Uncertainty-aware Visual Analytics and Its Applications

Thesis approved by
the Department of Computer Science
University of Kaiserslautern-Landau
for the award of the Doctoral Degree
Doctor of Natural Sciences (Dr. rer. nat.)

to

Robin Georg Claus Maack

Date of Defense:	23.08.2024
Dean:	Prof. Dr. Christoph Garth
Reviewer:	Prof. Dr. Christoph Garth
Reviewer:	Prof. Dr. Ross Maciejewski

DE-386

Acknowledgments

I would like to express my deepest gratitude to my two esteemed advisors, Hans Hagen and Christoph Garth. Their unwavering support, insightful guidance, and invaluable expertise have been instrumental throughout the course of my doctoral journey. Their encouragement and constructive feedback have significantly shaped this dissertation, and for that, I am profoundly grateful.

A special thanks goes to my friend, co-worker, and mentor who supported me from literally day one at the university, Christina Gillmann. You are the biggest reason I began and finished a career in science, as your dedication to research, support in my studies, and open ears helped me succeed throughout many of the challenges on my path.

To all my collaborators, thank you for supporting me in our numerous research adventures. Your expertise and guidance fueled our fruitful discussions, culminating in many successful projects.

I would also like to thank my research assistants, Kurt Schardt and Leo Sperling, whose dedication, hard work, and support have been a huge motivating factor in this research.

To all my colleagues, thank you for making life in our working group a fascinating adventure, filled with many insightful discussions and amusing moments. Keep up this great culture of inclusion, open-mindedness, and curiosity.

To my friends and family, thank you for your steady support and encouragement throughout my whole studies. Your belief in my abilities and your constant motivation has been fundamental. To my parents, thank you for the countless sacrifices you've made and for always being there with love and wisdom. Your support has been invaluable, and this accomplishment is as much yours as it is mine.

For her constant support and love throughout this journey, I would like to thank my girlfriend Katharina Roth. Your encouragement and belief in me have been my greatest source of strength. Thank you for your patience, understanding, and constant motivation, which helped me overcome the challenges along the way. This achievement was only possible due to your steadfast presence by my side.

Abstract

Effective communication of uncertainty is crucial for informed decision-making in data analysis workflows to estimate the reliability of available data and resulting insights. Here, visual analytics is a modern process combining algorithmic data analysis techniques and interactive visualization to extract insights from datasets. Integrating uncertainty in visual analytics is not trivial, and the growing field of uncertainty-aware visual analytics applications shows the importance of integrating uncertainty information for decision-making. Still, there is no general systematic model and guideline to integrate uncertainty in visual analytics applications to assist application designers.

This raised the questions "How to define an uncertainty-aware visual analytics process?" and "Which steps are necessary to map a visual analytics process to an uncertainty-aware visual analytics process?".

Accordingly, this thesis addresses these questions in three steps. First, an uncertainty-aware visual analytics process describes all components and transitions when dealing with data containing uncertainty, providing a model to check for compatibility of an application. A guideline for creating a compatible uncertainty-aware application delivers detailed development steps that allow converting existing applications and creating new ones from scratch. At last, multiple uncertainty-aware visual analytics applications are showcased to provide realizations of the uncertainty-aware visual analytics process. Those examples either represent a post hoc mapping to the uncertainty-aware visual analytics process or deliver design decisions for recreating an existing application using the guideline.

In the end, the research questions and their solutions are discussed, closing with open challenges and future research directions.

Contents

List of Figures 1

List of Tables 3

1 Introduction 5
- 1.1 Research Questions and Goals . . . 7
- 1.2 Contributions . . . 8

2 Theoretical Background 11
- 2.1 Visual Analytics . . . 11
- 2.2 Uncertainty . . . 12

3 Uncertainty-aware Visual Analytics 17
- 3.1 Related Work . . . 18
- 3.2 The Role of Uncertainty in Visual Analytics . . . 24
- 3.3 Process Definition . . . 26
- 3.4 Guideline on Application Creation . . . 41
- 3.5 Example Application . . . 51
- 3.6 Conclusion . . . 53

4 Uncertainty-aware Visual Analytics Application for Proteins 55
- 4.1 State of the Art and Requirements . . . 55
- 4.2 The Role of Uncertainty in Molecular Data . . . 56
- 4.3 Related Work . . . 60
- 4.4 Uncertainty-aware Description and Properties of Proteins . . . 62
- 4.5 An Application for Uncertainty-aware Visual Analytics of Proteins . . . 65
- 4.6 Results . . . 75
- 4.7 Perspective on the Uncertainty-aware Visual Analytics Process . . . 81
- 4.8 Discussion . . . 84

5 Images Measures as Uncertainty Predictors 89
- 5.1 Evaluation of Tumor Detection under Uncertainty . . . 90
- 5.2 Image Filtering Using Uncertainty Measures in Cinema Database . . . 99
- 5.3 Perspective on the Uncertainty-aware Visual Analytics Process . . . 111
- 5.4 Conclusion . . . 114

6 Topological Data Analysis and Potential Use-cases for Uncertainty 115
6.1 Related Work . . . 117
6.2 Preliminaries . . . 119
6.3 Method . . . 122
6.4 Results . . . 129
6.5 Performance . . . 134
6.6 Limitations . . . 140
6.7 Perspective on the Uncertainty-aware Visual Analytics Process . . . 140
6.8 Conclusion and Future Work . . . 141

7 Discussion and Conclusion 143
7.1 Research Questions and Goals . . . 143
7.2 Open Challenges and Future Work . . . 144

Bibliography 147

Curriculum Vitae 167

Overview of Utilized Tools 169

List of Figures

1.1 VA process by Keim et al. 5

2.1 Error and uncertainty of a measurand 14
2.2 Probability of a PDF 14

3.1 Sources of uncertainty in the VA process 24
3.2 The UAVA process 26
3.3 Guideline Step 1: Design a VA process 42
3.4 Guideline Step 2: Quantify system uncertainty 44
3.5 Guideline Step 3: Integrate and connect UA solutions 46
3.6 Guideline Step 4: Separate the VA process feedback loop 48
3.7 Guideline Step 5: Generate Provenance 49
3.8 UAVA application: Explainable artificial intelligence for lesion prediction with slice view and histogram 52
3.9 UAVA application: Explainable artificial intelligence for lesion prediction classification diagram 53

4.1 Protein backbone structure 57
4.2 UA description of atom positions 63
4.3 UAVA protein application overview 66
4.4 UAVA protein application scalar field computation 68
4.5 UAVA protein application 3D visualizations overview 71
4.6 UAVA protein application colored vs. gray contour 72
4.7 UAVA protein application 1CGT example 73
4.8 UAVA protein application 1G03 example 77
4.9 UAVA protein application 1B03 example closeup 80

5.1 UAVA brain tumor application workflow 90
5.2 UAVA brain tumor application pipeline 92
5.3 UAVA brain tumor application questionnaire result 93
5.4 UAVA Cinema Database application overview 99
5.5 UAVA Cinema Database application feature visualization 106
5.6 UAVA Cinema Database application line chart 107
5.7 UAVA Cinema Database application image viewer 110

6.1 PLMSS overview 116
6.2 PLMSS path compression 125
6.3 PLMSS decision tree marching triangles 126
6.4 PLMSS marching triangles 127
6.5 PLMSS marching tetrahedra 128
6.6 PLMSS noisy terrain example 130
6.7 PLMSS molecule example 132
6.8 PLMSS viscous fingers example 133
6.9 PLMSS Rayleigh-Taylor instability example 134
6.10 PLMSS Rayleigh-Taylor instability performance 135
6.11 PLMSS foot dataset performance 136

List of Tables

3.1 UAVA approaches comparison 23
3.2 UAVA process components and transformations 28

4.1 UAVA protein application user evaluation 84

6.1 PLMSS binary codes marching tetrahedra 127
6.2 PLMSS performance data 135

1 Introduction

The increasing amount of data required to be analyzed led to a novel data processing concept called visual analytics (VA). It became a standard tool for many applications to ease data access and knowledge generation, based on the idea that interactive visual interfaces and automated analysis techniques should support analytic reasoning. This allows users to find novel insights in datasets and to perform effective decision-making [142]. Keim et al. [136] described the VA process as a cyclic graph consisting of four major components (dataset, hypothesis, visualization, and insight), connected by functions to enable transformation and analysis of the given input dataset while creating new insights. It provides the basis model for this thesis that will be extended to allow the processing of data containing uncertainty. Figure 1.1 depicts the graph and transitions of the VA process.

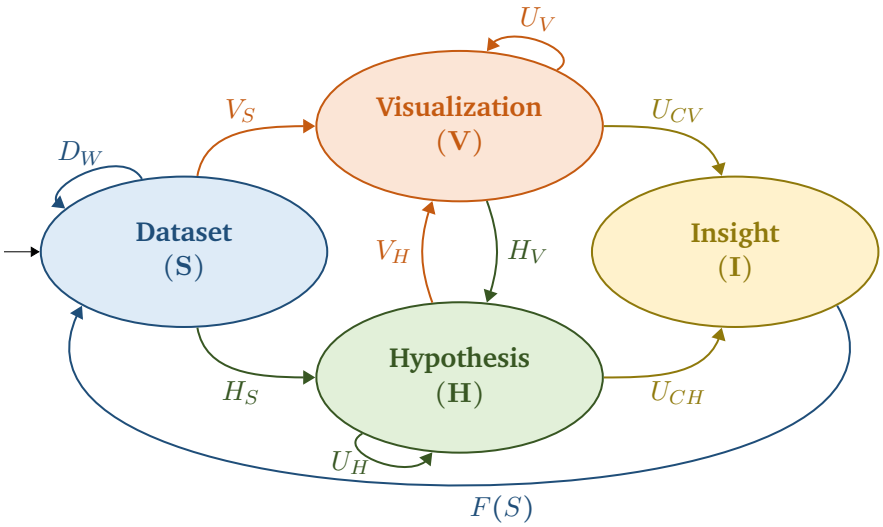


Fig. 1.1: The VA process introduced by Keim et al. [136]. It consists of four connected components (dataset, hypothesis, visualization, and insight), including a feedback loop that inserts insights back into the VA process. Data can be preprocessed D_W, transformed into a visualization V_S, or a hypothesis can be created H_S. Users can interact with U_V, derive insights from U_{CV}, or generate hypotheses from H_V visualizations. A hypothesis can be visualized V_H, specified further by the user U_H, and concluded to extract insight U_{CH}. Finally, the feedback loop $F(S)$ inserts the insights into the data to enhance it iteratively.

