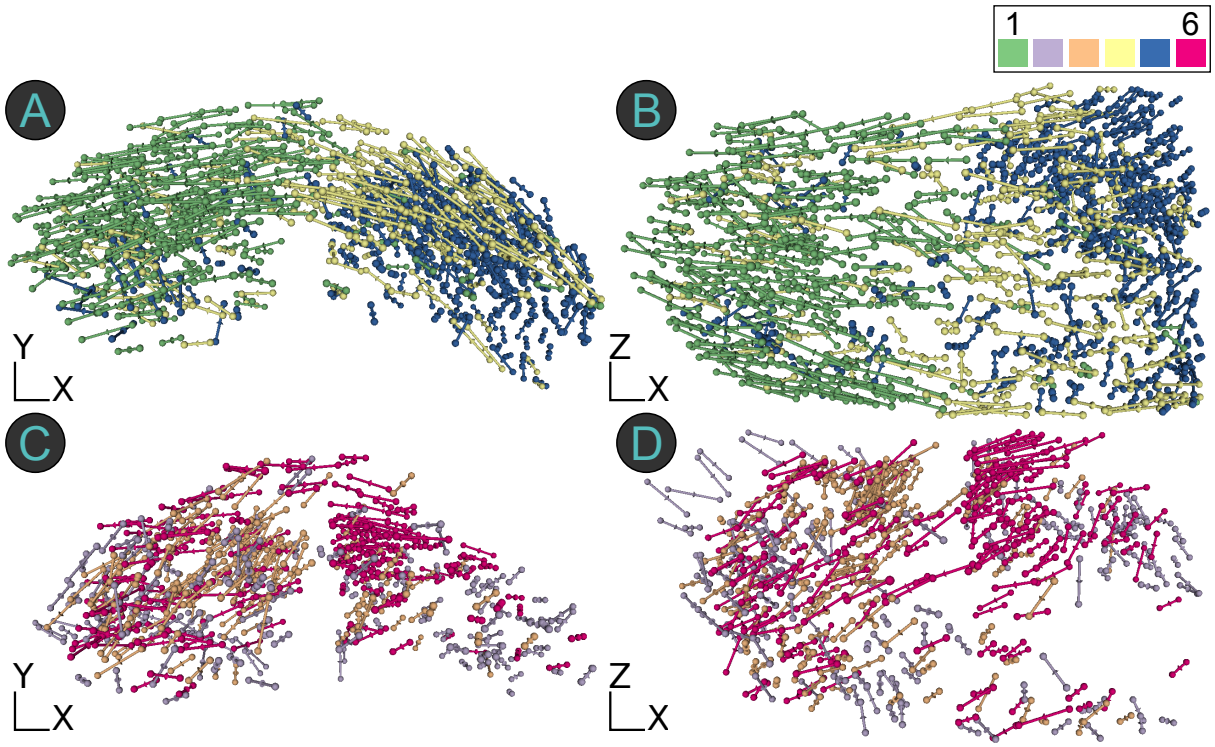


Figure 5.11: Cluster results based on orientation for the tailbud data set. The images show the tailbud from a side (**A, C**) and top view (**B, D**).

towards the front tail. The trajectories in the remaining violet cluster are distributed all over the tail and highlight cell developments against the principal growth direction which might be erroneous cell tracks.

A visual analysis based on velocities ($\lambda_2 = 1, \lambda_{\{1,3,4\}} = 0$) reveals a cluster (blue) of the fastest trajectories that mainly occur at the periphery, i.e. the outer part of the tail and at the tailbud (Figures 5.12A and B on page 83). Three other clusters (violet, yellow, and green) share almost the same velocities (Figure 5.12C) and dominant appearances are located at the tailbud (Figure 5.12D). It is assumed that the cells at the front tail feature a fast growth while the migrations at the right push the cell migrations towards the principal growth direction. The last two clusters (orange and red) contain the slowest cell migrations for which a dense region at the top right is prominent (Figures 5.12E and F). Cluster analyses based on the length and delta time spans (cell cycle durations) show similar results of long cell trajectories at the periphery. Hence, the cells are moving there consistently without abrupt changes in their velocities.

As a next analysis step, the shape structure ($\lambda_4 = 1, \lambda_{\{1,2,3\}} = 0$) without LOD reduction is investigated. I choose eight clusters in order to provide more flexibility for the cluster algorithm to separate the trajectories. In Figure 5.13 on page 84, four similar cluster sets can be distinguished. Similar to the previous results, in Figures 5.13A and B, the main trend of collective cell migrations at the periphery and at the front tail is identified. Furthermore, a dense region of short cell migrations at the top right of the tail is detected (Figures 5.13C and D). The two clusters in Figures 5.13E and F allow a separation of similar trajectories into a left and right region for which the left one is more dominant with more cell activities. The remaining three

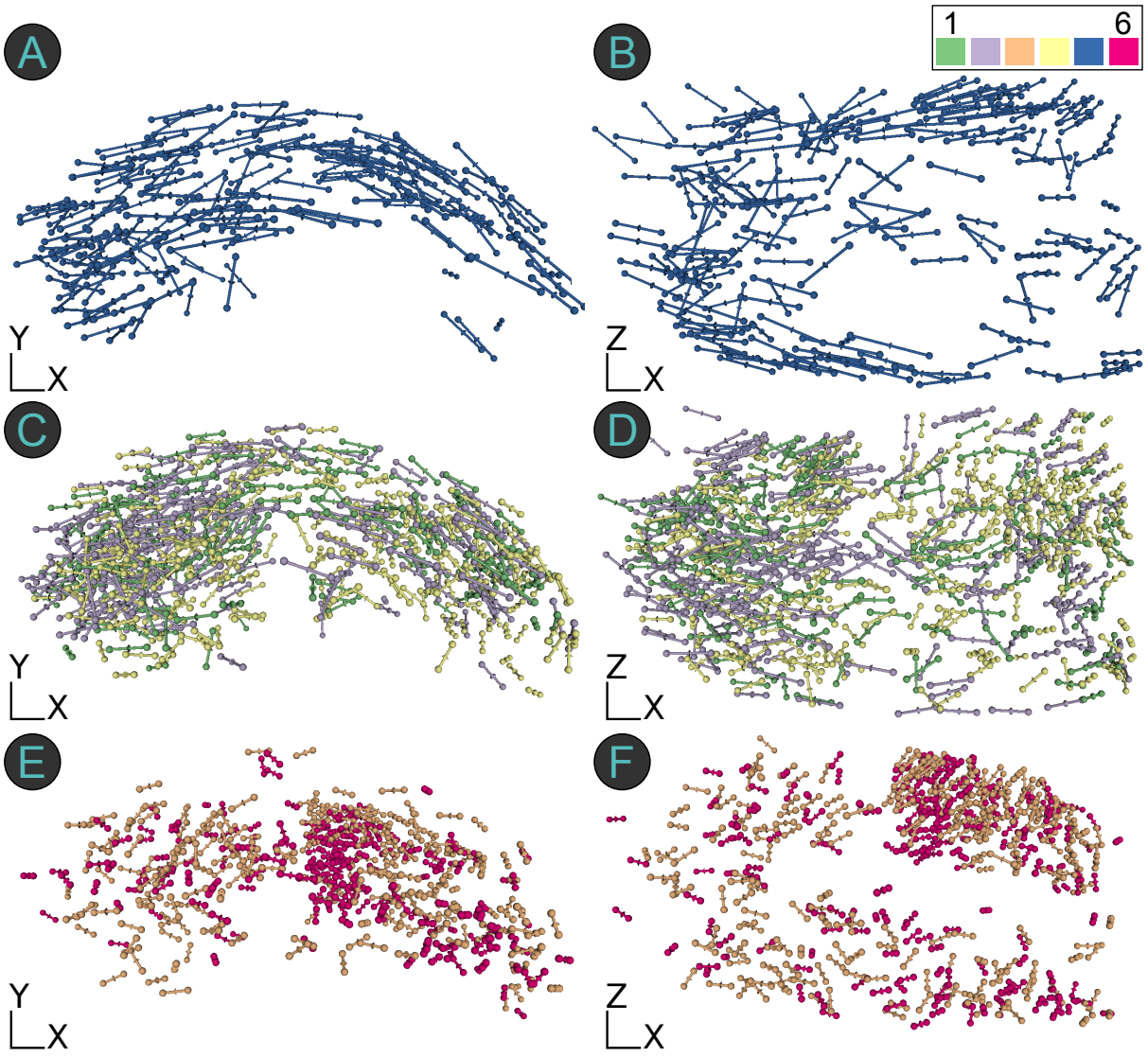


Figure 5.12: Cluster results based on velocity for the tailbud data set.

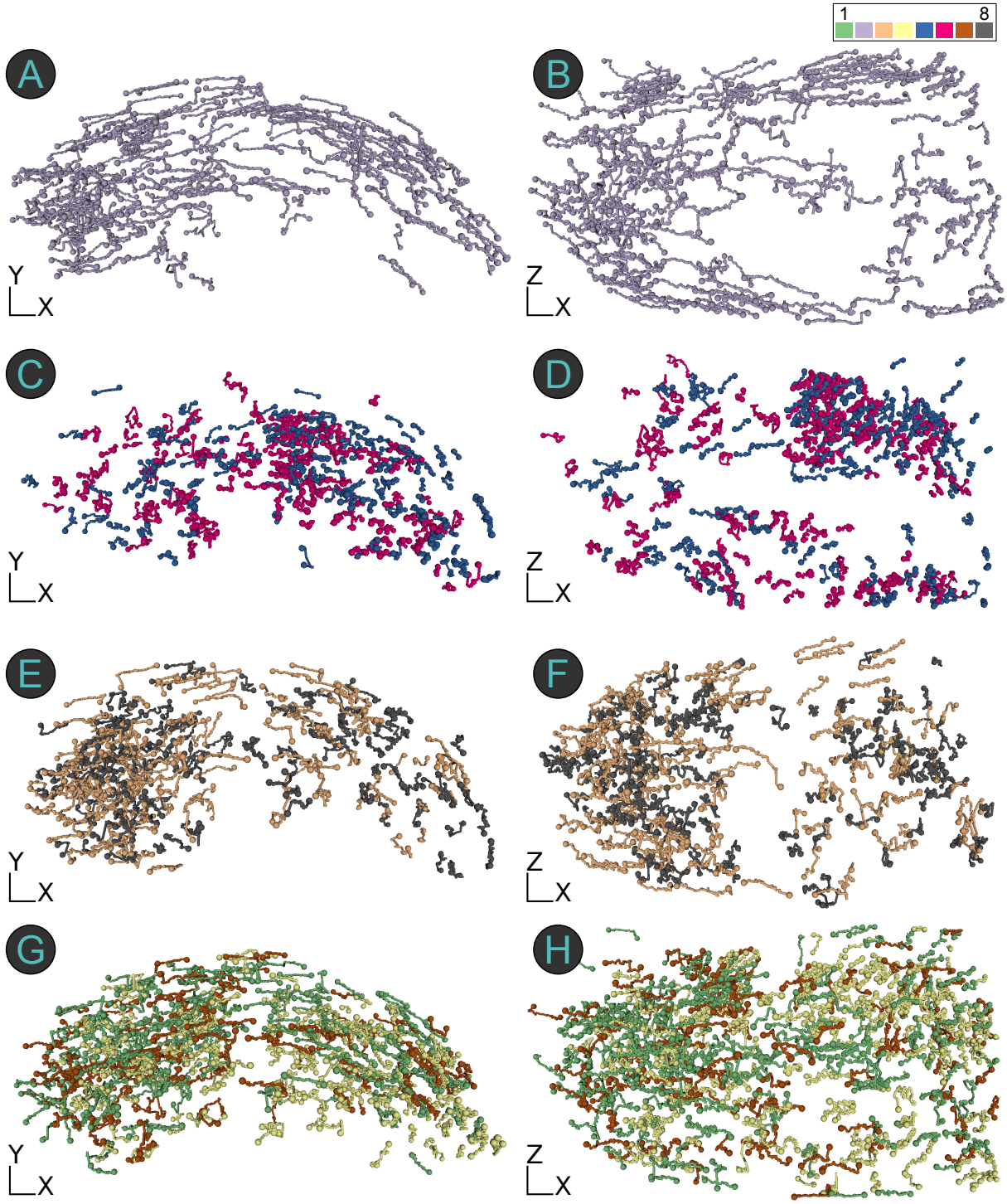


Figure 5.13: Cluster results based on shape for the tailbud data set. In this case, eight clusters are chosen for investigating the shape of the trajectories.

Data set	Divisions	Cluster α			Cluster β		
		A	P	R	A	P	R
120830	166	79	56	15	65	80	37
120830 (LOD)	166	112	24	11	32	114	39
121204	156	109	40	13	57	68	25
121204 (LOD)	156	119	24	7	49	86	27
121211	242	102	75	36	70	149	62
121211 (LOD)	242	104	74	32	68	140	66
130508	134	71	52	15	31	60	39
130508 (LOD)	134	77	15	8	25	97	46
130607	252	140	88	30	80	100	66
130607 (LOD)	252	116	70	26	104	108	70

Table 5.7: Cluster results of Arabidopsis data focused on division types within clusters. Cluster α represents the combination of the clusters that mainly includes trajectories starting with anticlinal divisions while cluster β contains more trajectories evolving from periclinal and radial divisions. The highest number of such divisions between the two clusters are indicated by the three colors: red (anticlinal), green (periclinal), and blue (radial).

clusters (Figures 5.13G and H) include similar trajectories that are distributed all over the tail expressing the diversity of numerous cell migrations during the tail growth.

In summary, the cluster analysis of the trajectories for the tailbud data reveals two important insights. First, similar tendencies of collective cell migrations are detected based on all applied parameters: their orientation, velocity, shape and duration. Most of the prominent events are located at the periphery and at the front of the tail. Second, dense regions of similar cell behavior are detected that further push cells into the direction of the tail growth.

Arabidopsis Data

For the visual analysis of the Arabidopsis data, I extend the visualization by considering the division type information that was determined by the automatic classification algorithm in chapter 4. These division types (anticlinal, periclinal and radial) are based on the orientations of divisions with respect to the generated isosurfaces (see Section 4.2.2 for more details). Note that there are no cell movements in plants but only nuclei displacements. Thus, the cell trajectories refer to the nuclei migrations. I intend to visually analyze if there is a correlation between these division types and the orientations and shapes of trajectories using the clustering algorithm. Based on the principal growth direction of the lateral root in height (periclinal division) and in length (anticlinal division) four clusters are generated. This number is chosen in such a way that a pair of opposing nuclei migrations is captured. For this purpose, $\lambda_1 = 0.8, \lambda_{\{2,3\}} = 0$ and $\lambda_4 = 0.2$ are chosen to focus mainly on the orientation and shape information. Initially,

I apply the lowest LOD to the trajectories focusing on the nuclei directions and their lengths. Note that the segmentation and tracking are done manually for the different plants. Thus, the cell lineages have a high quality without outliers. Consequently, all trees and all time steps are considered without ignoring any trajectories of a specific length.