Many data sources used in the VA process are either augmented with uncertainty information or affected by uncertainty without determining its extent. When the user is unaware that the utilized system contains uncertainties, the chances of errors in the decisions are high [203]. Those errors can lead to wrong decisions that can be detrimental in some application areas, making communication of uncertainty essential to effective decision-making.

In real-world applications, data uncertainty stems from various sources such as incompleteness, imprecise measurements, reconstruction artifacts, or model imprecision [96]. On top of the uncertainty that stems from the dataset, the components in the VA process can introduce additional uncertainty that should be quantified to consider its influence. Each visualization hides or emphasizes certain data points, hypotheses simplify or approximate the underlying model, and human decision-making has biases towards previous or confirmatory results when extracting insights [95]. Those diverse sources of uncertainty in the VA process make its inclusion a difficult challenge.

Uncertainty itself is not trivial to consider in a VA application, as it does not simply pose another attribute or dimension that must be shown. Instead, it directly influences the data, transformations, and representations. For example, when preprocessing uncertainty-augmented images, uncertain pixels should have a lower impact on the result than trusted pixels. At the same time, when visualizing the preprocessed data, the user should be informed about uncertainty in the data (visualization of uncertainty) and the uncertainty that occurs when processing data (uncertainty of visualization) [27].

Sacha et al. [203] already stated that decision-makers cannot perform their tasks properly if uncertainty is not appropriately communicated, motivating further investigations into implementing uncertainty in VA applications. Keim et al. [135] stated that incorporating uncertainty into the VA process is an important challenge and "Dealing with uncertainty in VA is nontrivial", indicating that a systematic approach is necessary to guide VA application developers.

Although various approaches for creating uncertainty-aware visual analytics (UAVA) applications exist for specific application areas [46, 36, 132, 213], a process definition to describe the properties of a UAVA application and general guidelines towards mapping a VA application to a UAVA application are missing. Both challenges motivate the creation of this thesis.

1.1 Research Questions and Goals

Two research questions have been extracted that depict the lack of guidance in UAVA. "How do you define an uncertainty-aware visual analytics process?" **(RQ1)** and "Which steps are necessary to map a visual analytics process to an uncertainty-aware visual analytics process?" **(RQ2)**. As this poses multiple challenges, three goals for finding an answer to those questions are provided.

RG1 Define an adapted VA process that can handle uncertainy.

RG2 Provide instructions to map an existing VA application to a UAVA application.

RG3 Give examples of this mapping based on existing applications.

Uncertainty influences each component and function in the VA process, making its incorporation nontrivial. Just displaying and dragging along the uncertainty information in a dataset is not enough, as each function in the process alters the data and attached uncertainty information. Additionally, uncertainty influences the functions and changes the outcome to emphasize trustworthy information, still keeping less likely information in the context. This requires substantial changes to the VA process C defined by Keim et al. [136] to create a UAVA process C' **(RG1)**. Such a process definition can answer the first research question **(RQ1)**.

A process to check for compatibility can already steer the design process, but it does not provide a clear development pathway. Therefore, a guideline is invaluable to creating new applications, especially with a newly adapted and extended process. It should create a map from a VA application $c \in C$ to a UAVA application $c' \in C'$, posing a function $f : C \to C'$ **(RG2)**, answering the second research question **(RQ2)**.

Even though instructions provide a step-by-step view of the pathway to a new application, the description can be rather abstract to a reader from a certain application area. Mapping the instruction steps to a project will not always be straightforward, posing additional questions on implementation details. Here, example applications explicitly realizing the instructions $f : c \mapsto c'$ **(RG3)** are needed. Design decisions should be documented with direct correlation to the guideline steps to reveal the application development to the reader.

Solving these objectives is the ultimate goal of this thesis to provide a systematic workflow for the creation of UAVA applications. This workflow should contain a process definition to give a theoretical background model **(RG1)**, a guideline to steer users in the design process **(RG2)**, and example applications to give explicit realizations of these instructions **(RG3)**.

1.2 Contributions

This thesis provides its core contribution in Chapter 3, defining the UAVA process **(RG1)**, providing a guideline to create UAVA applications with detailed instructions **(RG2)**, and showcasing an example application **(RG3)**. The following chapters exhibit applications and algorithms that are either a UAVA tool (Chapter 4), use image measures for uncertainty quantification and similarity analysis (Chapter 5), or provide an algorithm to analyze scalar fields through topological data analysis (TDA) (Chapter 6).

First, the definition of the UAVA process C' extends the definition of the VA process C by additional and adapted components that allow to hold and handle uncertainty information. Those new components are connected by transformations that allow uncertainty propagation, aggregation, and quantification.

The guideline provides each step in creating a UAVA application, starting with creating a VA application. It can be seen as a function that maps a VA process to a UAVA process $f : C \to C'$. This whole system proposes a workflow that allows the extension of existing VA applications and the creation of new UAVA applications.

To promote this workflow, an example application from the area of explainable AI [221] provides a realization of the function $f : c \mapsto c'$ with $c \in C$ and $c' \in C'$. Here, physicians use neural networks to predict where a lesion, i.e., altered tissue, is located while receiving insights on the decision-making logic of the network by visual exploration. The application creation via the guideline is depicted in a post hoc mapping, giving references to each guideline step.

The following chapters describe UAVA applications in detail. Additionally, step-by-step application generation is simulated to exemplify the new UAVA workflow. This also includes potential future additions that are needed to successfully map the applications onto the UAVA process. Here, Molecule exploration and stability checking via 3D and 2D representations provide an interactive tool to analyze the positional uncertainty of atoms. Two other applications in tumor detection and image database querying use image measures as uncertainty predictors and similarity metrics.

At last, the piecewise linear Morse-Smale segmentation (PLMSS) showcases a TDA algorithm that can be employed to analyze uncertain data, delivering potential use cases for UAVA applications.

1.2.1 Author's Contributions

The work presented in this thesis is a result of joint work with many collaborators who put tremendous efforts and time into our projects. Therefore, I use the pronouns "we", "our", and "us" throughout the chapters where appropriate and only refer to "I" in places of my personal opinion. All publications used for the creation of this thesis are listed below, and my own contributions are explicitly stated:

1 R. G. C. Maack, H. Hagen, and C. Gillmann. "Evaluation of the Uncertainty-awareness Principle Using a Tumor Detection Scenario". In: Proceedings of the VisGuides: 2nd Workshop on the Creation, Curation, Critique, and Conditioning of Principles and Guidelines in Visualization (IEEE Vis). 2018.

In this project, a prototype created by Christina Gillmann was improved. I converted the project to a modular tool that can apply pipeline steps in an arbitrary order. Additionally, I designed and conducted the questionnaire together with Christina Gillmann.

2 R. G. C. Maack, C. Gillmann, and H. Hagen. "Uncertainty-Aware Ramachandran Plots". In: IEEE Pacific Visualization Symposium (PacificVis). IEEE, 2019.

This project started with a prototype by Christina Gillmann that I completely reimplemented. The final implementation was refined in close collaboration with Christina Gillmann.

3 R. G. C. Maack, D. Rogers, and C. Gillmann. "Exploring Cinema Databases using Multi-dimensional Image Measures". In: LEVIA. OSF Preprints, 2020.

This project was implemented in a research stay at the Los Alamos National Laboratory. The idea to use image measures from the tumor detection project arose in meetings with David Rogers and Christina Gillmann. I extended an existing project to implement an efficient querying task by using those image measures.

4 R. G. C. Maack, M. L. Raymer, T. Wischgoll, H. Hagen, and C. Gillmann. "A Framework for Uncertainty-Aware Visual Analytics of Proteins". In: Computers & Graphics, 2021.

In meetings with Michael Raymer, Thomas Wischgoll, Hans Hagen, and Christina Gillmann about the "Uncertainty-Aware Ramachandran Plots", we identified the need to add 3D visualizations and ensemble datasets to create a more versatile tool. I extended my implementation and discussed new features in frequent meetings with all collaborators.

5 R. G. C. Maack, G. Scheuermann, H. Hagen, J. T. H. Peñaloza, and C. Gillmann. “Uncertainty-aware Visual Analytics: Scope, Opportunities, and Challenges”. In: The Visual Computer, 2022.

Joint work with Gerik Scheuermann, Hans Hagen, Jose Tiberio Peñazola, and Christina Gillmann. Christina Gillmann and I came up with the concept and refined it together by reviewing insights taken from previous projects.

6 R. G. C. Maack, J. Lukasczyk, J. Tierny, H. Hagen, R. Maciejewski, and C. Garth. “Parallel Computation of Piecewise Linear Morse-Smale Segmentations”. In: IEEE Transactions on Visualization and Computer Graphics, 2023.

Joint work with Jonas Lukasczyk, Julien Tierny, Hans Hagen, Ross Maciejewski, and Christoph Garth. Jonas Lukasczyk proposed speeding up the Morse-Smale complex computation using path compression. I implemented the necessary filters in the TTK library and presented updates in frequent meetings with all collaborators.

7 R. G. C. Maack, F. Raith, J. F. Pérez, G. Scheuermann, and C. Gillmann."A Workflow to Systematically Design Uncertainty-Aware Visual Analytics Applications". The Visual Computer, Under review after a major revision.

Joint work with Christina Gillmann, Felix Raith, Juan Pérez, and Gerik Scheuermann. Christina Gillmann and I saw a lack of guidance for application creation under the usage of the UAVA process and developed the guideline. Felix Raith contributed insights from his experience of UAVA application creation.

Additionally, I contributed to the following publications:

(8) C. Gillmann, R. G. C. Maack, T. Post, T. Wischgoll, and H. Hagen. “An Uncertainty-aware Workflow for Keyhole Surgery Planning using Hierarchical Image Semantics”. In: Visual Informatics. Proceedings of PacificVAST 2.1 (2018), pp. 26–36.

(9) R. G. C. Maack, D. Saur, H. Hagen, G. Scheuermann, and C. Gillmann. “Towards Closing the Gap of Medical Visualization Research and Clinical Daily Routine”. In: The Eurographics Association, 2020.

(10) K. Schardt, R. G. C. Maack, D. Saur, H. Hagen, G. Scheuermann, and C. Gillmann. “Multi-modal Visualization of Stroke Lesion CT- Imaging”. In: LEVIA. OSF Preprints, 2020.

(11) C. Gillmann, R. G. C. Maack, F. Raith, J. F. Pérez, and G. Scheuermann. “A Taxonomy of Uncertainty Events in Visual Analytics”. In: IEEE Computer Graphics and Applications 43.5, 2023.

2 Theoretical Background

Both uncertainty and visual analytics (VA) are essential concepts to this thesis and must be understood before diving into the intermix of both in the upcoming chapters. Readers familiar with the VA process and the concept of uncertainty may proceed.

2.1 Visual Analytics

VA is a process of transforming input data into insights through visual abstractions and automated analysis algorithms, answering analyst-defined questions. It solves the problem of understanding complex data sets by providing a systematic approach that combines computational and visual methods. By using various visualization techniques and interactive interfaces, analysts can explore, analyze, and interpret complex data more intuitively and interactively. This process enables analysts to identify data patterns, trends, and relationships that may not be immediately apparent in raw data sets. Ultimately, VA helps analysts generate insights more quickly and accurately than traditional data analysis techniques, empowering them to make informed decisions.

The insight generation process in VA mostly comes in an iterative form, where data has to be enhanced in multiple intermediate steps, forming a complex pipeline of transformations. This iterative process has been formulated as a process by Keim et al. [136] (see Figure 1.1), providing a systematic description of the VA process, defining four components that transform data into hypothesis and visualization to create insights.

In the initial data acquisition step, data is collected from various sources, including databases, spreadsheets, and sensors. They are loaded and preprocessed D_W to prepare for analysis, including cleaning, filtering, and transforming tasks. Ensuring the data is accurate, relevant, and of high quality is critical for effective analysis.

Subsequently, the preprocessed data is examined using various visualization techniques V_S and algorithmic analysis steps H_S. The goal is to identify patterns, trends,

and relationships in the data that can provide insights into the underlying phenomena. Here, the visualizations, designed to help users understand and interpret the data more easily, are equipped with interaction options U_V to inspect the data representations further. Also, users can check algorithmic analysis results that may emphasize more tests U_H. Similarly, visualizations can hint towards using algorithmic analysis H_V and vice versa V_H.

Finally, the insights concluded from data analysis U_{CH} or human perception and reasoning U_{CV} are fed back into the data for further enhancement in the process $F(S)$. After multiple repetitions, it can be used to present insights to decision-makers or save it for later processing. This iterative process has been summarized in the VA mantra by Keim et al. [136]: "Analyse First - Show the Important - Zoom, Filter and Analyse Further - Details on Demand".

In summary, the VA process provides a model for transforming data into insights that inform decision-making. Integrating computational and visual methods enables analysts to explore, analyze, and interpret complex data more intuitively and interactively. Therefore, the VA process is a powerful tool for understanding complex data sets, combining computational and visual methods to help analysts process data in real-time, generating insights to power decision-making.

2.2 Uncertainty

Uncertainty is ubiquitous in the real world, where an action can result in a range of outcomes whose chances often cannot be easily predicted. When buying a raffle ticket, the chances of winning a certain amount of money, even though minimal, are often given by the provider. However, investing in the stock market is another situation with many unknowns and constantly changing economic conditions. As decisions are made on the available information, taking them as the only truth with full certainty can cause catastrophic outcomes. Here, knowing the boundaries of the information at hand allows decision-makers to take calculated risks instead of making vaguely supported decisions.

Therefore, uncertainty plays a crucial role in VA, providing a measure of trustworthiness to the data at hand. Here, uncertainty refers to the lack of knowledge or errors in the data. It can arise due to various reasons, such as measurement errors, sampling errors, or incomplete data, but also due to mapping data to visual representations [27] or user biases [225]. So, the data, the system, and the user

can introduce uncertainty in the VA process, making it difficult to implement and combine all those sources of uncertainty meaningfully.

One of the most common sources of uncertainty is measurement error, which occurs when the data is collected with instruments that have a certain resolution by nature. Another source of uncertainty is sampling error, which is introduced when the data is collected from a sample instead of the entire population. This can lead to biases in the data if the sample does not represent the population correctly. During data preprocessing, missing or incomplete data may require estimating or neglecting data points, introducing additional uncertainty. Moreover, data transformations and feature extraction techniques may introduce uncertainty due to the subjective nature of their implementation and the assumptions made in the process.

2.2.1 Uncertainty Modelling

Independent of the data source, task, and user of a VA application, datasets are usually acquired by measuring or simulating a phenomenon. Various mathematical models can be used to represent the uncertainty of the considered data, listing the representations relevant to this thesis below. In the following, error and uncertainty are differentiated first, where error refers to a situation of knowing the exact deviation from the actual value, using uncertainty where this is not possible.

Let a be a measurand and a^* be the true value of this measurand. When performing a measurement, the result a' represents the outcome. a^* and a' may be the same value, but their values most likely differ due to various effects. The difference between the true value and the measured value is known as the error e, which can be calculated using the equation $e = \|a^* - a'\|$. Quantifying an error requires a ground truth to determine the difference between the actual and measured values. Conversely, uncertainty indicates how reliable the measurement result is in representing the true value when an error cannot be computed. The relationship between these concepts is illustrated in Figure 2.1.

Uncertainty can take on different forms, depending on the nature of the data and the measurement process. Statistical uncertainty, for example, is often used to describe the distribution of possible outcomes based on the available data. This type of uncertainty is characterized by a probability density function (PDF) that represents the likelihood of different outcomes. Here, the peak of the PDF represents the most probable outcome. In some cases, multivariate data is used to generate statistical uncertainty. It refers to data that includes multiple variables, which can

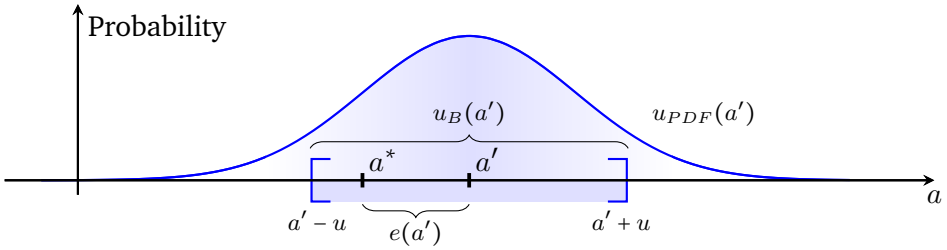


Fig. 2.1: Error e and uncertainty u of a measurand. The error requires a ground truth to be measured, whereas uncertainty captures the doubt about a measurement. This can be achieved by either bounded uncertainty $u_B(a)$ or probability distribution functions $u_{PDF}(a)$. (Image created using a TikZ template by Izaak Neutelings [174])

be used to generate a more complete picture of the data distribution by exhibiting multiple samples of the same phenomenon. Bounded uncertainty, on the other hand, provides a range within which the true value of the measurand is expected to lie without specifying a distribution. An example for bounded uncertainty and a PDF is given in Figure 2.1.

To define bounded uncertainty and PDF, let us assume a dataset that maps a set of n points in a d-dimensional space $\mathbb{D}$ to a set of output values. For simplicity, let us consider scalars

$$f_i(x_1, ..., x_d) = y_i, \; i = 1, 2, ..., n$$

Statistical uncertainty is often represented by a PDF, here called g, modeling the density of a random variable X. The probability of a result lying in the range between a and b (see Figure 2.2) is defined as

$$Pr(a \leq X \leq b) = \int_a^b g(f)df$$

where integrating over $\mathbb{D}$ has to sum up to 1 [27, 92]

$$\int_{\mathbb{D}} g(f)df = 1$$

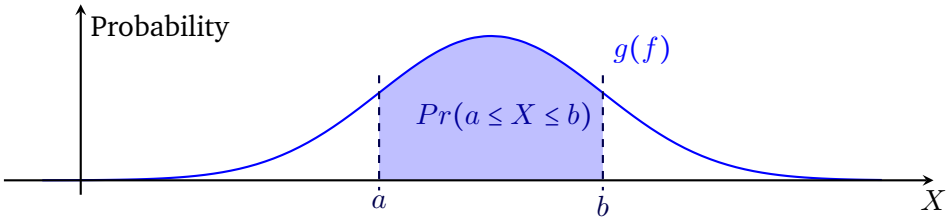


Fig. 2.2: Probability of a PDF, given by $g(f)$, between bounds $[a, b]$. (Image created using a TikZ template by Izaak Neutelings [174])

One common type of PDF is the Gaussian distribution [112], also known as the normal distribution. This distribution is defined by two parameters: the mean, which represents the most probable value, and the standard deviation, which describes the spread of the distribution.

Other types of PDF include the Poisson distribution, which models the probability of rare events occurring, and the binomial distribution, which is used to model the probability of a certain number of successes in a fixed number of trials.

Multivariate data resembles results where multiple realizations are present, for example, by computing an ensemble via multiple simulations or measuring multiple molecule conformations (see Section 4.4). This means that k values are present at each point, one for each realization.

$$f_i^j(x_1, x_2, ..., x_n) = y_i^j,\ j = 1, 2, ..., k$$

Interestingly, multivariate data can be converted to a PDF, for example, by kernel density estimation, at the loss of information.

Bounded uncertainty provides a fixed range within which the true value of a measurand is expected to lie [180]. This range is usually defined by two values, a lower bound and an upper bound, say $[L, H]$. The exact value of the measurand is assumed to be within this range $L \leq a^* \leq H$, and there is no probability distribution associated with it. This type of uncertainty is often used when there is not enough information to estimate the statistical uncertainty or when the data itself is bounded, for example, by physical constraints.

2.2.2 Quantification, Propagation, and Aggregation of Uncertainty

This section provides the definitions of uncertainty quantification, propagation, and aggregation since they pose important operations when dealing with uncertainty.

Quantification of uncertainty is the process of assigning a numerical value to the degree of uncertainty associated with a particular measurement or prediction. This involves estimating the likelihood of different outcomes and the range of possible values associated with each outcome. The goal is to provide decision-makers with a clear understanding of the level of uncertainty associated with a particular situation or decision, enabling them to make more informed choices.

Propagation of uncertainty refers to how uncertainties in input data or parameters propagate through a system or process to affect the uncertainty of the output. When

input data or parameters have uncertainties associated with them, these uncertainties can be propagated through a mathematical or computational model to determine the resulting uncertainty in the output. This is important in many fields where accurate predictions or decisions require an understanding of the uncertainties involved. The propagation of uncertainty can be calculated using various methods, including Monte Carlo simulations and sensitivity analysis. In the adapted VA process, uncertainty propagation is used in most functions to consider the uncertainty introduced in the function's execution.