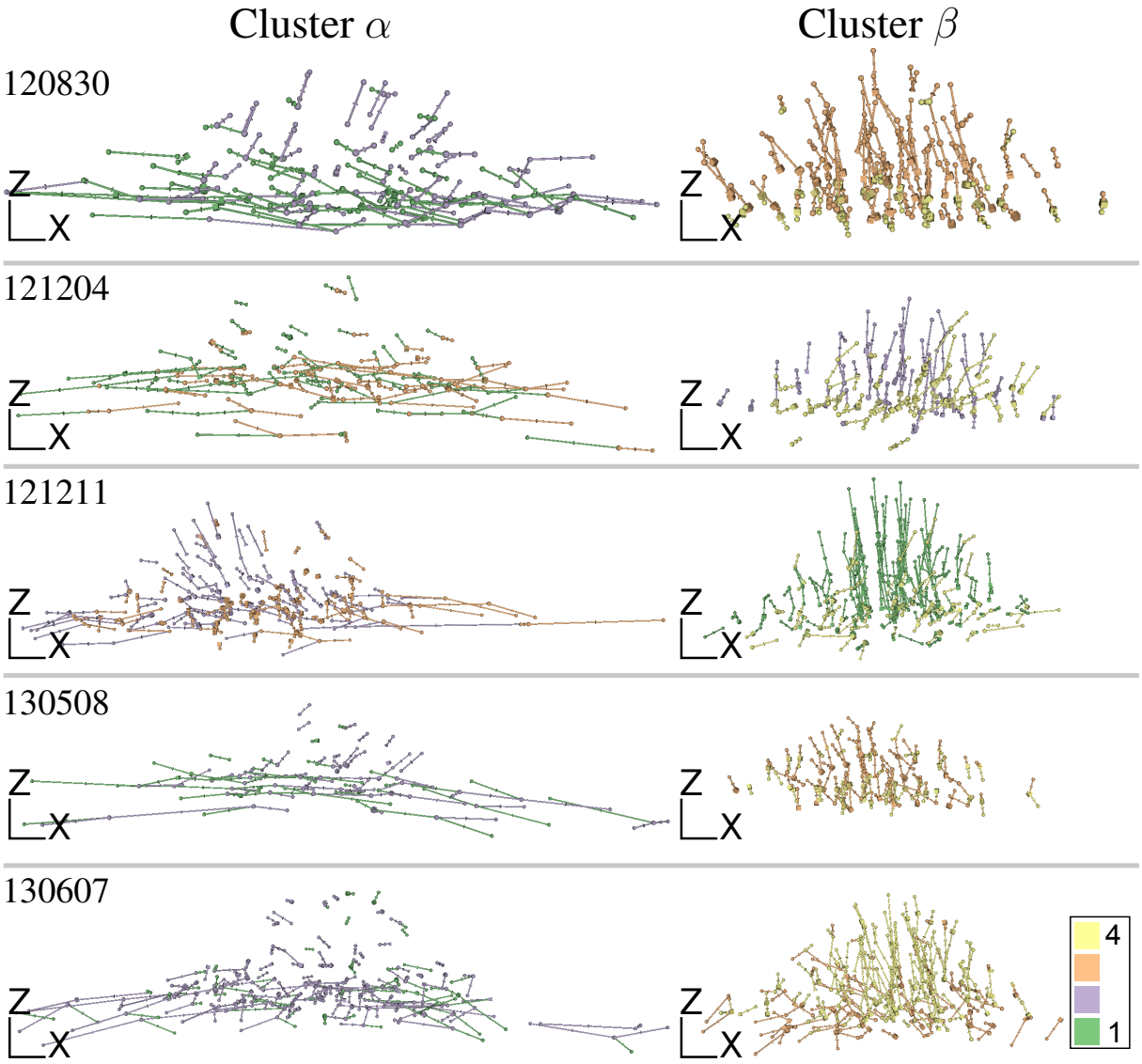


Figure 5.14: Visual cluster results of the Arabidopsis data sets based on orientation and shape. For each lateral root growth, four clusters are generated with weight parameters of $\lambda_1 = 0.8, \lambda_{\{2,3\}} = 0, \lambda_4 = 0.2$.

Figure 5.14 shows the clustering results for all five lateral root growths of the Arabidopsis plants. Two different main trends of trajectories can be observed. On the left side, two clusters are shown that correspond mainly to trajectories oriented almost parallel to the x-axis. This combination is referred to as cluster α. On the right side, for each root growth the remaining two clusters correspond to trajectories perpendicular to the x-axis. This combination is referred to as cluster β. Table 5.7 on page 85 lists the corresponding division types for the trajectories

in each cluster. Based on the numbers of divisions (colored cells in table) in the clusters α and β, the trajectories of cluster α mainly correspond to trajectories evolving from an anticlinal division. Cluster β mostly includes trajectories that start with a periclinal and radial division, thus increasing the height and width of the dome-like structure. The number of anticlinal divisions dominate in cluster α (highlighted in red color) while there are more periclinal and radial divisions in cluster β (highlighted in green and blue).

This behavior is also observed when the trajectories are clustered using the highest LOD, i.e. the actual shape of trajectories is considered in the similarity analysis (Table 5.7). Note that the number of divisions for a data set is always half of the sum of all divisions in cluster α and β (for example for 120830: $166 \cdot 2 = 79 + 56 + 15 + 65 + 80 + 37$). This is caused by the fact that for each dividing cell two trajectories are generated inheriting the same type of division.

As a result, based on the dominant numbers of division types, if a cell divides anticlinally then its two daughter cells keep on moving into the same direction (cluster α). Analogously, trajectories that evolve from periclinal and radial divisions, tend to move into the same division directions. Note that one could have expected that a third cluster could be generated that only includes trajectories that start with a radial division, while cluster β only consists of periclinal divisions. However, radial divisions are not clearly separable from anticlinal and periclinal divisions and sometimes even a manual identification of a radial division is not correct. Although radial divisions do not contribute to the height of the primordium, they are merged with periclinal trajectories into the same cluster β. This is because their directions are more oriented in periclinal direction. It is assumed that this is a consequence of the growing behavior of the lateral root pressing cells to the top forming the dome. A clustering only based on orientations yields similar results like in Figure 5.14. However, the usage of additional shape information significantly improves the cluster assignment of trajectories. However, in contrast, a clustering only based on shape information fails to reveal a correlation and therefore a separation between the different division types.

5.5 Summary

In this chapter, I introduced a visual similarity analysis method for 3D cell trajectories. A level of detail technique is used to simplify the visualization and to focus on main tendencies of collective cell migrations. The similarity measure is based on the weighted combination of migratory and geometrical features in a hierarchical clustering approach. It is capable of automatically discerning and highlighting major differences in shape structures and movement-based properties of trajectories. The weight parameters λ_i permit users to influence the clustering with respect to domain-specific biological knowledge. The robustness of the resulting clusters with regard to changing parameters was discussed. As a result, the cluster stability depends on the quality and trajectory properties of the data sets. I demonstrated the practicability of the clustering method using two experimental data sets recording zebrafish embryogenesis and five different lateral root growths of Arabidopsis plants.

For the zebrafish, similar and dense regions of cell migrations are identified that share the same shape and orientation. Moreover, trajectories belonging to a specific cell cycle also share geometrical similarities in migration and cell cycle length. The clustering approach can also be used to identify cell trajectories that are biologically implausible, e.g. too long or too short cell migrations. For the Arabidopsis data sets, a novel correlation between division types

and subsequent orientations of trajectories is detected. More precisely, almost all trajectories starting with a certain division type and orientation keep on moving into the same direction until the next division occurs. This insight supports the hypothesis that a cell's fate and its further development is influenced by its division type.

While this method allows a similarity analysis for 3D+t cell trajectories, a new visualization method is required to analyze thousands of associated cell lineages. In the next chapter, I introduce a new visual analysis method called the structure map that focuses on the similarity analysis of a huge collection of tree structures.

Chapter 6

Visual Analysis of Large Cell Path Collections

"The Grid. A digital frontier. I tried to picture clusters of information as they moved through the computer. What did they look like? Ships? Motorcycles? Were the circuits like freeways? I kept dreaming of a world I thought I'd never see. And then, one day... ."
— Kevin Flynn, Tron Legacy, 2010

Developmental biologists analyze the process of how embryos develop from single cells into complete organisms. During this *embryogenesis*, patterns in cell migrations and divisions are believed to play a crucial role in determining cell organization into tissue and organs. In the last chapter, cell trajectories are clustered based on user-selected values for features such as orientation, velocity, cell cycle length and shape structure. However, the clustering results strongly depend on the quality of the segmentation and tracking. The latter one tends to result in many error-prone cell lineage trees. These may contain cell events that are biologically implausible. Due to technical reasons, data acquisition of single cells cannot start until multiple dozens of cells have already been developed using the light-sheet microscope [KSWS08]. This means that for each of these cells a lineage tree is generated in contrast to the initial cell (*zygote*) development that would result in one cell lineage for the whole *embryogenesis*. Furthermore, noise in the raw data leads in the automatic segmentation process to the detection of non-cell artifacts. These errors are further inherited by the tracking process and yield wrong biological behavior in the lineage trees. For example, the event of subsequent cell divisions that occur too fast or missing division events for exceedingly long cell migrations are indicators for implausible biological developments. In order to detect these manifold structures an interactive similarity analysis of cell lineages is required. Through this, erroneous behavior can be identified and excluded from the analysis to interpret actual biological events. In addition, the exploration of errors in lineage trees can help improve the data acquisition process to maximize its quality.

In this chapter, I present a novel visualization method called the *structure map* [FHR+15]. This map enables the comparison and highlighting of cell paths and cell branches in hundreds to thousands of trees that share similar patterns. The structure map is a matrix-based, color-

coded 2D grid that arranges trees into tiles along a *Hilbert curve*. Prior to the arrangement of trees, the similarity between trees based on *spectral analysis* is computed and trees are sorted using *principal component analysis*. The interactive visual analysis supports both, a global observation of complete cell lineages and a local investigation of highlighted cell paths and branches according to user-selected tree descriptors. The compact and uniform representation of the map supports domain experts in the identification of similar substructures in thousands of trees and in the detection of outliers. Moreover, it permits the comparison of various cell lineage trees among several data sets to investigate similar biological cell developments. I apply the structure map to the zebrafish data sets and the five lateral root growths of the Arabidopsis plants to demonstrate the benefit of the map. For the zebrafish data, the structure map helps to find structural differences and erroneous cell behaviors. A supplemental video [FHR+14] demonstrates these results and the main functionalities of the map. Furthermore, I use the map to compare similarities of all Arabidopsis data sets with each other. Through this, similar appearances of certain features are detected especially in the master cell file (see Section 4.1) of all plants.

6.1 The Structure Map

The structure map is an interactive visual analysis method that groups trees according to their structural similarity. Figure 6.1 illustrates the steps for generating the map and its application in the visual analysis. In the following, I describe each underlying method in detail followed by an explanation of the chosen tree descriptors. Afterwards, the complete algorithm for creating the map is presented with a performance analysis. In a final step, the visual exploratory methods of the map are defined prior to the presentation of application results.

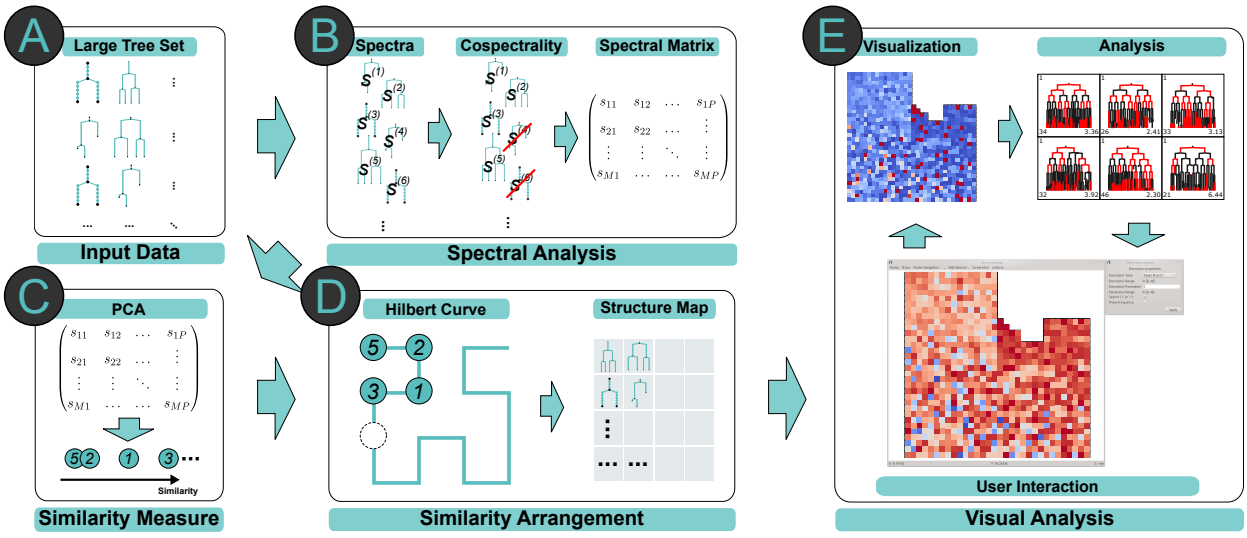


Figure 6.1: Workflow for generating the structure map. **A** A large set of lineage trees is selected as input data. **B** *Spectral analysis* of the tree data is used to derive structural similarity and to merge isomorphic trees. **C** To transform the spectral matrix S to a lower dimension, *principal component analysis* is applied. **D** The intrinsically 1D data is laid out in the plane along a *Hilbert curve*. **E** The visual analysis is a loop consisting of the visualization of color-coded tree descriptors in the structure map (Section 6.1.3), the analysis and interpretation of the result (Section 6.2).

6.1.1 Similarity Measure based on Spectral analysis

In order to group a large set of cell lineages a similarity measure is required. The most common technique for computing the similarity between trees is the *tree edit distance* [Tai79, Bil05]. This distance between two labeled ordered trees is defined as the minimum number of edit operations, i.e. insertions, deletions, and modifications, that are required to transform one tree into another. However, the straightforward implementation has an exponential complexity and the algorithm requires a labeling. For the structure map, I apply a similarity measure between cell lineages based on *spectral analysis*. This approach is used in many areas of computer science [ACSv12] and analyzes a graph or tree based on the spectrum s_i, i.e. the set of eigenvalues associated with a matrix. I choose this measure because of two reasons: First, the spectrum is a graph invariant that refers to the structure of a graph or tree and not on any labels or layout. Second, isomorphic graphs, i.e. graphs that are identical up to symmetry, share the same spectrum and it is known that small changes in a graph result in small differences in its spectrum, called *interlacing theorem* [Hae95]. Thus, this spectrum can be used to identify unique cell lineage trees. There are many matrices that can be used for the analysis of a graph (N, E) of a set of nodes N and edges E. Examples are the adjacency matrix A, the Laplacian $L = D - A$ or the signless Laplacian $|L| = D + A$. D is a diagonal matrix of node degrees. I choose the *normalized Laplacian matrix* $\overline{L_{ij}}$:

$$\overline{L_{ij}} = \begin{cases} 1 & \text{if } i = j \text{ and } d_i \neq 0 \\ -(d_i d_j)^{-\frac{1}{2}} & \text{if } \{i, j\} \in E \text{ and } d_i, d_j \neq 0 \\ 0 & \text{otherwise} \end{cases} . \tag{6.1}$$

This matrix contains structural properties as well as connectivity information of the graph or tree. Arsić et al. [ACSv12] give an overview of the properties of its eigensystem. Two trees i, j are said to be *cospectral* [WZ08] if they share the same spectrum, i.e. $s_i = s_j$ sorted in descending order. There are upper bounds for the uniqueness of a tree spectrum [WZ08, ME12]. Schwenk [Sch73] assumes that almost all trees are cospectral. But his study shows that the probability of cospectral trees is going to 1 only appears as the number of vertices goes to infinity for randomly chosen trees that are not used in practice. In another experiment, Matsen and Evans [ME12] show that the fraction of binary trees with unique spectrum goes to zero as the number of leaves goes to infinity. However, they observe that this convergence is very slow and in their experiments with more than $50{,}000$ randomly chosen trees with unique spectra, less than 0.14% of these trees are not uniquely identified by their spectra. This means that if two spectra are identical then the corresponding trees are not necessarily isomorphic. But two isomorphic trees share the same spectrum. Especially for the cell lineage data from

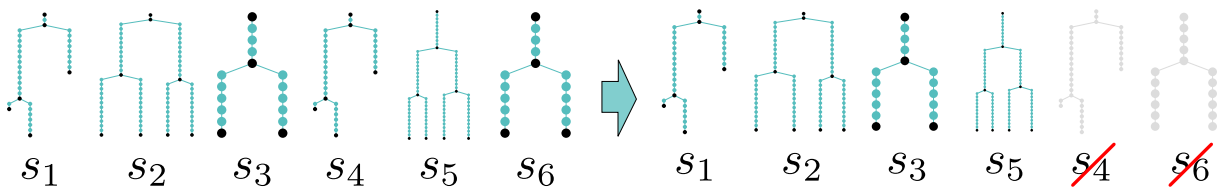


Figure 6.2: Handling cospectral trees: In this example of six trees, two pairs of cospectral trees (s_1, s_4 and s_3, s_6) are identified for which only one representative tree is further considered in the analysis.

developmental biology, it is unlikely that trees with the same spectrum are non-isomorphic which is verified by applying a tree isomorphism test [Bus97] to separate isomorphic trees into equivalence classes. Consequently, for the cell lineages in this thesis, trees are assigned to the same equivalence class if they share the same spectrum. Figure 6.2 illustrates an example of six trees for which two pairs of trees are cospectral. The check for isomorphic trees results in a lossless compression of the input data set and consequently in the visual analysis of it. For the zebrafish data, compression rates between 82% (epiboly) and 95% (tailbud) are observed. The cell lineage trees of the Arabidopsis plants are not compressed at all because of the higher quality due to manual segmentation and tracking. The similarity between two trees is realized as the Euclidean distance of their spectra and referred to as the *spectral distance* [PP09, Cve12]:

$$d_S\left(s_i, s_j\right) = \sqrt{\sum_{k=1}^{n} \left(s_{i_k} - s_{j_k}\right)^2}, \tag{6.2}$$

For each isomorphic tree i the spectrum $s_i \in \mathbb{R}^P$ is stored as a row in a spectral matrix $S \in \mathbb{R}^{M \times P}$:

$$S = \begin{pmatrix} s_{11} & s_{12} & \dots & s_{1P} \\ s_{21} & s_{22} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ s_{M1} & \dots & \dots & s_{MP} \end{pmatrix}. \tag{6.3}$$

M is the number of isomorphic trees and P is the maximal number of nodes regarding all trees. The number of nodes of a tree is identical to the number of eigenvalues. For different dimensions each spectrum is padded with zeros in such a way that all share the same dimension P. Wilson and Zhu [WZ08] show that adding an isolated node in the graph yields an additional zero eigenvalue, but preserves the other eigenvalues and does not change the connectivity properties.