Aggregation of uncertainty refers to the process of combining multiple sources of uncertainty into a single representation. This can be done through statistical techniques such as variance or standard deviation or simple aggregation functions (e.g., minimum, maximum, average). It can be used to summarize multiple contributions of uncertainty into a single measure.

3 Uncertainty-aware Visual Analytics

The increasing amount of data required to be analyzed led to a novel data processing concept called visual analytics (VA). It developed into a standard tool in many applications to ease data access and knowledge generation, allowing to find novel insights in datasets and perform effective decision-making [142] by facilitating analytical reasoning by interactive visual interfaces [44]. Many data sources examined in VA applications exhibit data with either available uncertainty information or unquantified uncertainty that may be introduced by various sources, e.g., noise or missing values. In addition, uncertainty can be introduced in the VA process by automatic analysis methods or human perception influencing the knowledge generation process. Ignoring this influence can lead to vaguely supported decisions that stem from untrustworthy data.

As described in Section 2.1, Keim et al. [136] described the VA process as a graph consisting of four major components (dataset, hypothesis, visualization, and insight) (see Figure 1.1). These components are connected with each other by functions to enable the transformation and analysis of the given input dataset while creating new insights. Still, this classic VA process does not provide mechanisms to handle uncertainty.

The role of uncertainty in the VA process has been described by Sacha et al. [203] with the main statement that decision-makers cannot perform their tasks properly if uncertainty is not appropriately communicated. Keim et al. [135] stated that incorporating uncertainty into the VA process is one of the major challenges in this research area. By now, many applications have started incorporating uncertainty analysis in their VA tools, but an overall definition of a uncertainty-aware visual

This chapter has been partially published in the following articles:

- R. G. C. Maack, G. Scheuermann, H. Hagen, J. T. H. Peñaloza, and C. Gillmann. “Uncertainty-Aware Visual Analytics: Scope, Opportunities, and Challenges”. In: The Visual Computer, 2022.
- R. G. C. Maack, F. Raith, J. F. Pérez, G. Scheuermann, and C. Gillmann."A Workflow to Systematically Design Uncertainty-Aware Visual Analytics Applications". The Visual Computer, In Revision.

analytics (UAVA) process is missing. Additionally, guidelines that assist visualization researchers in identifying the necessary development steps are sparse and only available in specific application areas.

This motivated the creation of a uncertainty-aware visual analytics (UAVA) process C' that explains the sources and effects of uncertainty in the VA process C by revisiting and extending the definition of Keim et al. [136]. It defines all necessary components and functions to include uncertainty in the VA process to achieve uncertainty-awareness, allowing users to map their application onto this definition to check for compatibility.

In its entirety, this formulates a UA description of the VA process. A guideline provides the necessary steps to map an existing VA application $c \in C$ to a UAVA application $c' \in C'$, proposing a function

$$f : \begin{cases} C & \longrightarrow & C' \\ c & \longmapsto & c' \end{cases} \tag{3.1}$$

After the introduction, Section 3.1 exhibits related work with similar approaches and guidance in creating UAVA applications. Then, the role of uncertainty in each of the VA components is described in Section 3.2, providing possible sources of uncertainty. The novel UAVA process is described in Section 3.3. To ease the creation of UAVA applications, Section 3.4 provides a guideline, followed by an example application in Section 3.5 and concluding remarks in Section 3.6.

3.1 Related Work

In VA, multiple definitions have been developed, many of which do not handle uncertainty systematically. Still, UA visualization approaches have been created, leading to the inclusion of UAVA in specific scenarios. Also, related disciplines, such as sensitivity analysis, enabled similar visual analysis approaches that hint towards successfully including uncertainty in VA. Those approaches have been selected according to the keywords of the presented work (VA, uncertainty analysis, and UA visualization). Articles that contain at least two of these keywords are included in this review.

Therefore, Section 3.1.1 summarizes the most essential definitions in VA. Important UA visualization techniques are summarized in Section 3.1.2. Approaches targeting the inclusion of UAVA for specific scenarios are outlined in Section 3.1.3. Applications

and similar definitions are showcased in Section 3.1.4. Approaches giving guidelines for creating UA visualization (Section 3.1.5) and VA approaches (Section 3.1.6) are described next. At last, Table 3.1 in Section 3.1.7 provides an overview of similar approaches that develop VA concepts, describe UAVA approaches systematically or present considerations on UA concepts in a general form.

3.1.1 Definition of Visual Analytics

The definition of VA has been constantly developed throughout the years. Thomas and Cook [44] developed the term VA and highlighted the need for systematic development in this area. Keim et al. [134] converted the idea into a systematic description of the VA process while defining four components that transform data into hypothesis and visualization to create insight. This insight can be fed back into the process as novel data. Sacha et al. [204] further separated and examined the feedback loop. This resulted in the knowledge generation model in VA. Further, Andrienko et al. [5] interpreted the term VA as model building, refining the original definition of Keim et al. They provide a systematic description of the VA process, forming a starting point for developing a systematic approach to handling uncertainty in VA.

MacEachren [161] stated that a classic visualization approach is insufficient to deal with uncertainty. They proposed that UAVA is required as it provides the user with an approach to tackle data not restricted to visualization. We use this statement to motivate the presented description of UAVA.

3.1.2 Uncertainty-aware Visualization

The VA process highly depends on the contained visualizations, forming the primary components of visual interaction and knowledge extraction.

Surveys on uncertainty visualization are manifold, playing a major role when designing the visualization component. Potter et al. [191] achieved a taxonomy of uncertainty visualization. Brodlie et al. [27] followed a similar classification scheme to provide a general taxonomy. Based on previous work, Bonneau et al. [19] presented a STAR on uncertainty visualization, forming a basis for Jena et al. [125] to build an online browsing tool to explore several UA visualization approaches. Here, uncertainty visualization approaches were classified based on the underlying data. Olston et al. [180] presented a STAR report regarding the visualization of bounded

uncertainty, whereas Hullman et al. [119] presented a STAR report targeting the evaluation of uncertainty visualization.

Kamal et al. [130] summarized recent challenges in uncertainty visualization. The challenges are presented in a structured manner, leading to a summary of open challenges mainly centered around the inclusion of the user in the visualization process. Bhatt et al. [17] provided guidelines that suggest using uncertainty visualization. Still, these findings need to be transferred to VA. Schulz [210] summarized his discoveries on identifying, modeling, and visualizing uncertainty.

These approaches are a helpful starting point for the presented research area and give insights into the design of the visualization component in UAVA.

3.1.3 Uncertainty-aware Visual Analytics

Various specialized descriptions of UAVA and related application areas exist. Still, a systematic workflow to create a UAVA application is missing.

UAVA approaches have been designed for multiple data types and computational models such as multi-variate time-series [21], principal component analysis [93], merge-trees [251], moving object detection [116] or tensor analysis [74].

Many VA applications have been developed and applied based on the classic VA process definition by Keim et al. [136]. Still, the field of VA holds a set of open problems. One is the proper quantification, communication, and visualization of uncertainty in the VA process.

UAVA has tackled specific application scenarios such as keyhole surgery assistance [80], analysis of Choropleths [118], and wildfire analysis [192]. Although these approaches show a suitable UAVA approach in the specific target area. therefore, generalizing these approaches to a systematic method to include uncertainty in the VA process is non-trivial.

Other applications of UAVA include medicine [82, 84, 83], biochemistry [160], environmental sciences [194], urban planning [66], mechanical engineering [145] and digital humanities [137, 123]. These solutions are highly specialized and cannot be easily generalized. But, they give valuable insights into creating UAVA tools in general that can help extract application domain-specific guidelines.

Sacha et al. [203] formulated requirements that need to be fulfilled to obtain a UA visualization. Their suggestions include uncertainty quantification, propagation,

visualization in each component, and suitable interaction with UA visualization. These requirements will be used to adapt the classic VA process in this work.

Correa et al. [46] provide a framework for modelling uncertainty in VA by Gaussian Mixture Models, giving an example model to implement the requirements to make the VA process UA. They hint at constructing a UAVA process in a specific case where Gaussian mixture models can represent the uncertainty information. In contrast to the presented work, we aim to provide an adapted and extended VA process that incorporates the suggestions by Sacha et al. [203].

Karami [132] provided a UAVA process that allows the processing of large datasets. Their work includes precise descriptions of each component in the VA process when considering big data. Similarly, Senaratne [213] described the role of uncertainty for spatio-temporal data and presented solutions for UA image-based volunteered graphic information, exploration of location-based mobile communication data, and bi-dimensional numerical data. Both works pose a limitation that neglects further flavors of data to be targeted, making them a great starting point for designing a general UAVA process.

3.1.4 Uncertainty-awareness in Related Disciplines

Uncertainty-awareness is highly related to other disciplines, including sensitivity analysis or ensemble visualization. These approaches will be reviewed and used to define starting points for defining parts of the UAVA process. Search terms related to uncertainty, as shown in the taxonomy by Rocha et al. [201], have been used to find such approaches.

VA in the context of ensemble datasets is highly related to the presented topic, as ensemble data can be transformed into uncertainty data (at the loss of information) and vice versa. Wang et al. [245] provided a state-of-the-art analysis for VA of ensembles and showed that VA approaches could achieve a suitable communication of variability in an ensemble.

Liu et al. [155] showed that data quality is an important aspect that needs to be monitored in the VA process. In their work, they provided a mechanism that extends the VA process to enhance data quality and create awareness of data flaws. This provides a strong link between data quality and data uncertainty.

Sensitivity analysis [166] is highly related to uncertainty analysis, as this discipline examines the effect of changing input variables on the output variable(s). Especially

in machine learning, VA approaches are derived to conduct sensitivity analysis [228]. This also highly relates to uncertainty analysis, as uncertainty expresses the variability of parameters in a system. Here, sensitivity analysis will be used in the presented work if applicable.

3.1.5 Workflows and Guidelines for Uncertainty Visualization

Many approaches that give guidelines for creating UA visualization are available. These guidelines must be adapted and extended to design UAVA applications.

Haegele et al. [110] provided several examples of workflows for applications such as hierarchical graphs and texts. Padilla et al. [181] summarized the potential design space for uncertainty visualization and provided existing visualization theories, which aim to minimize the large design space.

Frequency framing [76] is based on the idea that uncertainty is more intuitively understood in a frequency framing (e.g., 1 out of 10). Further, attribute substitution [129] allows viewers to substitute uncertainty information for data that are easier to understand mentally. Visual boundaries [182] separate a continuous space into a bounded space for easier recognition. At last, visual semiotics [162] provide visual variables that are suitable to indicate uncertainty. As these guidelines provide insights for visualizing uncertainty, those findings can help create the visualization component in UAVA applications.

3.1.6 Workflows to Generate Visual Analytics

Multiple works that address workflows and guidelines to create visualization or VA approaches can be identified. To tackle uncertainty systematically, they must be extended and adapted to receive a workflow for creating UAVA applications.

Gleicher et al. [88] defined the problem space to design visualization. Their approach defines six axes along the problem space that must be considered while designing visualization. This gives a good starting point for visualization design in the VA process. Booth et al. [20] provided a visual analytic space based on user goals without handling uncertainty in this process. We will use this as a starting point in the presented work.

Conlen et al. [42] provided VA design principles that are mainly data-driven. They suggest how data should be visualized and which interactions should be provided.

Tab. 3.1: Approaches related to the UAVA. Theory summarizes work that aims to develop general concepts of VA. Systematic approaches aim to describe UAVA systematically. UA describes general considerations of UA concepts in general. None of the related approaches can provide a theoretical and systematical description of UAVA.

Work	Theory	Systematic	UA
Thomas and Cook [44]	✓	✗	✗
Sacha et al. [203]	✓	✗	✗
Correaet al. [46]	✓	✗	✗
Keim et al. [134]	✗	✓	✗
Sacha et al. [204]	✗	✓	✗
Andrienko et al. [5]	✗	✓	✗
Bors et al. [21]	✗	✗	✓
Görtler et al. [93]	✗	✗	✓
Yan et al. [251]	✗	✗	✓
Höferlin et al [116]	✗	✗	✓
Gerrits et al. [74]	✗	✗	✓
Potter et al. [191]	✓	✓	✗
Brodlie et al. [27]	✓	✓	✗
Bonneau et al. [19]	✓	✓	✗
Jena et al. [125]	✓	✓	✗
Olston and Mackinlay [180]	✓	✓	✗
Kamal et al. [130]	✓	✓	✗
Bhatt et al. [17]	✓	✓	✗
Hullmann et al. [119]	✓	✓	✗
Karami [132]	✗	✓	✓
Senaratne [213]	✗	✓	✓
Gillmann et al. [82]	✗	✓	✓
Gillmann et al. [84]	✗	✓	✓
Gillmann et al. [83]	✗	✓	✓
Maack et al. [160]	✗	✓	✓
Raith et al. [194]	✗	✓	✓
Güell et al. [66]	✗	✓	✓
Kretzschmar et al. [145]	✗	✓	✓
Khulusi et al. [137]	✗	✓	✓
Jänicke et al. [123]	✗	✓	✓
Wang et al. [245]	✓	✗	✓
Liu et al. [155]	✓	✗	✓
Saltelli et al. [166]	✓	✗	✓
Leffrang and Müller [150]	✓	✗	✓

Federico et al. [64] provided a nested VA design model based on a user-task-data model. A VA system must be described to generate it.

There exist (open) software solutions that aim to guide users through the creation process of VA, such as *RapidMiner*, *ParaView* and *Tableau*. They assist users in designing prototypes, allowing users to implement workflows manually, and can play an important role in developing ideas for UAVA applications.

3.1.7 Overview of Similar Approaches

Various approaches deal with uncertainty concerning VA, summarizing a selection in Table 3.1. It outlines which approaches provide a theoretical foundation in VA, enable a systematic approach independent of the application, or consider uncertainty-awareness. No approach provides a theory that describes a systematic workflow with a process definition for UAVA, motivating the creation of such a workflow to fill this research gap.

3.2 The Role of Uncertainty in Visual Analytics

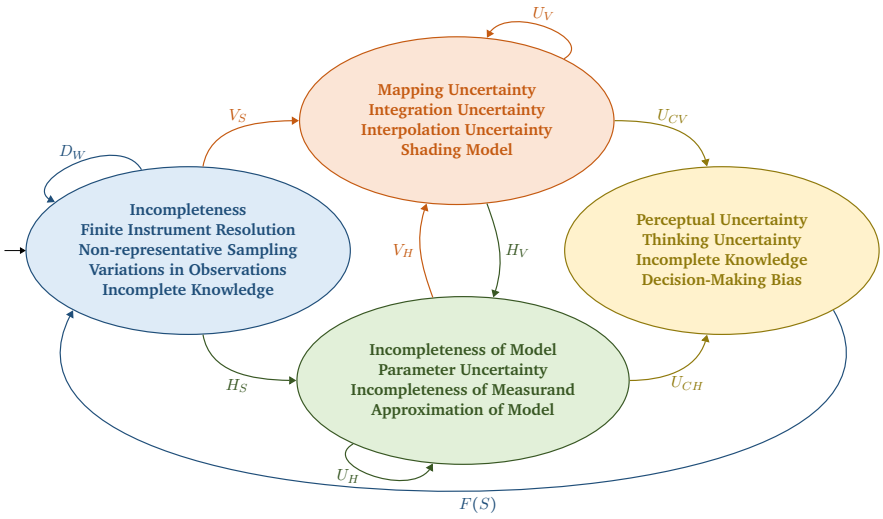


Fig. 3.1: Sources of uncertainty in the VA process.

Keim et al. [135] proposed that including uncertainty in the VA process is non-trivial. This comes from the large variety of uncertainty sources that can be present in each component of the process. This section summarizes these sources to create a basis for the required adaptations in the VA process to make it UA.

Each main component of the VA process can introduce uncertainty, as depicted in Figure 3.1. Uncertainty can arise from various sources, and it is important to distinguish between them. One type of uncertainty is epistemic uncertainty, which is based on the underlying model. Another type of uncertainty is aleatoric uncertainty, which results from variations in the measurement outcome when conducting an experiment multiple times. Finally, there is subjective uncertainty, which originates from humans interacting with the VA system. Understanding the different types of uncertainty and their origins is crucial in developing effective strategies for addressing them.

Data Starting from the input dataset, uncertainty can be present in the input data that can stem from various effects like data incompleteness, finite instrument resolution, non-representative sampling, variations in observations, and incomplete knowledge about the measurand [22]. For example, medical blood sugar measurements over time often require the patients to test regularly, resulting in missing measurements in some cases.

Hypothesis When considering the Hypothesis component, uncertainty can be introduced by parameter uncertainty. This means that computational models often require parameters, which can be hard to find in many cases, or it is hard to determine if a chosen parameter is optimal [163]. By model, we refer to the computational theory used to transform the input data. Furthermore, the computational model introduces uncertainty into the VA process due to simplifications made in their realization. Models are incomplete or approximate physical behavior of natural phenomena by definition. As our knowledge of the world and computational power is limited, hypothesis forming is affected by uncertainty [70]. An example would be a simulation of stiffness in a particular material where boundary conditions must be set, as not all physical behavior in the world can be modeled.

Visualization In terms of visualization, uncertainty can be introduced by mapping visual variables, hiding or highlighting certain information, and the resolution of the display device [165]. Also, users reviewing the shown visualization can introduce uncertainty into the VA process that stems from perceptual uncertainty, memory uncertainty, and thinking uncertainty [63]. For example, users might fail to perceive depth in a volume rendering due to inappropriate visualization approaches.

Insight At last, uncertainty can be introduced into the VA process while forming insights. Here, users can introduce uncertainty through a decision-making bias. They may ignore VA results, as previous results might favor a different outcome. The experience and knowledge of domain experts can also introduce uncertainty into the VA process due to unexpected results that are out of their expertise [242].

This also relates to further disciplines, such as cognition theory and psychology, as uncertainty in the human component is hard to handle. Enke and Graeber [63] provided a theoretical framework to address this issue and performed experiments on how humans deal with uncertainty. Their findings indicate that uncertainty is perceived differently depending on the decision maker. Still, the authors suggest that uncertainty needs to be communicated and discussed to allow secure decision-making.

Please note that each application scenario features a different set of uncertainty sources that have to be extracted to handle them appropriately. Also, cases exist where a specific source of uncertainty may be present but is neglected as its influence is too small or expensive to compute. This decision highly depends on the use case, data source, and user groups. Still, multiple sources of uncertainty are likely introduced into a specific implementation of the VA process and should be handled accordingly.

3.3 Process Definition

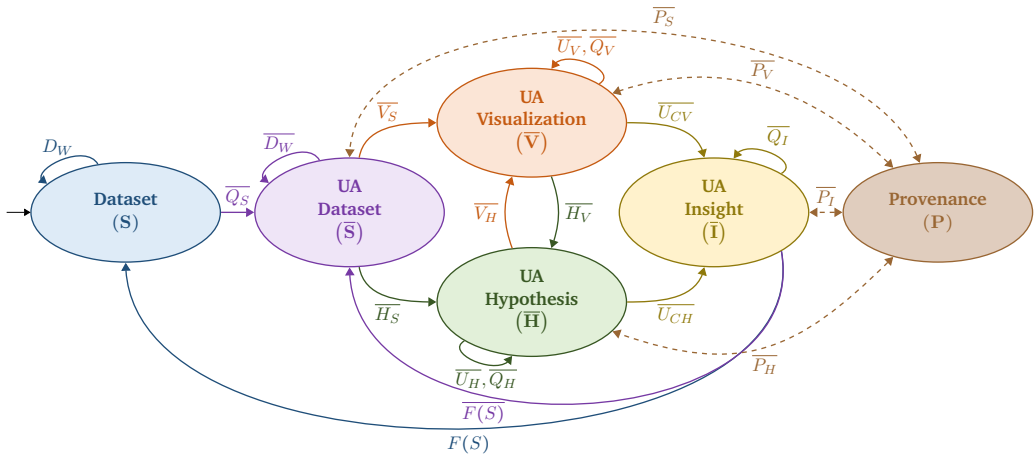


Fig. 3.2: The UAVA process. A UA dataset $\overline{S}$ can be created starting from an input dataset S by quantifying its uncertainty. This $\overline{S}$ can be utilized to create a UA hypothesis $\overline{H}$ or a UA visualization $\overline{V}$. Based on these analysis techniques and user interaction, UA insights $\overline{I}$ can be generated. They can be fed back into S or $\overline{S}$. The provenance component P keeps track of any changes in the pipeline, allowing the supervision of the amount of uncertainty and utilized functions when executing the process.

This section defines the UAVA process that provides the components and functions available within a UAVA application. It delivers a formal model for those applications, allowing to check if it conforms to the definition.