6.1.2 Layout of Tree Collection

There a lot of approaches to visualize and layout trees. These are commonly divided into three groups: *Space-filling*, *node-link-based*, and *hybrid* approaches. Space-filling techniques use the complete area of a display in order to illustrate hierarchies in a tree. Here, the information of relationships is given by enclosure (e.g. *treemaps* [Shn92] and *squarified treemaps* [BHvW99]), adjacency (e.g. *SunBurst* [SZ00] and *icicle plots*) or crossings (e.g. *beamtrees* [vHvW03]). Quantitative properties can be given either by the area size, color or height of items. The main advantage of treemaps is the optimal space usage in which child nodes are enclosed within a parent node. However, the overlapping in the parent node can yield to interpretation problems of the hierarchy structure. Adjacency space-filling methods do not suffer from overlapping issues but the advantage of optimal space usage is lost. Crossing techniques only partially allow overlap and adjacency information. But they are complicated to read and to interpret. Node-link-based techniques use links between nodes to represent their relationships. Common layouts are set horizontally, radially, or in balloon form in 2D [HMM00]. Other layouts are *Cone trees* [RMC91], *point based trees* [SHS09], *Phyllotrees* [NCA06] and *hyperbolic layouts* [Mun97, AH98]. Although these visualizations provide an intuitive interpretation of hier-

archy information they often do not scale to larger collections of trees. This results in a more complex overview and complicates the user interaction. Hybrid approaches combine node-link-based techniques with treemaps and visualize subsets of the hierarchy information [ZMC05].

In the structure map, the layout is realized in a hybrid approach by using a space-filling *Hilbert curve* with no overlapping problems and a linear ordering of node-link rendered trees. More precisely, the ordering of the lineage trees satisfies the clear separation between large trees with many nodes and small trees with few nodes and the node-link based approach yields an intuitive representation of the cell developments. For this purpose, I apply a 1-dimensional *principal component analysis* (PCA) [Jol02] to the spectral matrix S. PCA transforms a data set into a variance-maximizing coordinate system of linearly uncorrelated combinations called *principal components*. This 1-dimensional embedding already accounts for more than 90% of the variance for the zebrafish data and 85% for the Arabidopsis plants. This means that the ordering is well-suited for sorting the cell lineages. Note that PCA is based on the Euclidean distance which corresponds to the spectral distance used to compute the difference between spectra. The ordering result of PCA is used to align the sorted trees along the space-filling Hilbert curve. This curve (Figure 6.3 for several iterations) is a continuous fractal curve whose range is a 2-dimensional square. I choose this method for the arrangement of trees because points that are near when traversing the curve are also likely to be close in the embedded space of the curve, thus it preserves locality in a compact visualization. Furthermore, any overlapping issues are avoided because the structure map is a grid of tiles in which each tile represents a single isomorphic tree. In practice, I choose the number of iterations in such a way that the

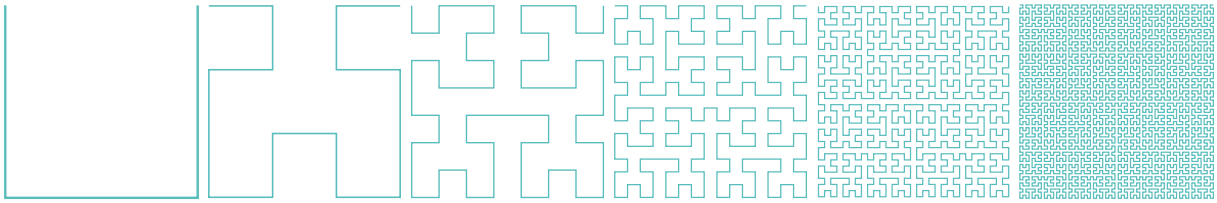


Figure 6.3: The first six iterations of the Hilbert curve.

curve contains all trees in a data set. However, the structure map may not use the complete screen space in general. For example, the layout of $2^{12} + 1 = 4{,}097$ trees requires a Hilbert curve with $2^{13} = 8{,}192$ nodes, resulting in almost 50% empty tiles. But in practice, the empty tiles do not impede the visual analysis of tree structures in any way. In the concluding chapter, I address this issue in more detail and give ideas for improvement.

Tile Design

Each tile is a colored square that represents a single isomorphism class of the trees. This single tree is drawn in white on top of the tile using the Reingold-Tilford algorithm [RT81]. This drawing style is best-suited for domain experts to detect substructures and their patterns. I apply an adaptive anisotropic scaling method to each tree in such a way that the tree structure fits completely into the tile. Cell paths or cell branches in a tree that match a given tree descriptor are highlighted in red. Furthermore, labels are used to represent relevant information of the corresponding tree this tile represents. The tiles are also colored based on that value. The right image of Figure 6.5 on page 98 shows an example of such a tile.

6.1.3 Tree Descriptors

In fact, arbitrary descriptors can be defined according to the requirements of domain experts. Navigli and Lapata [NL07] survey common descriptors for developing unsupervised algorithms for graph analysis such as *betweenness centrality* to measure how often a node is visited for all shortest paths or *compactness* to quantify how easy each node can be reached from other nodes. In discussions with developmental biologists, I define a set of tree descriptors suited for the similarity analysis and error detection of cell developments. These descriptors are defined with respect to the several types of cell path and cell branch definitions introduced in section 3.5. Each tree descriptor yields quantitative information about a tree and the tiles of the structure map encode this information through different colors.

Number of Nodes and Leaves: The number of nodes is positively correlated with the total size and depth of a binary tree. A tree with n nodes has exactly $n-1$ edges. Note that for biological data sets, the number of nodes corresponds to the number of tracked cells. Even without any further biological analysis, trees with a comparatively small number of nodes are more likely to contain erroneously tracked cells. Moreover, for the cell lineages, based on the number of leaves l, the number of cell divisions is $d = l - 1$.

Cell Path Length: The lengths of different cell paths within a cell lineage tree is an essential quantity for the similarity analysis because it corresponds to the duration of a cell cycle. The definition of a cell path is given in 3.1. To recap, a cell path p of length $l_p = |p| - 1, (l_p > 0)$ is a sequence of ordered connected pairs (n_i, t_i), i.e. $p = \{(n_1, t_1), \ldots, (n_{|p|}, t_{|p|})\}$. Note that the cell paths are distinguished between *cell root paths*, *cell division paths* and *cell leaf paths*. These differ in their start and end node type (Section 3.5). This information is used as a descriptor to find similar cell cycle durations or outliers. A tree usually contains multiple paths of different lengths. Hence, in order to quantify them, I compute the mean cell path length

$$\overline{p}_k = \frac{1}{|p|} \sum_{i=1}^{|p|} p_i, \tag{6.4}$$

where $|p|$ is the number of cell paths in the tree k and p_i denotes the i-th path in the tree.

Cell Division Path Length: By ignoring all cell root and cell leaf paths, I only consider the mean cell division path length $\overline{q}_k$ for a tree k, which is a variant of the previous measure $\overline{p}_k$. The mean cell division path length only counts cell cycles that have been fully captured. This length is used as a measure to identify two cell divisions that occur in quick succession, for example.

Cell Branch Asymmetry Length: The length of a pair of cell cycles evolving from the same division node also describes a biologically important substructure within a cell lineage tree. This structure in the tree is defined as a *cell branch* b_n (Section 3.5). Note that b_n is *symmetric* if the lengths of its both cell paths l_n and r_n are equal. For each branch b_n, I calculate the *cell branch asymmetry* as the difference in lengths of its two paths: $\Delta b_n = |l_n - r_n|$. From this, the mean branch asymmetry $\overline{b}_k$ of a tree is derived when there are several branches in a tree:

$$\overline{b}_k = \frac{1}{|b|} \sum_{n} \Delta b_n, \tag{6.5}$$

where $|b|$ denotes the number of branches in the tree and the sum ranges over all cell division nodes. If a tree does not have any branches, I assign $\overline{b}_k = -1$. Since Equation (6.5) always results in positive numbers, trees without branches are easily identified. A red tile color is used to encode such trees in the structure map in order to signify that they have maximum dissimilarity to all other trees with at least one branch.

Inner Cell Branch Asymmetry Length: The mean inner cell branch asymmetry length $\overline{d}_k$ of a tree k is a subset of the previous measure $\overline{b}_k$. For this descriptor, I only consider branches that consist of a pair of cell division paths, else the branch is not used in the analysis. Through this, only branches are investigated that are fully captured in the lineage structure.

Number of Division Types: When analyzing the Arabidopsis data, the number of different division types (anticlinal, periclinal, and radial) are of interest for domain experts. These are classified with the first visual analysis method in section 4.2.2. Through this, division schemes in several trees and among different lateral root growths can be compared.

6.1.4 Algorithm and Performance Analysis

The pseudo-code in Figure 6.4 illustrates the processing of the underlying methods in order to generate the structure map. For each cell lineage in the set C, the spectra s_l are computed (line 8) while updating P which represents the maximal dimension regarding all spectra. In the same for-loop, each spectrum is checked on cospectrality (line 11). If a tree structure is isomorphic, then its ID with an initial counter of one of its occurrence in the data set is stored in a map `Iso`. Else if the spectrum already exists, the counter of the corresponding cospectral tree already inserted into the map is increased by one. Through this, isomorphic trees are detected and their number of appearance is stored in `Iso`. When all trees are traversed, M denotes the number of isomorphic trees and the spectral matrix S is initialized with M rows and P columns (line 13). In a second for-loop over all isomorphic trees, the mean values of the descriptors are determined and stored in a map `Descriptor` of vectors for each unique tree ID (line 15). The vector holds all descriptor values explained in the previous section. The spectra are padded with zeros to ensure equal dimensionality P and stored in the spectral matrix S (line 17). This matrix is then used in the PCA to determine a sorted 1-dimensional embedding of the spectra (line 18). The ordering result `Order` is then used in the structure map to align the trees along a Hilbert curve (line 19).

The structure map is applied to all zebrafish and Arabidopsis data sets. (Hardware setting: Intel Core i7, 3.20 GHz, 12 GB of memory and an NVidia GTX 480). For the further analysis, I call the combination of the data sets 120830, 121204, 130508 and 130607 the Arabidopsis collection for which 121211 is excluded because the recording of this data set starts at a later time step. More information about this is given in the result section. In general, the generation of the structure map is realized by applying two dimension reduction methods using spectral analysis and PCA. The decomposition of the eigenvalues is realized using the library for linear algebra operations called *Eigen* (http://eigen.tuxfamily.org/). For all cell lineages $|C|$, the decomposition has a worst time complexity of $\mathcal{O}(\sum_{i=1}^{|C|} N_i^3)$ with N_i denoting the number of nodes of the tree i. Isomorphic trees with a total number of M are identified in linear time in the number of trees $|C|$. The mean values for the tree descriptors such as (inner) cell cycle length and cell branches are computed in linear time in the number of nodes N_i for each cell lineage (binary tree). The PCA has a time complexity of $\mathcal{O}(M^2P)$ for $M < P$ or $\mathcal{O}(P^2M)$ for $P \leq M$. This is

realized by a *two-sided Jacobi singular value decomposition (SVD)* in *Eigen*. This means that the time complexity is always linear in the greater dimension of the spectral matrix $S \in \mathbb{R}^{M \times P}$. The alignment using the Hilbert curve takes linear time in the number of isomorphic trees M. In total, the generation of the structure map has a cubic time complexity.

Input : Set of cell lineages C and number and types of descriptors.
Output: Structure map.

```
1  S = s(i, j) : matrix [1..M, 1..P] of real;
2  Iso(i) : map with i ∈ ℕ of unsigned integer;
3  Order(i) : vector with i ∈ [1..M] of unsigned integer;
4  Descriptor(i) : map with i ∈ ℕ of vector with j ∈ [1..numDescriptor] of real;
5  begin
6      P ← 0;
7      for l ← 1 to |C| do
8          s_l ← determineEigenValues(C_l);
9          if |s_l| > P then
10             P ← |s_l|;
11         Iso(l) ← checkCospectrality(C_l);
12     M ← |Iso|;
13     initSpectralMatrix(M, P);
14     for l ← 1 to M do
15         Descriptor(l) ← determineDescriptors(C_l);
16         padSpectraWithZeros(s_l, P);
17         S(l, :) ← s_l;
18     Order ← applyPCA(S);
19     alignLineagesInMap(Order, C);
```

Figure 6.4: Algorithm for generating the structure map.

Table 6.1 lists several properties such as the number of lineage trees and nodes for each investigated data set and the corresponding computation times in seconds. Because there is a high variance of the number of nodes, the minimum, maximum, and average numbers are given to get an impression of the lineage tree sizes. The computation times for the spectral analysis corresponds to the decomposition of the eigenvalues for all trees and the finding of cospectral trees. The times for the similarity measure in the last column include the mean value computations for the tree descriptors, the PCA and the alignment using the Hilbert curve. Except for the tailbud data, the spectral analysis takes the longest time with a maximum of approximately 9 hours for the Arabidopsis collection. In contrast, the tailbud data only requires 19 seconds. These long durations are caused by the large amount of trees with a huge number of nodes. For example, the tailbud data set features $3{,}221$ isomorphic trees but these have only 8 nodes on average with a maximum of 189 nodes. However, the Arabidopsis collection has on average $1{,}880$ nodes with a maximum of $5{,}164$ nodes. Although only 49 cell lineages are considered, the huge number of nodes is the reason of the long processing time for the decomposition of the eigenvalues. In order to avoid these long times, for new data sets the

Data set	Lineage trees (L)	Trees in map (M)	Number of nodes (N_i)			Computation times [s]	
			Min	Max (P)	Average	Spectral analysis	Similarity measure
Epiboly	4,896	875	1	2,323	25	3,347.22	12.37
Tailbud	58,048	3,221	1	189	8	19.79	10.87
120830	10	10	302	4,055	2,109	6,342.41	0.92
121204	15	15	306	3,577	1,324	3,743.78	0.86
121211	18	18	477	2,724	1,591	3,646.31	1.28
130508	9	9	621	5,164	2,123	11,939.20	0.85
130607	15	15	386	4,543	2,139	13,174.10	1.39
Arabidopsis collection	49	49	302	5,164	1,880	33,064.60	3.81

Table 6.1: Performance analysis of main tasks for generating the structure map. The last two columns are the computation times measured in seconds for the spectral analysis and the similarity measure.

spectra are computed only once for each tree structure and stored on the disk. This minimizes significantly the loading times to just a few seconds for each reprocessing of the structure map. The similarity measure (descriptor computation, PCA, Hilbert curve) requires much less time to be processed. In this case, the zebrafish data sets require the longest times of approximately 12 seconds because both share between hundreds and thousands of isomorphic cell lineages in contrast to the Arabidopsis collection with 49 trees. Note that the maximal number of nodes in a data set defines the dimension P of the spectra that are padded by zeros. Consequently, P also affects the computation times for PCA but the greater dimension is processed in linear time. For example, for the Arabidopsis collection $P_A = 5{,}164$ and $M_A = 49$ and for the tailbud data, $P_T = 189$ and $M_T = 3,221$. When choosing the linear term for the greater value, for the tailbud much more operations have to be performed in comparison to the Arabidopsis collection ($P_A M_A^2 < P_T^2 M_T$). Because these measurements depend on the structure of the cell lineages which are never modified, they have to be applied only once for a new data set and are also stored on the disk for later sessions.