This definition includes two crucial adaptations to the traditional VA process [136]. First, all existing components and functions in the VA process must be extended or adapted to incorporate uncertainty information. Second, the existing traditional VA process does not hold mechanisms to insert uncertainty knowledge into the VA process and keep track of them, which means that there are missing components and functions in the classic VA process that need to be added.

To make this transformation more understandable, a real-world example from the domain of medicine [80], proposing a UA keyhole surgery to plan the removal of a brain tumor, is utilized. The task of this tool is to find a secure way through the patient's head under an uncertain segmentation. The goal is to avoid important parts of the brain that would result in the loss of human abilities as much as possible.

As a reference, the VA process definition by Keim et al. [136] is depicted in Figure 1.1. Adapting the process to be UA is shown in Figure 3.2, listing all components and functions in Table 3.2. The classic VA process by Keim et al. [136] is composed of four components:

- Dataset (■)
- Hypothesis (■)
- Visualization (■)
- Insight (■)

The components are connected by operations required in the VA process. These operations are encoded as functions between components and are defined as functions that transform one component into another. They are colored consistently by their codomain to show where the functions map the current data.

All functions originating from the classic VA process will be marked by a box (■) in the respective color of their category. Two novel components and several functions are introduced to describe a complete UAVA process. Namely, the novel components are:

- UA Dataset (▲)
- Provenance (▲)

The novel functions include uncertainty quantification and provenance generation concerning the existing components and functions in the VA process. All novel elements will be marked by a triangle (▲) in the respective color throughout the entire manuscript for smooth reading. Additionally, functions creating or handling UA data are marked by a line over the function (e.g., $\overline{Q_S}$)).

Tab. 3.2: UAVA process components and transformations overview.

Dataset S	
D_W ■	Preprocessing
$F(S)$ ■	Feedback Loop

UA Dataset $\overline{S}$	
$\overline{Q_S}$ ▲	Uncertainty Quantification
$\overline{D_W}$ ▲	UA Preprocessing
$\overline{F(S)}$ ▲	UA Feedback Loop

UA Hypothesis $\overline{H}$	
$\overline{H_S}$ ■	Generation from UA Dataset
$\overline{H_V}$ ■	Generation from UA Visualization
$\overline{Q_H}$ ▲	Uncertainty Quantification
$\overline{U_H}$ ■	User Interaction

UA Visualization $\overline{V}$	
$\overline{V_S}$ ■	Generation from UA Dataset
$\overline{V_H}$ ■	Generation from UA Hypothesis
$\overline{Q_V}$ ▲	Uncertainty Quantification
$\overline{U_V}$ ■	User Interaction

UA Insight $\overline{I}$	
$\overline{U_{CV}}$ ■	Conclude UA Visualization
$\overline{U_{CH}}$ ■	Conclude UA Hypothesis
$\overline{Q_I}$ ▲	Uncertainty Quantification

Provenance P	
$\overline{P_S}$ ▲	Generation for UA Datasets
$\overline{P_V}$ ▲	Generation for UA Visualization
$\overline{P_H}$ ▲	Generation for UA Hypothesis
$\overline{P_I}$ ▲	Generation for UA Insight

3.3.1 Uncertainty-aware Visual Analytics Process

This section provides a formal description of the UAVA process that is kept similar to the definition of the VA process by Keim et al. [136] to allow an easy mapping. Each function is briefly described, but detailed descriptions are provided in the following sections.

To start, m datasets $S = S_1, ..., S_m$ are chosen with each $S_i, i \in (1, ..., n)$ consisting of attributes $A_{i1}, ..., A_{ik}$. The UAVA process is a transformation $G : S \rightarrow \overline{I}$ that aims to generate UA insight from the input data. The insights $\overline{I}$ can be directly concluded from visualizations $\overline{V}$ or by automated analysis methods results from hypothesis $\overline{H}$. G itself is a concatenation of functions
$g \in \{D_W, \overline{D_W}, \overline{Q_S}, \overline{Q_H}, \overline{Q_V}, \overline{Q_I}, \overline{H_S}, \overline{H_V}, \overline{V_S}, \overline{V_H}, \overline{U_V}, \overline{U_H}, \overline{U_{CH}}, \overline{U_{CV}}\}$.

D_W and $\overline{D_W}$ describe (UA) preprocessing steps $W \in T, C, S, I$ that can be data transformation functions $D_T, \overline{D_T}$, data cleaning functions $D_C, \overline{D_C}$, data selection functions $D_S, \overline{D_S}$, or data integration functions $D_I, \overline{D_I}$. The quantification of uncertainty in the data component $\overline{Q_S} : S \rightarrow \overline{S}$ adds an uncertainty quantification to each attribute to form a tuple $(A_{ij}, \overline{A_{ij}}), j \in (1, ..., k)$.

Now that an uncertainty description is present, the data can be visualized $\overline{V_S} : \overline{S} \rightarrow \overline{V}$ or a hypothesis can be generated $\overline{H_S} : \overline{S} \rightarrow \overline{H}$. When creating a visualization or hypothesis, the usage of them may introduce additional uncertainty $\overline{Q_H} : \overline{H} \rightarrow \overline{H}$, $\overline{Q_V} : \overline{V} \rightarrow \overline{V}$. Also, a visualization can be used to create a hypothesis $\overline{H_V} : \overline{V} \rightarrow \overline{H}$ and visualizations of a hypothesis can be created $\overline{V_H} : \overline{H} \rightarrow \overline{V}$.

Furthermore, user interactions steer the process by changing a visualization via controls $\overline{U_V} : \overline{V} \rightarrow \overline{V}$ or creating a new hypothesis from another hypothesis $\overline{U_H} : \overline{H} \rightarrow \overline{H}$. Also, users can conclude insights from visualizations $\overline{U_{CV}} : \overline{V} \rightarrow \overline{I}$ or hypothesis $\overline{U_{CH}} : \overline{H} \rightarrow \overline{I}$.

The insights gained from this process can be fed back into the (UA) dataset to restart the process and save the progress $F(S)$, $\overline{F(S)}$.

The functions $\overline{P_S}, \overline{P_V}, \overline{P_I}, \overline{P_H}$ are saving the processing steps and uncertainty information present after each process function $P_X : P \rightarrow P$ to keep track of levels of uncertainty and analysis steps for further analysis or steering the process.

3.3.2 Dataset S

The dataset component holds a number of datasets from heterogeneous data sources. Each dataset holds a number of attributes that define its content. They can be generally defined as a function with various relations normally used to differentiate various types of data, e.g., attributes aligned on a grid are usually referred to as image or scalar data.

Based on the respective problem description, a dataset S is loaded to be analyzed in the VA process.

Preprocessing D_W ■

The classic VA process allows processing of the input dataset by four different operations: Data Transformation D_T, Data Cleaning D_C, Data Selection D_S, and Data Integration D_I. Until no uncertainty definition or quantification has been performed, these operations can be applied as defined in the classic VA process [136].

Although data preprocessing is an important or even indispensable step in the VA process, applying it before uncertainty definition and quantification have been achieved [30] is not recommended. When preprocessing operations change attributes, the data uncertainty levels may change, resulting in a less reliable uncertainty quantification.

3.3.3 Uncertainty-aware Dataset $\overline{S}$

By quantifying the uncertainty $\overline{Q_S}$ in the input dataset S, a UA dataset $\overline{S}$, representing the data and its uncertainty, is created.

First, the uncertainty of an attribute is required. So each dataset is extended such that its attributes $A_{i1}, ..., A_{ik}$ are replaced by tuples $(A_{i1}, U_{i1}), ..., (A_{ik}, U_{ik})$, where U_{xy} can hold any model of uncertainty, e.g., a probability density function (PDF) or bounded uncertainty.

Attributes can be single measurements but also contain entire datasets (large and complex data). This means that combinations such as multi-field or ensemble datasets are explicitly possible. This also allows the quantification of uncertainty of a single dataset and aggregation of uncertainty to summarize multiple attributes.

Uncertainty Quantification $\overline{Q_S}$ ▲

Proper uncertainty quantification depends on the data format, application, and task the user needs to fulfill. It holds for each record and its attributes in a dataset. Various datasets, such as molecular data, are acquired in conjunction with uncertainty quantification that can be directly interpreted. In this case, the uncertainty quantification of the input dataset is given explicitly, where the uncertainty information and related data just have to be connected.

Multiple models can be exhibited for quantifying uncertainty, for example, a PDF, bounded uncertainty, or Gaussian mixture models [46], where the form strongly depends on the underlying data and domain. Halpern [111] provides an introduction to representing uncertainty.

Hands-On Example: In the hands-on example, the dataset consists of magnetic resonance imaging of a patient's brain that contains a tumor. Therefore, each attribute, here the pixels, consists of a scalar value that needs uncertainty quantification. Each pixel is assigned to a vector of a set of image metrics results [78], where the vector's magnitude can be used as an aggregated uncertainty quantification.

Uncertainty-aware Preprocessing $\overline{D_W}$ ▲

Once a UA dataset is achieved, preprocessing operations can be applied to transform the dataset into a format that allows the creation of hypotheses or apply visualization approaches. Here, data transformation $\overline{D_T}$, data cleaning $\overline{D_C}$, data selection $\overline{D_S}$ and data integration $\overline{D_I}$ are available, as defined in the original VA process. Still, they must be adapted to be UA.

The transformation of data is concerned with the application of mathematical functions to describe the transformation. As we consider UA datasets in the UAVA process in the form $(A_{i1}, U_{i1}), ..., (A_{ik}, U_{ik})$, we require mathematical operations that can be applied in this setting.

In the classic VA process and in most other data analysis scenarios, datasets are cleaned, selected, and integrated to provide a stable dataset that can be processed. When considering data cleaning, we propose two important adaptations in this process: Do not eliminate any captured data point and merged data points, including their uncertainty.

When eliminating a data point, the information, no matter how uncertain it is, is neglected in the VA process. No matter how well selected these points are, the

selection is based on a hypothesis or metric that could be wrong or incomplete. To avoid this, we propose to find a suitable uncertainty quantification that assigns very high uncertainty to the selected data point.

The merging of data points arises when a phenomenon is captured in the data multiple times. Here, data points are merged to avoid multiple occurrences of the same phenomena in the dataset. In this case, one must not only merge the data points. In addition, the uncertainty of the data points needs to be merged as well, resulting in an aggregation of uncertainty.

Hands-On Example: Images from Computer Tomography or Magnetic Resonance Imaging require further processing. Histogram equalization enhances the underlying image. In this example, the histogram equalization is not solely manipulating the image itself but also the underlying uncertainty quantification. The result is a contrast-enhanced MRI with an included uncertainty quantification.

3.3.4 Uncertainty-aware Hypothesis $\overline{H}$

A hypothesis is a supposition or proposed explanation based on limited evidence as a starting point for further investigation. The null hypothesis is usually utilized to achieve this, claiming the lack of any connection between the measured effect and the data. Hypotheses are formed and tested, resulting in them being either rejected or failing to be rejected.

In the classic VA process, the component hypothesis H is described as a general tool to create insight or knowledge based on automated analysis algorithms. When considering hypotheses that are based on UA datasets, a UA hypothesis $\overline{H} = (H, u(H))$ needs to be defined. Here, $u(H)$ describes the attached uncertainty. The UAVA process shows that the hypothesis can be further built from visualization. Still, the visualization can be considered as an input dataset that allows the creation of a hypothesis. Here, $u(H)$ describes a confidence value for the formulated hypothesis. This means that whatever the output of a statistical analysis method is, a result comprises the derived hypothesis H and an uncertainty quantification $u(H)$ of the generated hypothesis. The generation of a UA hypothesis and possible interaction methods will be shown in the following.

Generation from Uncertainty-aware Dataset $\overline{H_S}$ ■

The generation of a UA hypothesis $\overline{H}$ can be described by a function starting from two sources: a UA dataset ($\overline{H_S} : \overline{S} \rightarrow \overline{H}$) and a UA visualization ($\overline{H_V} : \overline{V} \rightarrow \overline{H}$). The latter is part of the hypothesis-visualization-subcycle of VA that will be discussed in Section 3.3.6.

In the classic VA process, the generation of hypothesis H can be based on a dataset utilizing a set of automated analysis algorithms. These statistical operations need to be redefined to provide a UA creation of a hypothesis. Fortunately, physicians and engineers have been concerned with this issue for decades, and a massive amount of literature that summarizes hypothesis generation based on statistical analysis is available. Devore [55] summarized UA descriptions of all standard statistical tests for UA datasets. It includes average, variance, standard deviation, the sum of squares, root sum of squares, pooled variance, linear interpolation, linear regression, sensitivity coefficient, covariance, and correlation.

During the last decades, machine learning approaches have become increasingly important in generating hypotheses and are a standard tool by now. In this context, clustering approaches are a popular form of machine learning. A survey on UA clustering approaches was presented by Aggarwal and Reddy [1]. These algorithms can transform uncertainty throughout their computational model and provide a UA hypothesis forming. Neural networks are increasingly popular in providing hypotheses as well. Here Gal provided a state-of-the-art analysis of UA approaches [73]. Deep learning approaches that utilize Bayesian theory [244] to output a UA hypothesis are most popular in this context.

Uncertainty Quantification $\overline{Q_H}$ ■

As shown in Section 3.2, the hypothesis component can introduce uncertainty through parameter uncertainty, incompleteness, and approximation of models. For input parameter uncertainty, sensitivity analysis can be used [216]. For a high number of input parameters, dimension reduction approaches [219] can be used for uncertainty quantification. A model's incompleteness and approximation approach can be described using model reliability approaches. A summary, including an evaluation of these approaches, can be found in [198].

The quantified uncertainties need to be combined with the uncertainty quantification that is attached to the input dataset or the visualization using an uncertainty aggregation approach.

User Interaction $\overline{U_H}$ ■

User interaction with a hypothesis usually concerns operations such as selecting a proper hypothesis generation algorithm or adapting previous parameter choices. The interaction should select UA hypothesis-forming operations in this context. Otherwise, a guess of certainty about the hypothesis result has to be taken. In addition, a user may be enabled to adapt input parameters required for computing UA hypothesis forming. Here, users need to be able to set the input parameter J. In addition, the input parameter needs to be expressed with an uncertainty quantification $u(J)$ as well. Thus, the user should be enabled to manipulate this uncertainty quantification. The resulting UA input parameter $\overline{J} = (J, u(J))$ needs to be considered in the UA computation. Here, sensitivity analysis can be utilized to quantify this uncertainty.

Hands-On Example: In the hands-on example, the goal is to understand surgery paths that are planned to remove a brain tumor. As the choice of these surgery paths massively affects the patient's health, depending on the areas that will be intersected, the goal is to identify different areas in the patient's brain. This results in a clear segmentation task for the given MRI dataset. Therefore, a fuzzy segmentation approach is selected that assigns a probability to each pixel to be contained in a specific area. The chosen approach is highly interactive, guiding the user through the segmentation process. Here, users can review their segmentation results and adjust the settings of the segmentation algorithm.

Based on the patient's brain segmentation, a probing is applied that samples different surgery paths according to their intersection with the identified brain regions. To create a hypothesis about what surgery paths are more suitable than others, a query procedure allows sorting the tested surgery paths based on user-defined criteria, such as excluding specific areas.

3.3.5 Uncertainty-aware Visualization $\overline{V}$

Visualization is a key component in the VA process. It allows users to gain valuable insight into the dataset and provide a natural understanding of the underlying uncertainty [119]. In the UAVA process, a UA visualization is defined as $\overline{V} = (V, u(V))$, where $u(V)$ describes an arbitrary type of uncertainty.

Generation from Uncertainty-aware Datasets $\overline{V_S}$ ■

The generation of a UA hypothesis $\overline{V}$ can be described by a function starting from two sources: a UA dataset ($\overline{V_S} : \overline{S} \rightarrow \overline{V}$) and a UA visualization ($\overline{V_H} : \overline{H} \rightarrow \overline{V}$). The latter is part of the hypothesis-visualization-subcycle of VA that will be discussed in Section 3.3.6.

UA visualization is a very active field that has been researched for decades, resulting in various visualization approaches. Still, it only represents one component of the VA process. Therefore, visualization can be seen as one computational step in the pipeline.

In general, the utilized visual variables that are considered to express uncertainty in visualization can be listed as follows: comparison techniques, attribute modification, glyphs, and image discontinuity [191]. The choice of uncertainty visualization and the visual variable expressing the uncertainty depends on the underlying dataset and the use case for which the VA process is designed.

Uncertainty Quantification $\overline{Q_V}$ ■

The visualization process itself introduces uncertainty into the VA process, namely mapping, perceptual, memory, and thinking uncertainty, as shown in Section 3.2. Dasgupta and Kosara [50] summarized the need for quality metrics in visualization that can quantify uncertainty, such as mapping uncertainty. Diamond [56] surveyed perceptual uncertainty and how it can be expressed. Coutinho et al. [47] described the role of memory and thinking uncertainty when reviewing a visualization. They propose that describing these uncertainties is hard to achieve as human cognition is very complex, and parts of its functionality are still unknown.

User Interaction $\overline{U_V}$ ■

User interaction with visualizations can be quite manifold. Brodbeck et al. [26] summarized available interaction techniques. Regarding interaction with uncertainty visualization. Sacha et al. [203] proposed a suitable user interaction with UA visualization approaches as a fundamental requirement to provide a suitable UAVA process. Still, a summary of all necessary interaction metaphors is not available. In this context, we would like to suggest the following considerations when designing UA interactions for visualization.

First, there needs to be a specific selection or zooming operation based on the data uncertainty but not the data itself. Second, the result of the current interaction methodology needs to provide information about the uncertainty currently shown and the overall uncertainty captured in the dataset.

Hands-On Example: The hypothesis generation in the presented case is highly connected to visualization. First, different brain regions are segmented via an interactive visualization. Users can define and review brain regions individually to control the result of the fuzzy segmentation approach. Second, the selection of surgery paths is assisted with visualization as well. Here, the surgery paths are mapped and color-coded to provide an understanding of which brain areas will be affected by the surgery paths. This also includes the visualization of the underlying fuzzy segmentation result. The visualization also provides an interactive interface to control the surgery path selection.

3.3.6 The Interplay of Hypothesis $\overline{H}$ and Visualization $\overline{V}$

Hypothesis and visualization form a subcycle in the VA process defined by Keim et al. This connection forms the core of the VA process and needs to be preserved in the constructed UAVA process. This process can be run arbitrarily often, which requires specific handling for the uncertainty aggregated along with these runs.

Propagation can technically be applied in the hypothesis-visualization-subcycle. Still, the question about the knowledge that can be extracted from UA hypothesis and visualization arises if the amount of captured uncertainty constantly increases. Here, we suggest setting a user-selected threshold that allows indicating data points containing a higher uncertainty quantification as this threshold. As a result, the user would interact with the UA hypothesis and visualization and increase the uncertainty attached to this process in each step. When the user-defined threshold is exceeded for specific data points, they will be highlighted, and the user can adapt the selections.

The interplay of a UA hypothesis $\overline{H}$ and a UA visualization $\overline{V}$ happens in both directions and will be explained in the following.

Visualization Generation from Hypothesis $\overline{V_H}$ ■

The process of generating a UA visualization based on a UA hypothesis can be described as $\overline{V_H} : \overline{H} \rightarrow \overline{V}$. Here, we assume that a UA statistical analysis has been

conducted, requiring a proper visualization. Depending on the algorithm's output, a UA dataset can be created. The specific data type depends on the underlying algorithm and requires a sophisticated visualization approach. Here, the same rules apply as in Section 3.3.5.

Hypothesis Generation from Visualization $\overline{H_V}$ ■

Building a UA hypothesis from UA visualizations is defined as the function $\overline{H_V} : \overline{V} \rightarrow \overline{H}$. Unfortunately, this process cannot be determined analytically in its entirety, as it involves the subjective impression of a user to refine a hypothesis regarding the available visualization. What can be determined is the user input that leads to a hypothesis. Here, we suggest letting the user quantify how certain his selections are to express the uncertainty of the hypothesis generation at least partially.

In these considerations, user bias is an important aspect to consider. Szafir [224] provided five suggestions to deal with this bias:

- Use diverging colors instead of rainbow color tables
- Avoid animated visualization but encode movement statically
- Instead of truncating axes, show relative between values
- Avoid 3D visualization if possible
- Provide a transparent description of the visualization process

Hands-On Example: In the presented example, the interplay between visualization and hypothesis is a crucial component. Here, UA segmentation results (hypothesis) are visualized. In addition, the computation of intersections of surgery tunnels and different areas in the brain are computed and converted into a visualization. On the other hand, the visualization is used as an indicator to capture if parameters need to be adjusted. In this example, this can be adaptations of the segmentation approach input or the selection of desired areas in the human brain. Here, the subcycle is clearly visible, as this process is often repeated multiple times before coming to a conclusion.