Regarding the space usage for generating the structure map, the spectral matrix $S \in \mathbb{R}^{M \times P}$, the information of isomorphic trees, the ordering as well as the descriptor values are stored. Additionally, the set of lineages trees C is represented by pointers in a binary tree structure storing for each node its ID and parent-children relationships. This means that the algorithm has a space complexity of at most $\mathcal{O}(MP + M)$. For example, the generation of the structure map for the Arabidopsis collection requires a space usage of approximately 3.6 GiBs and for the tailbud data it is approximately 2.4 GiBs (64 bits double precision). The space usage is very high but this storage guarantees that the interactivity with the structure map is realized in linear time for the number of total nodes. This permits an immediate visual feedback when steering

the parameters.

Visual Analysis with Structure Map

In this section, I briefly explain the functionality and methods for the visual exploration of the structure map. The map is displayed in a 2D window (Figure 6.5, left).

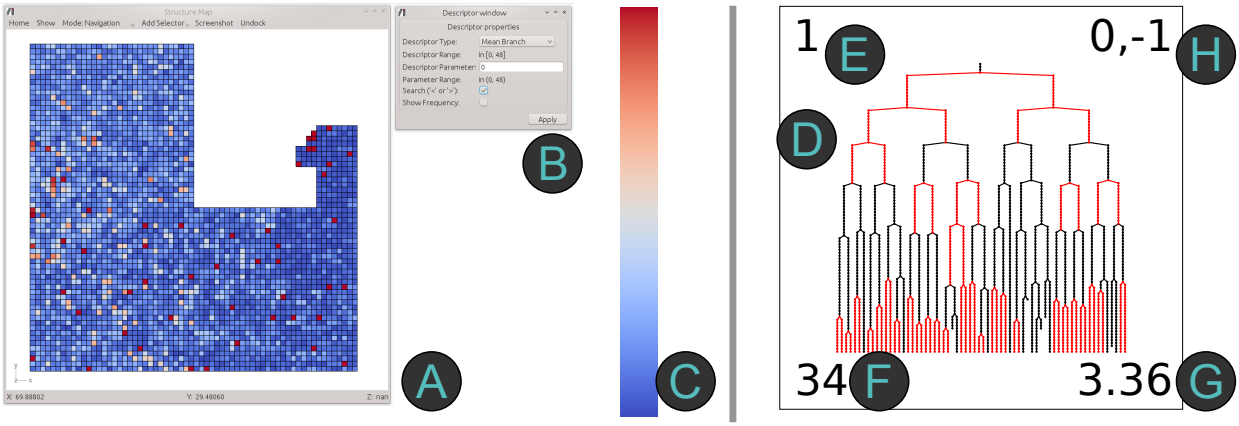


Figure 6.5: Left: Workspace for visual analysis. The structure map is displayed in a 2D window (**A**) with additional descriptor settings (**B**). I use a diverging color map described by Moreland [Mor09] to illustrate similarities (**C**). **Right: Magnified version of a common tile in the structure map.** Local features in the tree are colored red (**D**) based on the selected descriptor type. The frequency of the unique tree structure with respect to the complete data set is shown in the top left label (**E**). This indicates the total number of cospectral trees. The frequency of local features in the tree based on the selected descriptor (here: symmetric branches) is displayed at the bottom left (**F**). The descriptor value for the whole tree (here: mean branch asymmetry) is shown at the bottom right (**G**). For the Arabidopsis data, when investigating several data sets, an additional pair of labels in the top right corner of each tile is shown. The first value indicates the data ID while the second one denotes the cell file information (introduced in Section 4.1) of the lineage tree (**H**). This area is blank when investigating the zebrafish data.

Overview of complete isomorphic cell lineages: Users can use the map to get an overview of all isomorphic cell lineages in the data set. These are already sorted based on the similarity analysis of their spectra. This reveals structural differences encoded in the graph spectrum, such as the number of leaf or division nodes, as well as the total number of nodes.

Coloring of tiles based on tree descriptors: The tiles of the map are colored according to the selection of tree descriptors. By analyzing these color patterns in the map, structural descriptions are combined with highlighting of local tree features. The structure map features two coloring types for each tree descriptor. The color code can be adjusted to represent the *distance* between the descriptor-based values of trees and a user-selected scalar parameter P. For example, users can highlight tree structures with a specific mean cell branch asymmetry that are smaller or larger than P. Note that for this coloring type, (except for the number of nodes and leaves as well as the division types) the mean values of the descriptors are used in the distance computation. Furthermore, tiles can be colored by the *frequency* of a certain feature in the corresponding cell lineage. Users can search for cell cycles, for example, that are shorter than 10

time steps. Each tile is then colored according to the number of such short cycles occurring in the tree and the respective cell cycles are highlighted in red.

Semantic Zooming: In order to improve the visual analysis as well as the performance for interaction, the structure map features *semantic zooming*. Relevant objects are displayed in different levels of detail (LOD) depending on the zooming level. The level affects both the tree structure and the tile colors. Figure 6.6 illustrates the representations of the map using different LODs. The color change for the highest zoom level guarantees that the red color of the structure map does not interfere with the red color used for highlighting local features.

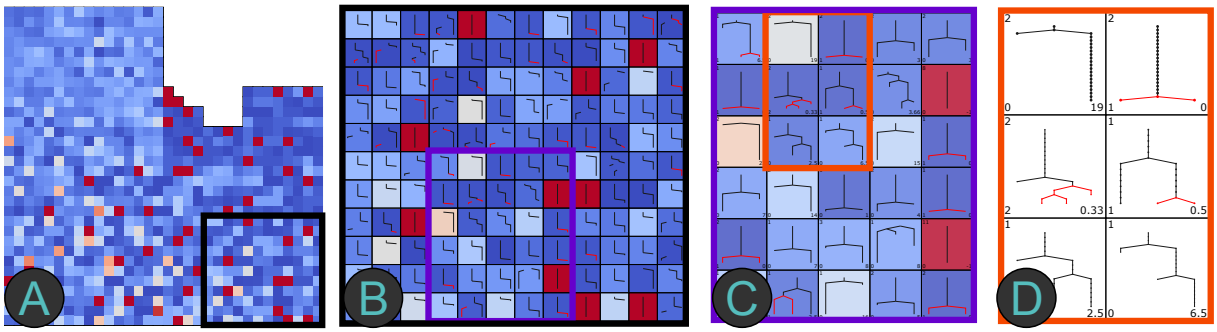


Figure 6.6: Different levels of semantic zooming: The different levels emerge in the order from left to right when zooming in while same rectangle colors represent same areas. For large distances between camera and structure map, only the tile colors and no trees are rendered (**A**). Upon zooming in, 50% of the lines and no nodes of the tree structures are displayed (**B**). With decreasing distance, each tile color fades from its original color (set by a tree descriptor) to a white background, while only lines and tile labels are drawn (**C**). The highest zoom level displays all details of the tree structure in a white tile (**D**).

6.2 Application Results and Data Comparison

I apply the structure map to the zebrafish and the Arabidopsis data sets. Through this, I demonstrate the usefulness of the map for a visual analysis of similarities and errors in thousands of cell lineages. In the epiboly data set, merging isomorphic trees reduces their number from $4{,}896$ to 875. The remaining trees are arranged in a Hilbert curve of level 5, resulting in a structure map of $2^5 \times 2^5 = 1{,}024$ tiles, 149 of which are empty. In contrast, for the tailbud data set, $3{,}221$ of originally $58{,}048$ trees remain after merging. Thus, a Hilbert curve of level 6 is generated, resulting in a structure map of $2^6 \times 2^6 = 4{,}096$ tiles, 875 of which remain empty. Both data sets contain numerous small trees with few nodes but the epiboly data set features several large trees with $2{,}000$ nodes on average, while the largest tree in the tailbud data set has merely 190 nodes.

Figure 6.7 shows structure maps for both zebrafish data sets based on the number of nodes and leaves. The number of nodes and the connectivity information of a tree is encoded by its Laplacian matrix and for cell lineage trees, both properties are correlated ($|N| - |E| = 1$ with N as the set of nodes and E as the set of edges). In Figures 6.7A and C, this correlation can be identified in the structure map. Trees with many nodes are colored in red while trees with few nodes are located in blue tiles. A similar grouping according to the number of leaves is given in Figure 6.7B. In contrast, the tailbud data set contains multiple smaller trees with approximately

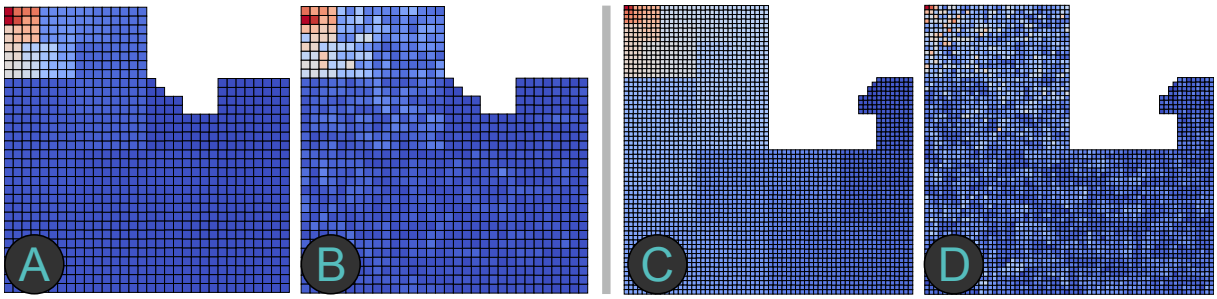


Figure 6.7: Distance-based coloring of structure maps for both zebrafish data sets. Left: Structure maps of the epiboly data based on the number of nodes (**A**) and leaves (**B**). **Right:** Structure maps of the tailbud data based on the number of nodes (**C**) and leaves (**D**).

the same number of leaves. The structure map thus cannot group them by the number of their leaves (Figure 6.7D).

Epiboly Data

The epiboly data set is assumed to contain symmetric cell division patterns in the first two cell cycles and asymmetric ones in later cycles. This is a result of the clustering of cell trajectories in chapter 5. To analyze these divisions, symmetric and asymmetric cell branches are explored.

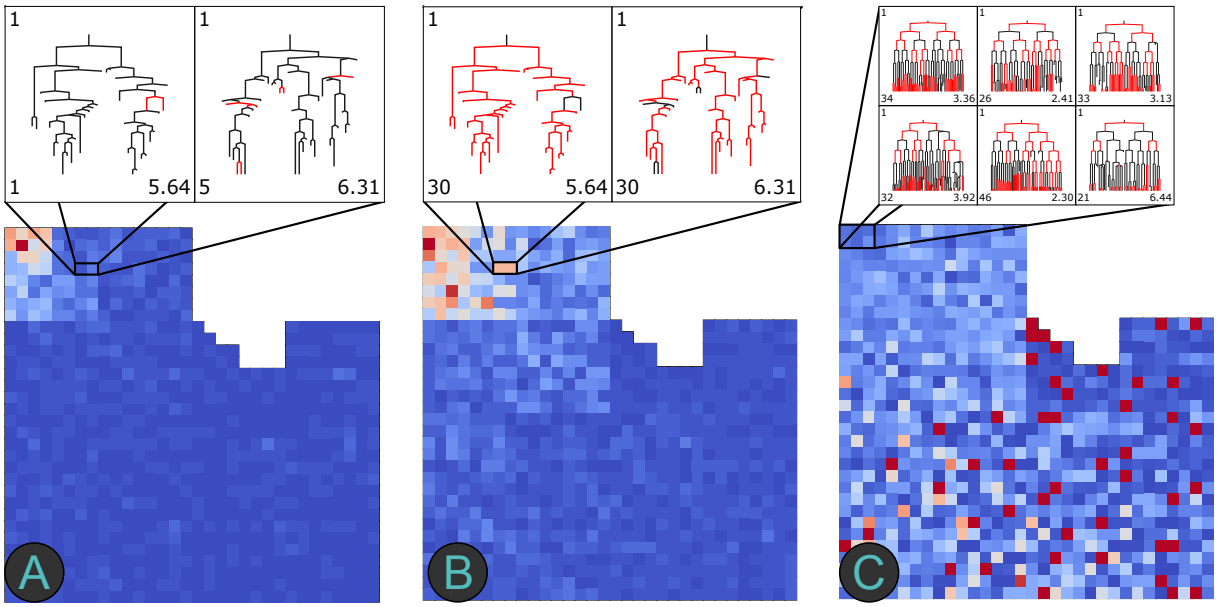


Figure 6.8: Structure maps of the epiboly data based on symmetric and asymmetric branches. The frequency-based coloring of the map is illustrated for symmetric (**A**) and asymmetric branches (**B**). (**C**) shows the distance-based structure map with focus on symmetric branches.

Figure 6.8 on page 100 shows the resulting structure map based on distances and frequencies. In Figure 6.8A, red tile colors indicate trees with the highest number of symmetric branches. These are located at the top left corner where the trees also feature the highest number of nodes. This means that large trees with many branches tend to have more symmetric ones than small trees with few branches but this is also a consequence of the higher number of nodes. However, this does not apply to all large trees, as illustrated in the pair of enhanced trees in

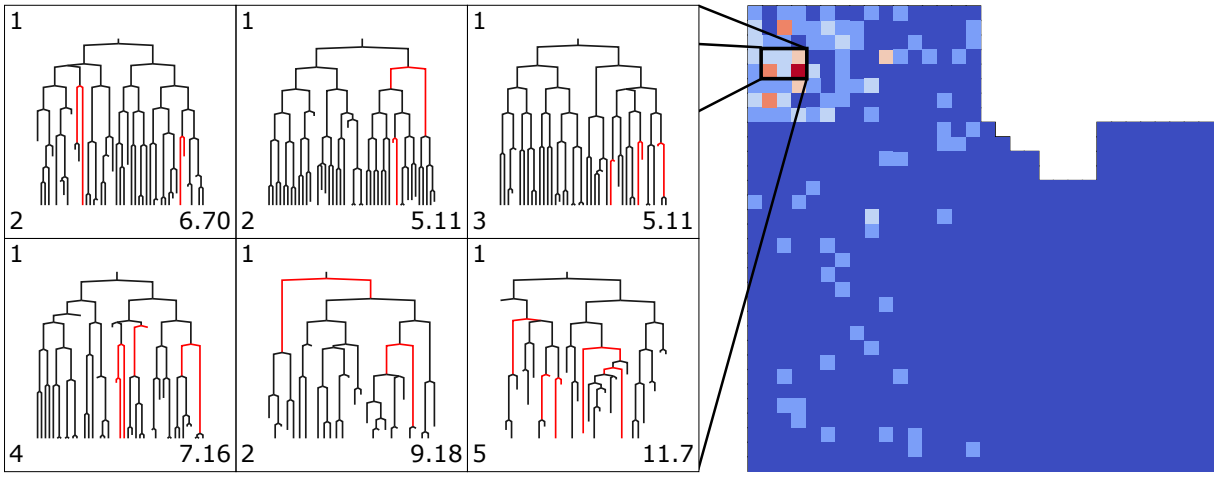


Figure 6.9: Frequency-based structure map of the epiboly data based on cell branch asymmetries greater than 25.

Figure 6.8A. Although these two lineage trees have many branches, they both exhibit erroneous connectivities as well as biologically-incorrect cell division behaviors. This is also indicated by the small number of symmetric branches (lower-left label: 1 and 5). A switching to the visual analysis of asymmetric branches is realized in Figure 6.8B. Using this visualization, the two enhanced lineage trees are identified as outliers because their tile colors indicate a comparatively large number of asymmetric branches. The majority of blue tiles in the distance-based structure map in Figure 6.8C illustrates structures that have a small mean cell branch asymmetry. Red and orange tiles correspond to trees without branches or with large differences between their mean branch asymmetry values and a chosen parameter of $P = 0$. A further analysis of blue tiles with red substructures reveals several symmetric branches mainly situated in the top left corner. These lineage trees exhibit an expected cell division behavior and they are very similar with respect to the frequencies of symmetric branches.

In order to detect outliers for which a cell division event has not been tracked, for example, cell branch asymmetries greater than 25 are analyzed. Figure 6.9 shows the resulting structure map with frequency-based coloring. The large amount of blue tiles indicates that the majority of trees do not contain many branches with large asymmetry values. This is also explained by the fact that trees whose depth is smaller than 25 cannot contain such branches. Outliers can easily be identified by red and orange tile colors, while the local highlighting of substructures serves to enable a more detailed investigation of the errors. Similar to the analysis of symmetric branches, large trees also tend to have large branch asymmetries.

Another implausible biological behavior is the occurrence of two cell divisions within a small time frame. This behavior can be analyzed by investigating cell cycle durations shorter than 4. Figure 6.10 shows the frequency-based coloring of the structure map. The majority of blue tiles implies that these trees contain longer paths but also some outliers are detected in red and orange tiles. Example of such outliers are highlighted in red and a local investigation reveals a multitude of cell division errors.

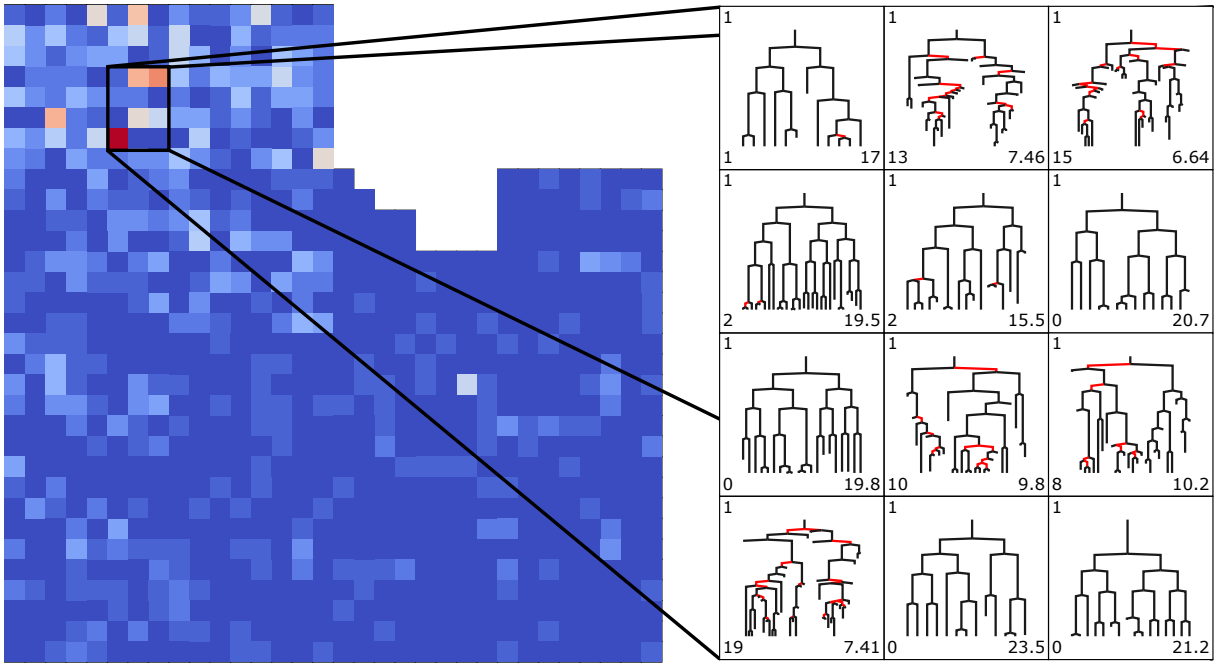


Figure 6.10: Frequency-based structure map of the epiboly data based on fully captured cell cycles smaller than 4.

Tailbud Data

Based on the results of the clustering of cell trajectories it is assumed that the tailbud data set contains fewer cell divisions, longer cell cycles, and a greater variety with respect to cell developments. Initially, cell cycles shorter than 4 are investigated in order to find implausible cell cycle phases. Figure 6.11B on page 103 shows the result using a frequency-based coloring of the structure map. Blue and light-blue are the prevailing tile colors, meaning that there are few trees with very short inner paths. This substantiates the assumption that, by and large, the data set contains longer cell cycles. In contrast to Figure 6.10A on page 102, tiles with light-blue colors are more evenly-distributed across the structure map. However, red and orange tiles indicate trees with short cell cycles (Figures 6.11A and C) that are detected immediately. To detect missing cell division events, cell branch asymmetry values larger than 25 are analyzed. In the frequency-based structure map in Figure 6.12, numerous asymmetric branches are identified. These large differences are most likely based on tracking errors missing the track for a cell migration in the longer cell division path.

In summary, these results are examples of how fast similar divisions and erroneous substructures in thousands of cell lineages based on the selected descriptor can be detected. Although, there are significantly more cell lineages in the tailbud data set, the structure map supports the detection of biological implausible structures. The different maps of both zebrafish data sets differ significantly in the frequency of identified errors. This is because of the different biological developments with either a lot of cell divisions and few cell migrations or vice versa.

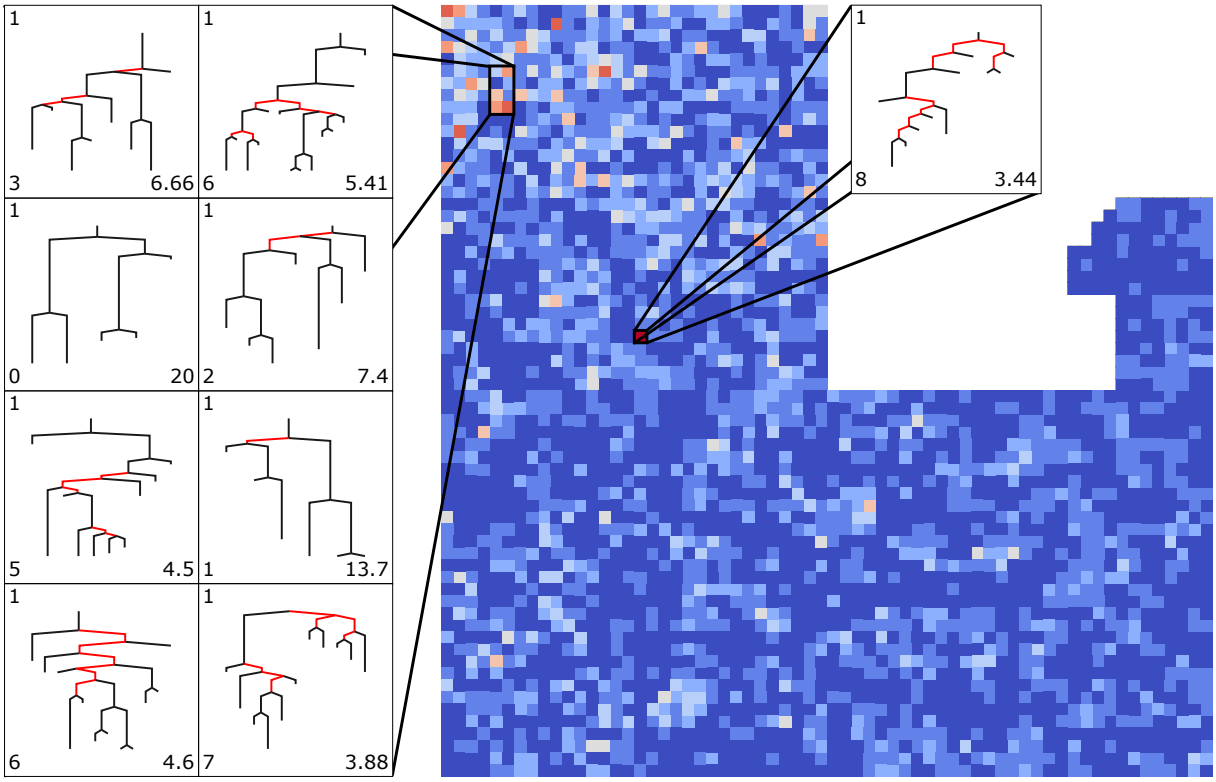


Figure 6.11: Frequency-based structure map of the tailbud data based on cell division path lengths smaller than 4.

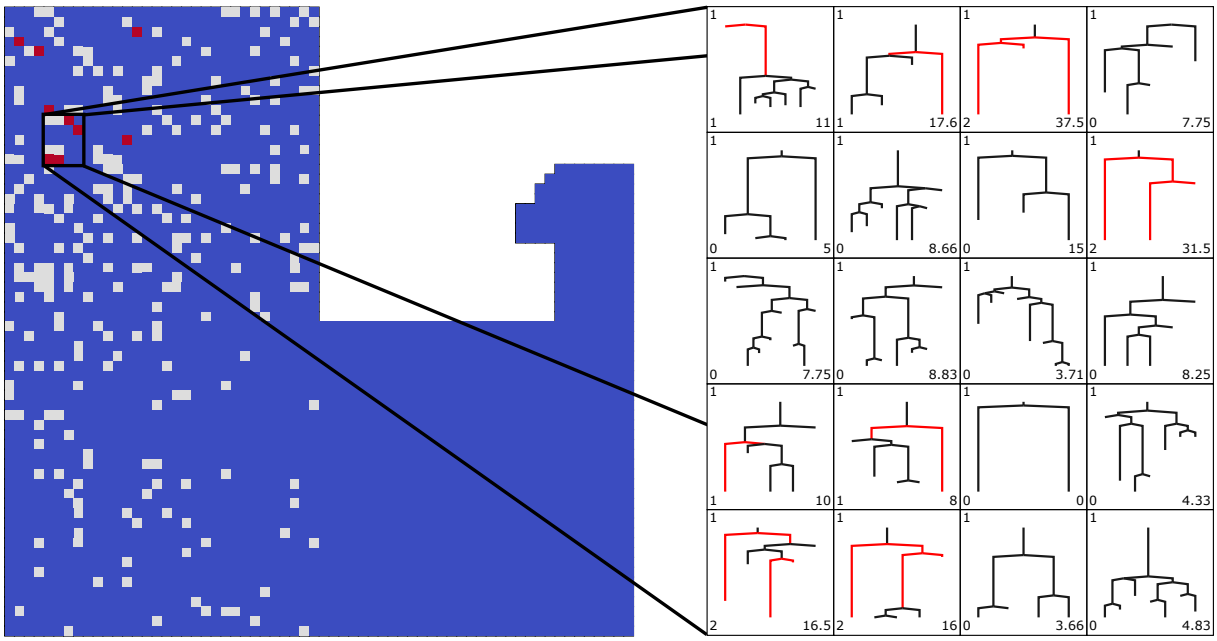


Figure 6.12: Frequency-based structure map of the tailbud data based on cell branch asymmetries larger than 25.

Arabidopsis Data

For the investigation of the Arabidopsis data, the different divisions types (anticlinal, periclinal and radial) are added as tree descriptors. Domain experts are interested in similarities and division patterns among different data sets that are supposed to describe the same biological event. The structure map provides the possibility to analyze all tree structures for all five plant data sets at once. For distinguishing between the different plants, a pair of labels at the right

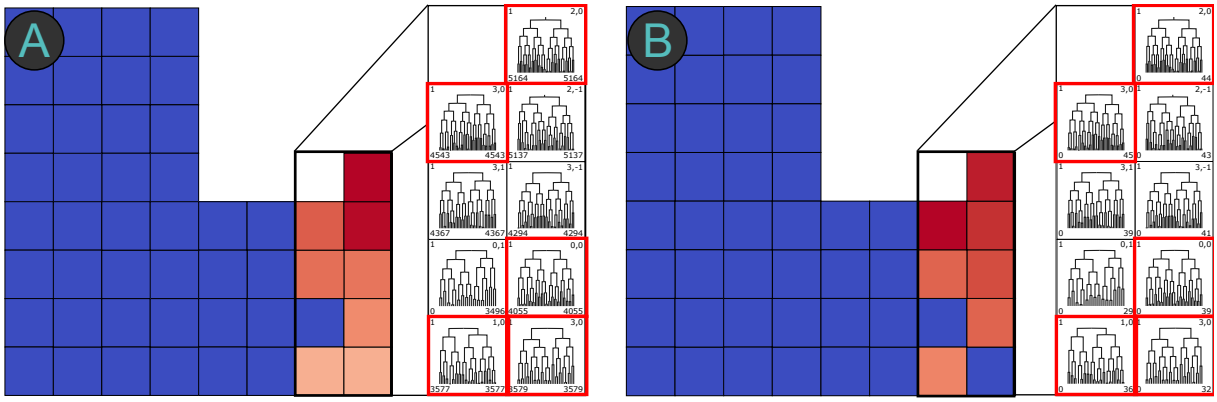


Figure 6.13: Frequency-based structure maps of the Arabidopsis collection based on the number of nodes and leaves. The structure maps are colored based on the number of nodes (**A**) and leaves (**B**) among all lineage trees. The trees in red and orange tiles have at least 3500 nodes and at least 35 leaves which are mainly located in the master cell file (red rectangles) of each plant data.

corner in each tile (see Figure 6.5, right on page 98) is shown. The first number indicates the data ID while the second one denotes the cell file assignment of the corresponding tree. Note that several cell lineage trees can belong to the same cell file.

A preliminary result is that all lineage trees have different spectra and are isomorphic. This means that each tree structure is unique up to symmetry (indicated by the 1 in each top left corner of a tile). The data set is thus not being compressed. Note that the recording of the data set 121211 starts a little bit later in comparison to the other plants. This means that for this data the first anticlinal division is missing eventually merging two different trees. As a consequence, I do not consider this data in the analysis with the structure map because I want to focus on the same biological period of cell events in order to arrive at valid conclusions about the division behaviors. Note that the combination of the data sets $120830, 121204, 130508$ and 130607 is called here the Arabidopsis collection and each data set has an unique ID in the map (120830 has 0, 121204 has 1, 130508 has 2, and 130607 has 3). The total amount of unique tree structures is $10 + 15 + 9 + 15 = 49$ resulting in a Hilbert curve of level 3 with $2^3 \times 2^3 = 64$ tiles in the structure map, 15 of which are empty.

In the Arabidopsis collection, I focus on dominant occurrences of structural features based on all tree descriptors such as cell cycle lengths, cell branches or the periclinal divisions. These divisions are mainly responsible for the growth in height of the lateral root. In order to explore dominant behaviors, relatively high parameter values for P are selected. For this purpose, the frequency-based coloring of the structure map is used beginning with an analysis of the number of nodes and leaves (Figure 6.13). Note that the trees with the highest number of nodes and

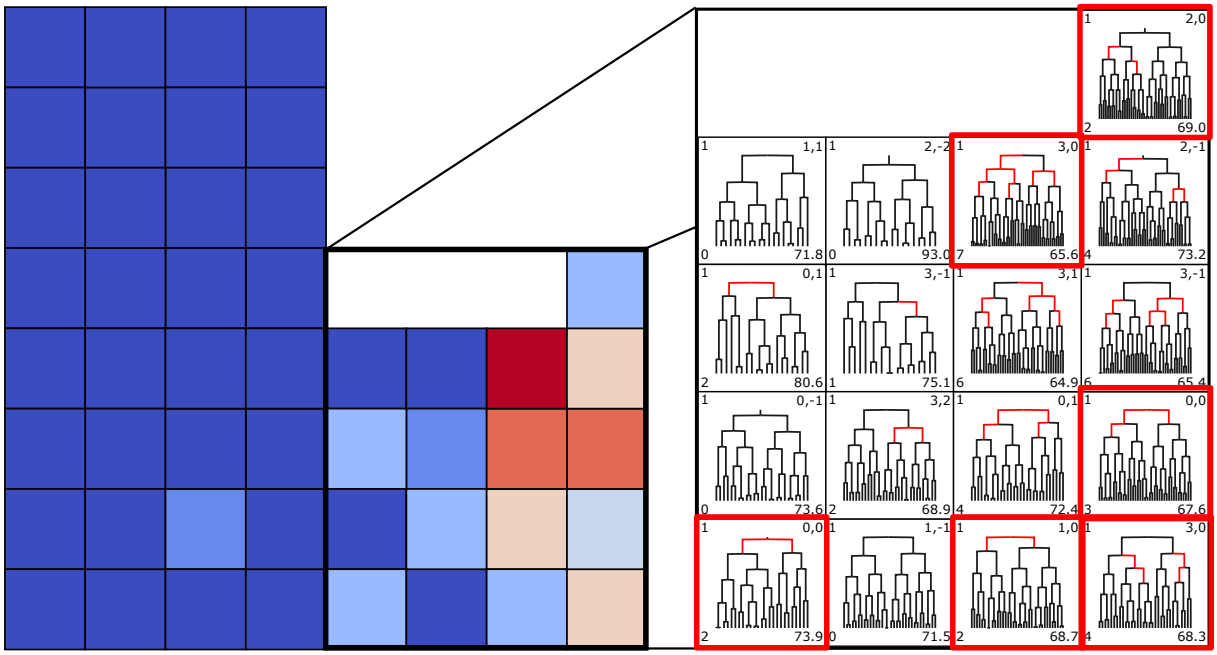


Figure 6.14: Frequency-based structure map of the Arabidopsis collection based on fully captured cell cycle lengths smaller than 50.

leaves are situated at the bottom right of the structure map because of the order of the Hilbert curve. Most of the highlighted trees in red tiles are located in the master cell file. Dominant numbers also exist for few trees situated at neighbor cell files (-1 and 1) but their amounts of nodes and leaves are always smaller compared to the master file. The remaining tiles are colored in blue, thereby illustrating that their numbers of nodes and leaves are below the selected parameters.

I further explore inner cell cycle lengths smaller than 50 that are fully captured. Figure 6.14 reveals two properties for such short cell division paths. First, these paths always occur during the first three cell cycles (highlighted cell paths in red in the enhanced trees). In later time steps,

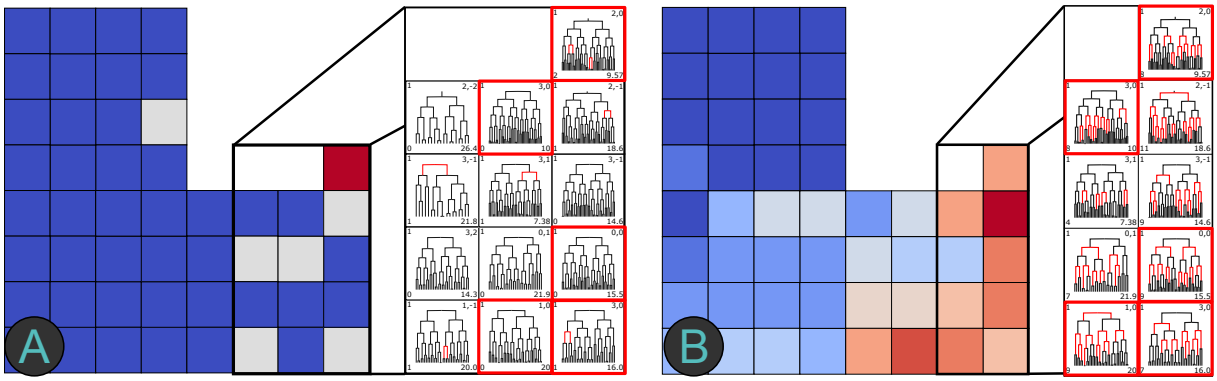


Figure 6.15: Frequency-based structure map of the Arabidopsis collection based on symmetric and branch asymmetric branches. The tiles in the map show the frequencies of symmetric (**A**) and asymmetric branches smaller than 10 (**B**).

the cell cycle durations are longer and the cells need more time to continue dividing. Second, the shortest cell paths belong predominantly to trees located in the master cell file with few in its neighbor cell files (-1 and 1). This distribution of fast cell developments in the first cell cycles implies that the cells are mainly growing in the master cell file and that the initial growth is predetermined for ultimately forming the dome-like structure.

In order to investigate the cell divisions, symmetric and asymmetric branches smaller than 10 are explored. Based on Figure 6.15A, the Arabidopsis collection features almost no symmetric branches (at most two for one cell lineage). However, in Figure 6.15B, for division branch asymmetries smaller than 10, the lineage trees located in the master cell file share almost the same number of branches. This is indicated by the same orange color of the tiles and further observable in a detailed analysis by the labels at the left bottom in each tile. These branches mainly occur during the third and fourth cell cycle indicating that at this stage, similar cell divisions based on their branch are taking place. The quantity and location of the different types of

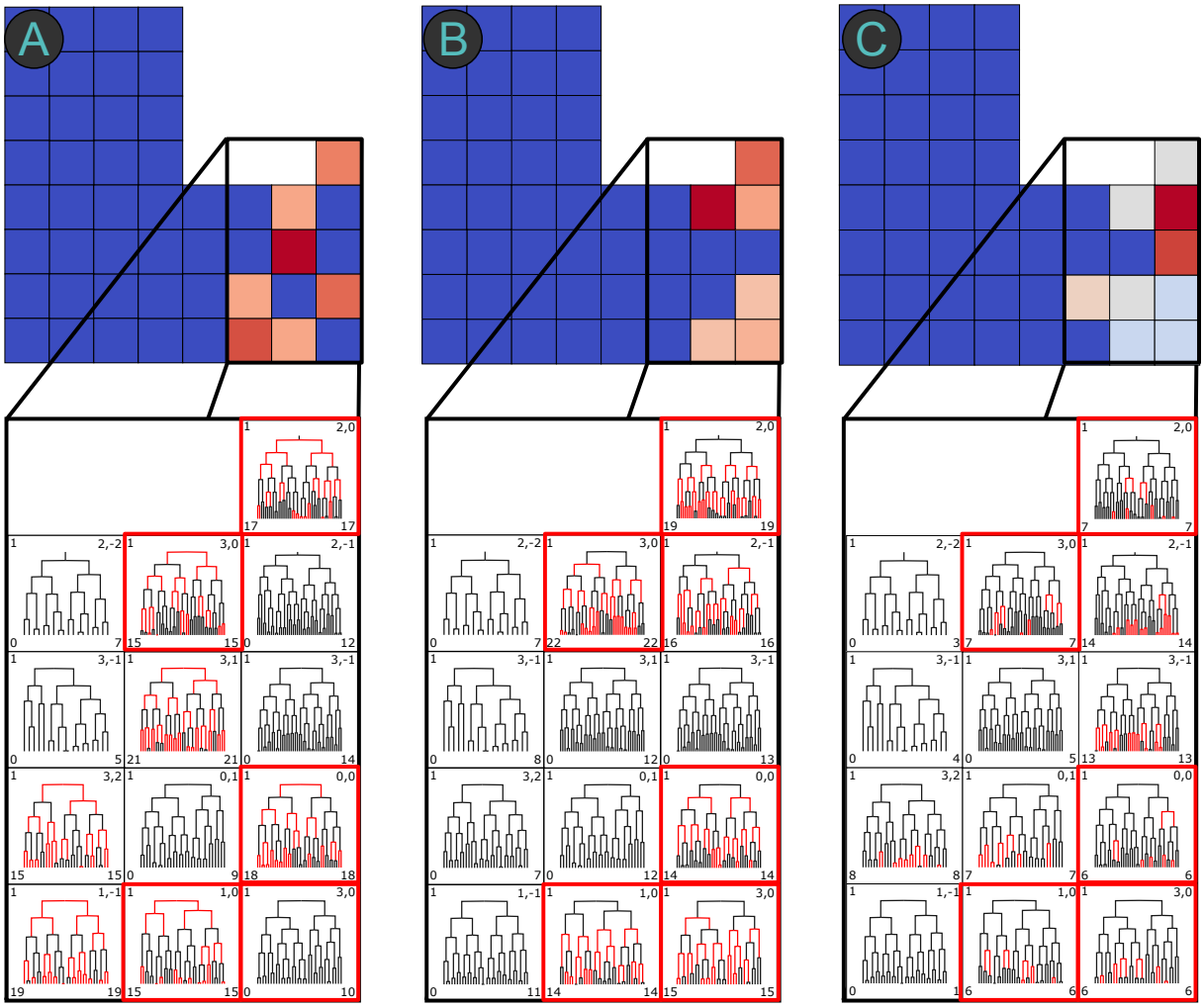


Figure 6.16: Frequency-based structure map of the Arabidopsis collection based on all three division types. In order to investigate and compare the different division schemes among the plants, dominant occurrences the division types are investigated. The trees have more than 14 anticlinal divisions (**A**), more than 13 periclinal divisions (**B**), and more than 5 radial ones (**C**).

divisions (anticlinal, periclinal, and radial) are also of high interest. Therefore, they are considered as additional descriptors in the map. Figure 6.16 shows the three resulting frequency-based maps for each division type. Different parameters for the investigation of anticlinal, periclinal, and radial divisions are chosen in such a way that all tiles with trees located in the master cell files are taken into account. The maximum occurrences for all types can be mostly identified in the master cell files while only a few trees with dominant numbers are assigned to neighbor files ($-1, 1$ and 2).

As a result, the tree structures located in the master cell file dominate in the number of appearances for all considered tree descriptors, i.e. nodes, leaves, cell cycle durations, branches, and division types. Note that the master cell file is defined by its maximal contribution to the complete tissue of the dome-like structure. Using the structure map reveals that this property correlates with the predominant occurrences of features in local tree structures. Consequently, the cells in the master cell file are essential for the total development of the lateral root.

6.3 Summary

In this chapter, I introduced the structure map which is a novel visualization method. It is well-suited for the interactive visual analysis of large tree data, especially on data from developmental biology. The structure map yields a simple and compact overview of several thousands of cell lineage trees. It permits both global analysis based on color-coding, as well as local investigation of substructures or cell paths in trees based on multiple tree descriptors.

I applied the structure map to two different data sets of the zebrafish. As a result, biologists are able to analyze similar biological events in highlighted cell paths of trees. The combination of visual analysis and the steering of parameters yield the identification of erroneous substructures such as too long cell cycles or subsequent cell divisions, for example. In addition, the comparison of cell lineage trees among several data sets reveals similarities and patterns among different Arabidopsis plants.

In the next concluding chapter, I discuss the usefulness of all presented methods (automatic classification of division types, clustering of cell trajectories, structure map), compare them to related research, and give proposals for future research for improving their functionalities.

Chapter 7

Conclusion

"Finally, in conclusion, let me say just this."
— Peter Sellers, in "Party Political Speech", 1958

In this thesis, I introduced novel interactive visualization methods that help identify similar migration and division patterns of 3D+t cell developments. Their usefulness was demonstrated by applications to the zebrafish and Arabidopsis model organisms. Through this, new insights and knowledge about cell behavior could be identified that support domain experts in their analyses and formulations of hypotheses. The first method enables a novel **automatic classification of different division types** (anticlinal, periclinal, radial) during the growth of lateral roots in plants. The visualization permits quantitative and geometrical comparisons of division schemes among several plants demonstrated in Arabidopsis data sets. The second approach facilitates a new **visual similarity analysis of plenty of 3D cell trajectories**. Its usefulness is demonstrated by application to data of both model organisms. Based on a weighted combination of migration and shape features, the trajectories are compared with each other and grouped using a hierarchical clustering technique. The resulting clusters are presented in 3D which permits an intuitive interpretation of the real-world cell developments. The **structure map** is the third visualization method which can be applied to thousands of 2D trees. This map permits an interactive comparison of specific features of unique tree structures in a compact matrix-based layout. Through this, erroneous substructures or biological plausible events based on user-defined tree descriptors can be identified immediately.

7.1 Discussion and Future Research

In this concluding chapter, I examine the three presented approaches and compare them to other related methods. I further discuss why I made the respective design decisions and give ideas for future research.

Automatic 3D Classification of Division Types

Applied to five Arabidopsis data sets, the classification method in chapter 4 yields the detection of similar division properties among all plants. This algorithm is a novel approach for deter-

mining the three division types and the visualization of the isosurfaces provides a new way to interpret the locations and contributions of periclinal divisions.

For such an analysis, an investigation in the 3D space is fundamental to process the data. Due to the recent availability of this 3D+t data, only little research has been done so far, e.g. the ongoing work by Wangenheim [vW15]. Before, mainly the manual inspection of anticlinal and periclinal divisions was in focus of research because only 2D data was available. For example, Malamy and Benfey [MB97], Dubrovsky et al. [DCCRD01], and Casimiro et al. [CBG$^+$03] explored the lateral root development of the Arabidopsis plant only in 2D based on microscopy images and formulate hypotheses according to this limited information. However, for a biological process that occurs in 3D, it is necessary to consider all three dimensions of cell migrations that affect the development of the lateral root. For this reason, I perform the presented classification algorithm and the visual analysis in 3D. The additional visualization of the division types and sequences in a lineage diagram gives an overview of the division properties and is well-suited to compare cell developments among several plants. As a result, an unknown order of division patterns is detected.

In this context, an extension of the method could be a clustering of the division sequences with respect to their order. The analysis of the 14 cell lineages located in the master cell file in each plant data in Section 4.3 is performed visually. However, if new data sets with more time steps are available or if the division types of all cells shall be compared at once then an automatic approach is required. Although the number of divisions in each sequence differs, a partial similarity analysis of *Longest Common Subsequences (LCSS)* [ALSS95] could be applied, for example. Consequently, similar division schemes could be identified. The question might occur whether the classification algorithm can also be applied to other data such as cells from zebrafish, for example. The lateral root development in plants is special in its growth behavior. It follows a high regularity in forming the dome-like structure only by cell division and growth because its cells exist in rigid cell walls without migration. The algorithm can be applied to all plant cells with slight modifications. In contrast, in the fish, cells can divide all over the embryo with arbitrary orientations. Thus, a unique classification of division types in this system would not make much sense.

As stated in Section 4.2.1, the classification algorithm is based on the generation of cell isosurfaces (α-shapes) for which the isovalue represents the number of periclinal divisions. I choose this design approach because it solely relies on real data points of single nuclei positions. For each time step, only cells in the next time step are taken into account without including other external information such as knowledge about future developments. I further use α-shapes because they yield an adequate representation of the tissue and layers of the developing lateral root in contrast to the rough convex hull that results from a Delaunay triangulation. As already discussed in Section 4.2.4, the resulting classifications of the division types are significantly influenced by the user-selected thresholds δ and ρ. Due to the variation of real-world data of cell developments, small changes of the parameters immediately affect the number of generated division types. Although a parameter setting of $\delta = 50, \rho = 45$ yields good results, the automatic approach fails to deliver a correct assignment for a low number of cells ($< 5\%$). These cases are fixed by a manual editing process and stored on the disk for later sessions. The decision for such a manual interaction is made because for real-world data of a living organism, it is assumed that there is always the possibility that cell outliers might occur that are not registered

in the automatic approach. A future improvement could minimize the manual effort to correct them by considering context-based information. For example, most of the cells with false division types are located at the periphery while for cells at the dome tip the classification is always correct. This spatial information could be set for the initial cells at time step t_0 at the periphery which is further inherited to later daughter cells evolving from the initial ones. Thus, for cells originating from periphery cells, exceptions in the classification process could be allowed and addressed as a special case.

Regarding the process of data reduction with respect to the huge initial volume size of the data sets of several hundreds of GiBs, the classification algorithm only requires the 3D nuclei positions and tracking information for each time step. This refers to a space usage of only a few MiBs for each plant that contains all relevant data for the algorithm. As already discussed in Section 4.2.3, the required computation time for each plant is less than one second. This fast processing together with the interactive 3D visualization of the isosurfaces and 2D lineage diagram permits a convenient analysis of the data.

Clustering of 3D Cell Trajectories

The presented clustering algorithm in chapter 5 permits the extraction of similar migration features in data of thousands of 3D cell trajectories. The weights λ_i in equation 5.10 allow users to emphasize the clustering results according to features of interest in an unprecedented way. As a result, in Section 5.4, similar trajectory properties such as cell cycle lengths, velocities, and shapes are identified in the epiboly data while similar collective cell developments are detected at the periphery in the tailbud data set. Furthermore, for all Arabidopsis plants, a hitherto unknown correlation between cell divisions and subsequent nuclei displacements could be found. The method can be applied to any kind of 3D+t trajectory or time-series data with appropriate changes according to how the trajectories are defined. In this thesis, the trajectory generation is motivated by the analysis of cell cycles between two subsequent cell divisions. However, this definition can be modified, e.g. to focus on cell developments in the complete time step range of the data record. This means that a trajectory would be defined between each root and leaf of a cell lineage. Nevertheless, 2D or 3D trajectories can be defined arbitrarily and analyzed with the presented method.

The visual analysis of the clustered cell trajectories is realized in the three-dimensional space. I choose this representation in order to investigate cell developments in the same space from which they originate. Consequently, users have an intuitive and natural representation of the real-world data such that the interpretation of cell events is easily accessible. Other approaches with biological applications are only realized in 2D resulting from microscopic movie data [WHN+10, dFCS03, SLM13]. These methods complicate an adequate interpretation of real-world data because the consideration of the third dimension is essential for a correct investigation of the data.

The investigated data sets with several tens to hundreds of GiBs are reduced in such a way that only relevant and important information is exploited. Note that the zebrafish data sets have a low quality with lots of outliers and thus erroneous cell information. For this reason, filter options such as the consideration of certain trajectory lengths are included prior to feature computation and clustering to exclude them in the analysis. Furthermore, a level of detail method (edge criterion in Section 5.1) is used to improve the visual analysis as well as to focus

on main migration directions of cells. For this purpose, only a binary choice between not using any LOD reduction and using the lowest LOD is allowed. At first, this processing might seem to be a rough simplification of the migration complexity but it is valid for the purpose of visually analyzing only trends and tendencies of cells. Furthermore, it is guaranteed that all trajectories are represented as a single line with equal dimension for the feature vectors $\vec{f}$. Thus, no pruning at the back of a trajectory is necessary to be further processed in the clustering approach. In total, the algorithm requires only at most 30 MiBs for the tailbud data, for example to ensure that all relevant information is preserved for clustering.

Other analysis methods focus on the identification of single similar properties of 3D trajectory data. For example, Khairy and Keller [KK11] as well as McMahon et al. [MSFS08] use color coding to represent similar migration features such as orientation or time. However, their analyses only allow a rough visualization of cell events and are not suited for a detailed similarity analysis of plenty of 3D trajectories. In order to find similar migration behaviors based on certain features and to cope with large data sets, I use a hierarchical clustering approach (Section 5.3). This approach permits an extraction and grouping of similar trajectories based on a distance function. This processing could also be realized with other clustering approaches as it was done, e.g. by Andrienko et al. [AAH$^+$13] with a density-based clustering of 2D trajectories. However, the generated hierarchy and its visual representation in a dendrogram are valuable information for users. Through this, an adequate number of clusters can be determined and it can be inspected when and where clusters are merged. The latter issues are not considered in this thesis. However, biologically motivated, an extension could be to analyze merged trajectories that are subsequently grouped into the same cluster (which are also the ones that are most similar to each other). This similarity could be compared with their positions within the embryo and result in knowledge about similar spatial cell developments. In this context, grid-based clustering approaches could support the aforementioned analysis of finding correlations between trajectories in similar regions of the embryo. Optionally, such context-based information could be assigned to each trajectory as a priori knowledge included in the similarity analysis.

I compute the distance between trajectories using a weighted combination of several migratory and geometrical features. This is a novel approach to emphasize certain movement-based features in the clustering. Motivated by the analysis of cell developments, I consider information of orientation, cell cycle duration, velocity, and the shape of trajectories (Section 5.2). For general 2D trajectories, information about speed, distance, duration, and directions are commonly used for the comparison [SBJ$^+$11, GWY$^+$11]. Geodesic and coupling distances have not been used for computation of orientations and shapes so far in this context. The focus on certain migration features can be realized by setting different values for the weight parameters λ_i in the distance function described in Section 5.2. According to the cluster validity check in Section 5.3.2, the robustness of the clustering results strongly depends on the data quality and the events in the data that are affected by the changed weights λ_i. For this reason, I only consider single features (only one $\lambda_i \neq 0$) in the similarity analysis of the zebrafish data because of its low quality and high variance of features. In contrast, a weighted analysis of trajectory properties is possible for the Arabidopsis plants that feature a higher quality and more structured migration directions.

According to the performance analysis in Section 5.3.1, the visual analysis of the clustering results is realized in real-time while the algorithm can take several minutes to finish. This

is caused by the quadratic time complexity to compute the coupling distance between each pair of trajectories. However, this information is computed only once for each data set which guarantees a fast reprocessing.

Visual Analysis in Structure Map

The structure map presented in chapter 6 is a visualization method that provides a similarity analysis of a huge collection of binary tree structures. Its usefulness is demonstrated by applications to the zebrafish and Arabidopsis data sets (Section 6.2). As a result, biological implausible substructures and similar features can be detected immediately.

In order to reduce the information of the large amount of thousands of tree structures, two dimension reduction techniques are applied: Spectral analysis and Principal Component Analysis (PCA). I determine the similarity between trees using spectral analysis. Other approaches like the *tree edit distance* [Tai79], for example, require an ordered labeling of trees. Moreover, this method is generally NP-complete for unordered trees [Bil05]. In contrast, spectral analysis provides an elegant way to describe arbitrary trees and even graph structures. This means that the structure map can be applied to any type of trees and graphs. As already discussed in Section 6.1.1, there are upper bounds for the uniqueness of a tree spectrum [WZ08, ME12] studied on randomly generated trees. But for the cell lineages of the investigated real-world data sets, all cospectral binary trees are isomorphic. As a result, the collection of trees is reduced significantly (82% for epiboly and 95% for tailbud) to unique tree structures which is a huge simplification for the visual analysis. Regarding the time complexity, the eigensystem decomposition required for the spectral analysis is cubic in the number of nodes which is the same for a robust version of the tree edit distance [PA11], for example. The determined spectra are then processed using PCA. This results in a further data reduction and an ordered one-dimensional embedding of the trees. This embedding is verified by the first principal component scores of more than 85% for all data sets. In total, the generation of the structure map has a cubic time complexity (Section 6.1.4) but the resulting information is stored and assures that the visual analysis is realized in real-time.

A straightforward horizontal alignment of the ordered unique tree structures, for example, does not scale to several thousands of trees which makes a visual comparison tedious. For this reason, I choose a matrix-based design consisting of unique tree tiles that takes advantage of the two dimensions. Through this, the map scales to even larger data sets and no overplotting issues occur in contrast to enclosure layouts like *treemaps* [Shn92] or crossings methods like *beamtrees* [vHvW03]. The tiles are arranged along a space-filling Hilbert curve which better preserves local similarity between trees than a simple arrangement of tiles line by line. However, as already mentioned in Section 6.1.2, the inefficient space usage depending on the number of unique trees and the level of the Hilbert curve may complicate the interpretation of the map. This could be solved by using approaches inspired by space-filling methods like *squarified treemaps* [BHvW99] without the enclosure of child nodes within parent nodes. In the map, unique tree structures could be assigned to quadrangular tiles with varying sizes depending on the number of nodes of a tree, for example. This could be realized in such a way that the complete available 2D space is used.

The current tree descriptors are biologically motivated but in fact, arbitrary descriptors or even graph invariants as suggested by Navigli and Lapata [NL07] can be defined. Especially

for the analysis of cell lineages, the time parameter of a cell development could be considered as well. In the current version of the structure map, the corresponding time of a tracked cell is not known. The temporal information could be integrated as an additional descriptor because cells could share identical tree developments but with completely different time spans. Another extension could be to consider weighted edges in graphs or trees. Weights may be used to emphasize or suppress certain connectivity information between nodes in the similarity analysis.

In summary, all visual analysis methods presented in this thesis are realized in real-time such that users are supported in answering research questions. The combined usage of two different analysis strategies (exploring 3D cell trajectories and the corresponding 2D cell lineages) is a new visualization approach that adds a contribution for an augmented understanding of cell behaviors. However, there is still a lot of potential in the visual analysis of 3D+t trajectory data. For example, in the zebrafish, it is still unclear what causes cells to move and how their cell fate can be determined. This requires a further global analysis of longer data records covering the development from single cells to complete organisms. Furthermore, for the recent 3D+t data of the Arabidopsis plants, the analysis so far does not completely answer the question how the plant cells in the lateral root are able to cope with both plasticity and robustness during their growth. These important questions will be the motivation for future research of these model organisms and there will always be a need for visual analysis methods to explore them.

Bibliography

[AA13] N. V. Andrienko and G. L. Andrienko. Visual analytics of movement: An overview of methods, tools and procedures. *Information Visualization*, 12(1):3–24, 2013.

[AAG00] N. V. Andrienko, G. L. Andrienko, and P. Gatalsky. Supporting visual exploration of object movement. In *Advanced Visual Interfaces*, 2000.

[AAH+13] G. L. Andrienko, N. V. Andrienko, C. Hurter, S. Rinzivillo, and S. Wrobel. Scalable analysis of movement data for extracting and exploring significant places. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1078–1094, 2013.

[AAW07] G. L. Andrienko, N. V. Andrienko, and S. Wrobel. Visual analytics tools for analysis of movement data. *Knowledge Discovery and Data Mining Exploration Newsletter*, 9:38–46, 2007.

[ABKS99] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In *Proceedings of the International Conference on Management of Data*, pages 49–60, 1999.

[ACSv12] B. Arsić, D. Cvetković, S. K. Simić, and M. Škarić. Graph spectral techniques in computer sciences. *Applicable Analysis and Discrete Mathematics*, 6:1–30, 2012.

[AG96] H. Alt and L. J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation – a survey. In *Handbook of Computational Geometry*, pages 121–153. Elsevier Science Publishers B.V., 1996.

[AH98] K. Andrews and H. Heidegger. Information slices: Visualising and exploring large hierarchies using cascading, semi-circular discs. In *Proceedings of IEEE Symposium on Information Visualization*, 1998.

[ALSS95] R. Agrawal, K.-I. Lin, H. S. Sawhney, and K. Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proceedings of the International Conference on Very Large Data Bases*, pages 490–501, 1995.

[Alt09] H. Alt. The computational geometry of comparing shapes. In *Efficient Algorithms*, pages 235–248. Springer Verlag, 2009.

[BB97] F. Bernardini and C. L. Bajaj. Sampling and reconstructing manifolds using alpha-shapes. In *Proceedings of the 9th Canadian Conference on Computational Geometry*, 1997.

[BBM+06] T. J. Boyle, Z. Bao, J. I. Murray, C. L. Araya, and R. H. Waterston. Acetree: a tool for visual analysis of caenorhabditis elegans embryogenesis. *BioMed Central Bioinformatics*, 7(275):275–289, 2006.

[BHvW99] M. Bruls, K. Huizing, and J. van Wijk. Squarified treemaps. In *Proceedings of the Joint Eurographics and IEEE TVCG Symposium on Visualization*, pages 33–42, 1999.

[Bil05] P. Bille. A survey on tree edit distance and related problems. *Theoretical computer science*, 337(1):217–239, 2005.

[Bus97] S. Buss. Alogtime algorithms for tree isomorphism, comparison, and canonization. In *Computational Logic and Proof Theory*, pages 18–33. Springer, 1997.

[BvTLH+11] S. Bremm, von T. Landesberger, M. Hess, T. Schreck, P. Weil, and K. Hamacherk. Interactive visual comparison of multiple trees. In *IEEE Conference on Visual Analytics Science and Technology*, pages 31–40, 2011.

[CBG+03] I. Casimiro, T. Beeckman, N. Graham, R. Bhalerao, H. Zhang, P. Casero, G. Sandberg, and M. J. Bennett. Dissecting arabidopsis lateral root development. *Trends in Plant Science*, 8:165–171, 2003.

[CBI+07] A. Cedilnik, J. Baumes, L. Ibanez, S. Megason, and B. Wylie. Integration of information and volume visualization for analysis of cell lineage and gene expression during embryogenesis. In *Proceedings of SPIE – The International Society for Optical Engineering*, volume 6809, 2007.

[CBR10] A. Cardona and R. Bryson-Richardson. Fiji plugin: Correct 3D drift. Developmental Imaging EMBO course, 2010.

[Chi01] A. Chisholm. Cell lineage. In *Encyclopedia of Genetics*, pages 302–310. Academic Press, 2001.

[Cve12] D. Cvetković. Spectral recognition of graphs. *Yugoslav Journal of Operations Research*, 22(2):145–161, 2012.

[DCCRD01] J. G. Dubrovsky, A. Colón-Carmona, T. L. Rost, and P. Doerner. Early primordium morphogenesis during lateral root initiation in Arabidopsis thaliana. *Planta*, 214:30–36, 2001.

[Del34] B. N. Delaunay. Sur la sphère vide. *Bulletin of Academy of Sciences of the USSR*, 6:793–800, 1934.

[dFCS03] L. da Fontoura Costa and D. Schubert. A framework for cell movement image analysis. In *Proceedings of the 12th International Conference on Image Analysis and Processing*, pages 271–276. IEEE Computer Society, 2003.

[DLM+98] G. Das, K.-I. Lin, H. Mannila, G. Renganathan, and P. Smyth. Rule discovery from time series. In *Knowledge Discovery and Data Mining*, pages 16–22, 1998.

[Dod11] S. Dodge. *Exploring Movement Using Similarity Analysis*. PhD thesis, University of Zurich, 2011.

[DWL08] S. Dodge, R. Weibel, and A.-K. Lautenschütz. Towards a taxonomy of movement patterns. *Information Visualization*, 7:240–252, 2008.

[EDSN11] B. Eriksson, G. Dasarathy, A. Singh, and R. D. Nowak. Active clustering: Robust and efficient hierarchical clustering using adaptively selected similarities. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 15, pages 260–268, 2011.

[ELLS11] B. S. Everitt, S. Landau, M. Leese, and D. Stahl. *Cluster Analysis*. Wiley, 5th edition, 2011.

[EM94a] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13:43–72, 1994.

[EM94b] T. Eiter and H. Mannila. Computing discrete Fréchet distance. Technical Report CD-TR 94/64, Technische Universität Wien, 1994.

[FHR+14] J. Fangerau, B. Höckendorf, B. Rieck, C. Heine, J. Wittbrodt, and H. Leitte. Supplemental video. `https://vimeo.com/86031849`, 2014. Accessed 06 Feb 2014.

[FHR+15] J. Fangerau, B. Höckendorf, B. Rieck, C. Heine, J. Wittbrodt, and H. Leitte. Interactive similarity analysis and error detection in large tree collections. *3rd International Workshop on Visualization in Medicine and Life Sciences*, 2015. To be published.

[FHWL12] J. Fangerau, B. Höckendorf, J. Wittbrodt, and H. Leitte. Similarity analysis of cell movements in video microscopy. In *Proceedings of the 2nd IEEE Symposium on Biological Data Visualization*, pages 69–76, 2012.

[FM83] E. B. Fowlkes and C. L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78:553–569, 1983.

[For65] E. Forgy. Cluster analysis of multivariate data: Efficiency versus interpretability of classifications. *Biometrics*, 21:768–780, 1965.

[Fur10] E. E. Furlong. The importance of being specified: Cell fate decisions and their role in cell biology. *Molecular Biology of the Cell*, 21(22):3797–3798, 2010.

[GK10] M. Graham and J. Kennedy. A survey of multiple tree visualisation. *Information Visualization*, 9:235–252, 2010.

[GMO99] M. T. Goodrich, J. S. B. Mitchell, and M. W. Orletsky. Approximate geometric pattern matching under rigid motions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):371–379, April 1999.

[Gow67] J. C. Gower. A comparison of some methods of cluster analysis. *Biometrics*, 23:623–637, 1967.

[GRS98] S. Guha, R. Rastogi, and K. Shim. CURE: An efficient clustering algorithm for large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 73–84. American Mathematical Society, 1998.

[GRS00] S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. In *Proceedings of the 15th International Conference on Data Engineering*, pages 512–521, 2000.

[GWY+11] H. Guo, Z. Wang, B. Yu, H. Zhao, and X. Yuan. TripVista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection. In *Proceedings of the IEEE Pacific Visualization Symposium*, 2011.

[Hae95] W. H. Haemers. Interlacing eigenvalues and graphs. *Linear Algebra and its Applications*, 226–228(0):593–616, 1995.

[HMM00] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6:24–43, 2000.

[HTC09] C. Hurter, B. Tissoires, and S. Conversy. FromDaDy: Spreading aircraft trajectories across views to support iterative queries. *IEEE Transactions on Visualization and Computer Graphics*, 15:1017–1024, 2009.

[HTW12] B. Höckendorf, T. Thumberger, and J. Wittbrodt. Quantitative analysis of embryogenesis: A perspective for light sheet microscopy. *Developmental Cell*, 23:1111–1120, 2012.

[Ins85] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1:69–91, 1985.

[Jen89] G. F. Jenks. Geographic logic in line generalization. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 26(1):27–42, 1989.

[Jol02] I. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, 2002.

[KBK+95] C. B. Kimmel, W. W. Ballard, S. R. Kimmel, B. Ullmann, and T. F. Schilling. Stages of embryonic development of the zebrafish. *Developmental dynamics*, 203(3):253–310, 1995.

[KBXS12] A. Krishnamurthy, S. Balakrishnan, M. Xu, and A. Singh. Efficient active algorithms for hierarchical clustering. In *Proceedings of the International Conference on Machine Learning*, 2012.

[Kel13] P. J. Keller. In vivo imaging of zebrafish embryogenesis. *Methods*, 62:268–278, 2013.

[KERC09] D. F. Keefe, M. Ewert, W. Ribarsky, and R. Chang. Interactive coordinated multiple-view visualization of biomechanical motion data. *IEEE Transactions on Visualization and Computer Graphics*, 15:1383–1390, 2009.

[KK11] K. Khairy and P. J. Keller. Reconstructing embryonic development. *Genesis*, 49(7):488–513, July 2011.

[KKEM10] D. A. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann. *Mastering The Information Age - Solving Problems with Visual Analytics*. Eurographics, 2010.

[KMNR10] S. Kisilevich, F. Mansmann, M. Nanni, and S. Rinzivillo. Spatio-temporal clustering. In *Data Mining and Knowledge Discovery Handbook*, pages 855–874. Springer, 2010.

[KMS+08] D. A. Keim, F. Mansmann, J. Schneidewind, H. Ziegler, and J. Thomas. Visual analytics: Scope and challenges. In *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*, pages 76–90. Springer-Verlag, 2008.

[Kra03] M. J. Kraak. The space-time cube revisited from a geovisualization perspective. In *Proceedings of the International Cartographic Conference*, pages 1988–1995, 2003.

[KSWS08] P. J. Keller, A. D. Schmidt, J. Wittbrodt, and E. H. Stelzer. Reconstruction of Zebrafish early embryonic development by scanned light sheet microscopy. *Science*, 322(5904):1065–1069, 2008.

[LC07] G. J. Lieschke and P. D. Currie. Animal models of human disease: Zebrafish swim into view. *Nature Reviews Genetics*, 8(5):353–367, May 2007.

[LDO+06] T. Langenberg, T. Dracz, A. C. Oates, C.-P. Heisenberg, and M. Brand. Analysis and visualization of cell movement in the developing zebrafish brain. *Developmental dynamics*, 4(235):928–933, April 2006.

[LFL+12] H. Leitte, J. Fangerau, X. Lou, B. Höckendorf, S. Lemke, A. Maizel, and J. Wittbrodt. Visualization software for 3d video microscopy: A design study. In *Proceedings of the Eurographics Conference on Visualization (EuroVis Short Papers)*, 2012.

[LGL+11] H. Liu, Y. Gao, L. Lu, S. Liu, H. Qu, and L. M. Ni. Visual analysis of route diversity. In *IEEE Conference on Visual Analytics Science and Technology*, pages 171–180, 2011.

[Li09] S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer, 2009.

[LKH10] O. D. Lampe, J. Kehrer, and H. Hauser. Visual analysis of multivariate movement data using interactive difference views. In *Vision, Modeling, and Visualization*, pages 315–322, 2010.

[LKL+11] X. Lou, F. O. Kaster, M. S. Lindner, B. X. Kausler, U. Köthe, B. Hockendorf, J. Wittbrodt, H. Jänicke, and F. A. Hamprecht. DELTR: Digital embryo lineage tree reconstructor. In *Biomedical Imaging: From Nano to Macro*, pages 1557–1560, 2011.

[LW67] G. N. Lance and W. T. Williams. A general theory of classificatory sorting strategies 1. hierarchical systems. *The Computer Journal*, 9(4):373–380, 1967.

[MB97] J. E. Malamy and P. N. Benfey. Organization and cell differentiation in lateral roots of Arabidopsis thaliana. *Development*, 124(1):33–44, 1997.

[McQ57] L. L. McQuitty. Elementary linkage analysis for isolating orthogonal and oblique types and typal relevancies. *Educational and Psychological Measurement*, 17:207–229, 1957.

[ME12] F. A. Matsen and S. N. Evans. Ubiquity of synonymity: almost all large binary trees are not uniquely identified by their spectra or their immanantal polynomials. *Algorithms for Molecular Biology*, 7(1):14, 2012.

[MGT+03] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou. TreeJuxtaposer: Scalable tree comparison using Focus+Context with guaranteed visibility. In *Proceedings of the SIGGRAPH Conference on Computer Graphics and Interactive Techniques*, 2003.

[Mor09] K. Moreland. Diverging color maps for scientific visualization. In *Proceedings of the 5th International Symposium on Advances in Visual Computing: Part II*, pages 92–103. Springer-Verlag, 2009.

[MS62] T. Murashige and F. Skoog. A revised medium for rapid growth and bio assays with tobacco tissue cultures. *Physiologia Plantarum*, 15(3):473–497, 1962.

[MSC+11] M. J. Mlodzianoski, J. M. Schreiner, S. P. Callahan, K. Smolková, A. Dlasková, J. Šantorová, P. Ježek, and J. Bewersdorf. Sample drift correction in 3D fluorescence photoactivation localization microscopy. *Optics Express*, 19(16):15009–15019, 2011.

[MSFS08] A. McMahon, W. Supatto, S. E. Fraser, and A. Stathopoulos. Dynamic analyses of Drosophila gastrulation provide insights into collective cell migration. *Science*, 322:1546–1550, 2008.

[Mun97] T. Munzner. H3: Laying out large directed graphs in 3d hyperbolic space. In *Proceedings of IEEE Symposium on Information Visualization*, pages 2–10, 1997.

[MvWF+11] A. Maizel, D. von Wangenheim, F. Federici, J. Haseloff, and E. H. Stelzer. High-resolution live imaging of plant growth in near physiological bright conditions using light sheet fluorescence microscopy. *The Plant Journal*, 68(2):377–385, 2011.

[NCA06] P. Neumann, M. S. T. Carpendale, and A. Agarawala. Phyllotrees: Phyllotactic patterns for tree layout. In *Proceedings of the Joint Eurographics and IEEE TVCG Symposium on Visualization*, pages 59–66, 2006.

[NL07] R. Navigli and M. Lapata. Graph connectivity measures for unsupervised word sense disambiguation. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 1683–1688, 2007.

[OFB10] P. Overvoorde, H. Fukaki, and T. Beeckman. Auxin control of root development. In *Cold Spring Harbor Perspectives in Biology*. Cold Spring Harbor Lab Press, 2010.

[OGG+10] S. I. O'Donoghue, A.-C. C. Gavin, N. Gehlenborg, D. S. Goodsell, J.-K. K. Hériché, C. B. Nielsen, C. North, A. J. Olson, J. B. Procter, D. W. Shattuck, T. Walter, and B. Wong. Visualizing biological data - now and in the future. *Nature Methods*, 7(3 Suppl):S2–S4, 2010.

[OLOD+10] N. Olivier, M. A. Luengo-Oroz, L. Duloquin, E. Faure, T. Savy, I. Veilleux, X. Solinas, D. Débarre, P. Bourgine, A. Santos, N. Peyriéras, and E. Beaurepaire. Cell lineage reconstruction of early zebrafish embryos using label-free nonlinear microscopy. *Science*, 329(5994):967–971, 2010.

[PA11] M. Pawlik and N. Augsten. RTED: A robust algorithm for the tree edit distance. *Proceedings of the International Conference on Very Large Databases*, 5:334–345, 2011.

[Pen08] H. Peng. Bioimage informatics: A new area of engineering biology. *Bioinformatics*, 24(17):1827–1836, September 2008.

[PP09] D. Phuc and N. T. K. Phung. Using spectral vectors and m-tree for graph clustering and searching in graph databases of protein structures. *World Academy of Science, Engineering and Technology*, 3:275–280, 2009.

[PTL+10] J. B. Procter, J. Thompson, I. Letunic, C. Creevey, F. Jossinet, and G. J. Barton. Visualization of multiple alignments, phylogenies and gene family evolution. *Nature Methods*, 7:516–525, 2010.

[RF81] D. Robinson and L. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53:131–147, 1981.

[RMC91] G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone trees: Animated 3d visualizations of hierarchical information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 189–194, 1991.

[Rok10] L. Rokach. A survey of clustering algorithms. In *Data Mining and Knowledge Discovery Handbook*, pages 269–298. Springer, 2010.

[RP66] A. Rosenfeld and J. L. Pfaltz. Sequential operations in digital picture processing. *Journal of the Association for Computing Machinery*, 14:471–494, 1966.

[RPN+08] S. Rinzivillo, D. Pedreschi, M. Nanni, F. Giannotti, N. V. Andrienko, and G. L. Andrienko. Visually-driven analysis of movement data by progressive clustering. *Information Visualization*, 7(3–4):225–239, 2008.

[RT81] E. M. Reingold and J. S. Tilford. Tidier drawings of trees. *IEEE Transactions on Software Engineering*, 7(2):223–228, 1981.

[RVC12] M. K. Rafsanjani, Z. A. Varzaneh, and N. E. Chukanlo. A survey of hierarchical clustering algorithms. *The Journal of Mathematics and Computer Science*, 5(3):229–240, 2012.

[SACF+12] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona. Fiji: an open-source platform for biological-image analysis. *Nature Methods*, 9:676–682, 2012.

[SBJ+11] D. Spretke, P. Bak, H. Janetzko, B. Kranstauber, F. Mansmann, and S. Davidson. Exploration through enrichment: A visual analytics approach for animal movement. In *Proceedings of International Conference on Advances in Geographic Information Systems*, pages 421–424, 2011.

[Sch73] A. J. Schwenk. Almost all trees are cospectral. In F. Harary, editor, *New Directions in the Theory of Graphs*, pages 275–307. Academic Press, New York, 1973.

[Sch11] H. Schulz. Treevis.net: A tree visualization reference. *IEEE Computer Graphics and Applications*, 31:11–15, 2011.

[SDD13] S. Saraçli, N. Doğan, and I. Doğan. Comparison of hierarchical cluster analysis methods by cophenetic correlation. *Journal of Inequalities and Applications*, 1:203–211, 2013.

[Shn92] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11:92–99, 1992.

[Shn96] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[SHS09] H.-J. Schulz, S. Hadlak, and H. Schumann. Point-based tree representation: A new approach for large hierarchies. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 81–88, 2009.

[SLM13] B. Slater, C. Londono, and A. P. McGuigan. An algorithm to quantify correlated collective cell migration behavior. *BioTechniques*, 54:87–92, 2013.

[SM58] R. R. Sokal and C. D. Michener. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38:1409–1438, 1958.

[Sne57] P. Sneath. The application of computers to taxonomy. *Microbiology*, 17(1):201–226, 1957.

[Sor48] T. Sorensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biologiske skrifter*, 5:1–34, 1948.

[SR62] R. R. Sokal and F. J. Rohlf. The comparison of dendrograms by objective methods. *Taxon*, 11:33–40, 1962.

[SZ00] J. T. Stasko and E. Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 57–65, 2000.

[Tai79] K.-C. Tai. The tree-to-tree correction problem. *Journal of the Association for Computing Machinery*, 26:422–433, 1979.

[TK99] S. Theodoridis and K. Koutroubas. *Pattern Recognition*. Academic Press, 1999.

[TSAA12] C. Tominski, H. Schumann, G. L. Andrienko, and N. V. Andrienko. Stacking-based visualization of trajectory attribute data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2565–2574, 2012.

[vHvW03] F. van Ham and J. J. van Wijk. Beamtrees: compact visualization of large hierarchies. *Information Visualization*, 2:31–39, 2003.

[vLKS+11] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. V. Wijk, J.-D. Fekete, and W. F. Dieter. Visual analysis of large graphs: State-of-the-art and future research challenges. *Computer Graphics Forum*, 30:1719–1749, 2011.

[Vor08] G. F. Voronoy. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. *Crelle's Journal*, 134:198–287, 1908.

[vR79] C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979.

[vW15] D. von Wangenheim. *Long term observation of Arabidopsis thaliana root growth in close-to-natural conditions using Light Sheet-based Fluorescence Microscopy*. PhD thesis, Goethe-Universität Frankfurt am Main, 2015. To be published.

[vWDL+14] D. von Wangenheim, G. Daum, J. U. Lohmann, E. K. Stelzer, and A. Maizel. Live imaging of arabidopsis development. *Methods in Molecular Biology*, 1062:539–550, 2014.

[War63] J. H. Ward, Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236–244, 1963.

[WHN+10] T. Walter, M. Held, B. Neumann, J.-K. Hériché, C. Conrad, R. Pepperkok, and J. Ellenberg. Automatic identification and clustering of chromosome phenotypes in a genome wide RNAi screen by time-lapse imaging. *Journal of Structural Biology*, 170:1–9, 2010.

[WL05] T. Warren Liao. Clustering of time series data – A survey. *Pattern Recognition*, 38:1857–1874, 2005.

[WSZ+13] H. Wang, H. Su, K. Zheng, S. Sadiq, and X. Zhou. An effectiveness study on trajectory similarity measures. In *Proceedings of the Australasian Database Conference*, pages 13–22, 2013.

[WT04] P. C. Wong and J. Thomas. Visual analytics. *IEEE Computer Graphics and Applications*, 24(5):20–21, Sep 2004.

[WvdWvW09] N. Willems, H. van de Wetering, and J. J. van Wijk. Visualization of vessel movements. *Computer Graphics Forum*, 28:959–966, 2009.

[WW07] S. Wagner and D. Wagner. Comparing Clusterings – An Overview. Technical report, Karlsruhe university, 2007.

[WY14] Z. Wang and X. Yuan. Urban trajectory timeline visualization. In *Proceedings of the International Conference on Big Data and Smart Computing*, pages 13–18, 2014.

[WZ08] R. C. Wilson and P. Zhu. A study of graph spectra for comparing graphs and trees. *Pattern Recognition*, 41(9):2833–2841, 2008.

[ZBB+08] Z. Zhao, T. J. Boyle, Z. Bao, J. I. Murray, B. Mericle, and R. H. Waterston. Comparative analysis of embryonic cell lineage between Caenorhabditis briggsae and Caenorhabditis elegans. *Developmental Biology*, 314:93–99, 2008.

[ZK08] C. Ziemkiewicz and R. Kosara. The shaping of information by visual metaphors. *IEEE Transactions on Visualization and Computer Graphics*, 14:1269–1276, 2008.

[ZMC05] S. Zhao, M. J. McGuffin, and M. H. Chignell. Elastic hierarchies: Combining treemaps and node-link diagrams. In *Proceedings of the IEEE Symposium on Information Visualization*, 2005.