3.3.7 Uncertainty-aware Insight $\overline{I}$

The term insight I can be defined as knowledge that is gained during analysis and has to be internalized, synthesized, and related to prior knowledge [203]. In terms of uncertainty, a UA insight $\overline{I} = (I, u(I))$ is composed of the insight generated from the UAVA process and quantification of the credibility of the derived result from $u(I)$. Here, $u(I)$ describes subjective uncertainty. In reality, insight cannot be defined mathematically in many cases, as it is a subjective impression of the user, often affected by personal bias that runs the VA process. Based on this problem, describing the respective uncertainty quantification might not be possible.

Conclude Visualization $\overline{U_{CV}}$ ■

Throughout the interaction of the user with UA visualization, insights are concluded by the user. These insights can usually not be described mathematically as they depend on a subjective user experience. Here, visualization evaluation approaches come into play, as they offer metrics and approaches to quantify the amount of insight generated by a visualization.

Regarding uncertainty visualization, Hullmann et al. [119] presented a state-of-the-art report that summarizes uncertainty visualization evaluation approaches. These approaches can be used to approximate at least the insight generated by a UA visualization approach. Also, users might want to express their uncertainty about their decisions in the form of text or on a given scale to create a rating.

Conclude Hypothesis $\overline{U_{CH}}$ ■

UA insight generated from a UA visualization can luckily be quantified mathematically (to the point where analysis results are interpreted). Here, the UA hypothesis directly implies the uncertainty of the derived UA insight. In fact, they are identical, which means $u(I) = u(H)$.

Uncertainty Quantification $\overline{Q_I}$ ▲

Insight generated in the VA process can be affected by uncertainty due to decision-making bias or experience and knowledge that may keep a user from accepting novel findings. Lewandowsky et al. [152] stated that uncertainty always affects knowledge. Unfortunately, insights are largely subjective, so uncertainty quantification is hard

to achieve. Most considerations are philosophical rather than computational. For evaluation purposes, benchmark tasks have shown to be useful for identifying and assessing analytic findings. Still, they are not sufficient in most cases [189]. Here, a clear strategy of uncertainty quantification is missing.

3.3.8 Feedback Loop $F(S)$ ■ and Uncertainty-aware Feedback Loop $\overline{F(S)}$ ▲

The classic VA approach indicates that VA is designed to be a process $F(S)$, where generated knowledge can act as further data input. As already shown, generated insight from the UAVA process can be of two types: insight with uncertainty quantification and insight without uncertainty quantification. These types of insight need to be treated differently. Insights without uncertainty quantification that need to be reinserted into the VA process are fed back into the dataset component. This motivates the existence of the standard dataset component in the UAVA process.

Starting from here, a suitable uncertainty quantification needs to be found according to the data structure of the insight. As the uncertainty of insight cannot be computed directly in many cases, insight can be modeled as a normal dataset and then be transferred into a UA dataset through a suitable uncertainty quantification. Here, the taxonomy of uncertainties in VA by Gillmann et al. [81] provides a great starting point.

On the other hand, insights with uncertainty quantification need to be inserted in the UA dataset component, as no additional quantification is required.

Hands-On Example: In the presented example, UA insights can be generated in various ways. Here, users can use the hypothesis to understand different regions in the patient's brain and obtain an impression of how safe the selection of a specific surgery path is. This results in the creation of the mentioned feedback loop. First, the segmentation and surgery path analysis result in novel insight into the structure and composition of the human brain and the uncertainty inherent in this computation. This knowledge can be directly fed back into the VA process. On the other hand, subjective considerations on trust in the visualization and computation techniques either have to be quantified by the user or are not quantified in terms of uncertainty and need to be fed back into the original data component.

3.3.9 Provenance Generation ▲

When running a UAVA process, uncertainty will be propagated and aggregated along with the operations performed during the VA process. Varga et al. [235] described the importance of provenance analysis and visualization. This implies the tracking of processing steps and uncertainty throughout each computational step of the VA process, referred to as provenance. Therefore, each time a UA dataset, hypothesis, or visualization is created, the current uncertainty quantification and the respective operation must be stored and are subject to further analysis.

We encourage providing a visualization and interaction tool to let users follow uncertainty development throughout the VA process. This can give users important hints on which operations caused a drastic increase in uncertainty or at which point the aggregated uncertainty exceeds a threshold that is known to be the highest amount of uncertainty that still allows for interpretation. Also, collecting all steps in the analysis can leave the users uninterrupted while solving a task, still allowing them to analyze single steps further afterward [203]. Herschel et al. [114] provided a survey on provenance creation. Still, selecting a suitable tool is highly dependent on the underlying dataset and the application scenario. Ragan et al. [193] characterized provenance in visualization and data analysis approaches. We will use this work as a basis for the following descriptions of uncertainty provenance that are required in the VA process.

Provenance Generation for Uncertainty-aware Datasets $\overline{P_S}$ ▲ and Uncertainty-aware Hypothesis $\overline{P_H}$ ▲

The provenance of data focuses on the history of changes and movement of data. Data provenance is often heavily emphasized in computational simulations and scientific visualization, in which significant data processing is conducted. The history of data changes can include subsetting, data merging, formatting, transformations, or execution of a simulation to ingest or generate new data [193]. This can be directly transferred to the uncertainty of a UA dataset and the uncertainty of a hypothesis.

Provenance Generation for Uncertainty-aware Visualization $\overline{P_V}$ ▲

As Ragan et al. [193] stated, visualization provenance concerns the history of graphical views and visualization states. This process is tightly coupled with data

transformation and the interactions that produce the visualization. These concepts must be adapted to provide a provenance generation for the uncertainty in UA visualization. A survey on available methods in provenance visualization and user interaction was conducted by Xu et al. [250].

Provenance Generation for Uncertainty-aware Insight $\overline{P_I}$▲

The provenance of UA insights needs to include the uncertainty of the user. Unlike data computations, insights are not directly observable in all cases, so their uncertainty is not observable directly, as shown in Section 3.3.7. Here, solely quantifiable insights can be included in the provenance generation of uncertainty.

Hands-On Example: In the described example, there exist a variety of computational steps as well as a variety of potential user interactions. To implement the principle of provenance, all computational steps, their intermediate computational results, and the attached uncertainty to each of these results should be stored and visually communicated in a story graph. This allows users to understand each computational step in the UAVA process.

3.4 Guideline on Application Creation

This section provides a guideline to transform a classic VA application, modeled using the process of Keim et al. [136], into a UAVA application. It defines five steps, providing a transformation to implement uncertainty systematically. The guideline starts with the definition of a VA process and continues with the quantification of uncertainty in the system components (dataset, visualization, hypothesis) and the functions of the new UA process components. Finally, the feedback function is adapted, and provenance generation is implemented.

We propose five steps to create a UAVA process, starting with a classic VA process. Various works exist that assist VA process design, and the resulting UAVA process can be constructed using most of the existing components and functions in the classic VA process.

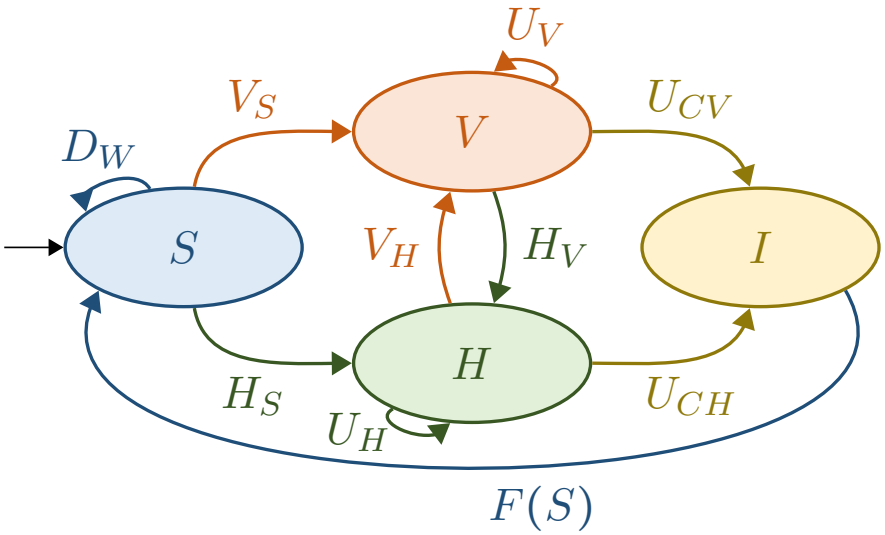


Fig. 3.3: Step 1: Design a VA process. All components of the classic VA process need to be created.

3.4.1 Step 1: Design a Visual Analytics Process

In the first step, a classic VA application (Figure 3.3) is created that may be skipped when such an application is already present. It is highly dependent on the task, user, and dataset, as these categories imply the selection of proper hypothesis, visualization, and interaction techniques.

Keim and Zhang [135] show the general description of the classic VA process. This step selects all components and functions in the classic VA process necessary for the task. This may not include all functions that are present in the general description of the VA process.

In this stage, it is important to note that required data transformation (D_W) operations and the feedback loop ($F(S)$) are identified. Conlen et al. [42] provided a first attempt to provide design rules for VA approaches. These have been further refined and sorted into a workflow by Wang et al. [246]. We will use these stages as substeps in the presented guideline to generate a VA application.

Step 1.1 Content Gathering and Aggregation First, the content and information necessary for the task are collected by checking existing solutions and workflows. Users identify, seek, and extract information sources relevant to analytical tasks in their domain, meaning that they identify all data relevant to the task to create the dataset component S.

Step 1.2 Content Filtering and Customization In this step, users use filtering to familiarize themselves with the collected content. They also personalize the analysis environment in which this content is filtered to fit their workflow. Here, the data pre-processing steps D_W are defined. This step is important, as data is often overwhelming, or not all aspects are necessary in an analysis scenario.

Step 1.3 Content Organization and Information Analysis Users organize and examine the collected content from multiple perspectives to look for data patterns and desired information. Multiple levels of granularity can be found where different visualization techniques are needed. This refers to creating the visualization component in the VA process, where numerous visualization and interaction approaches must be examined, creating the functions V_S and U_V.

Step 1.4 Evidence Collection and Hypothesis Generation Possible automated tests and testable hypotheses are found next, collecting supporting evidence in the process. Sources of evidence for supporting or rejecting a hypothesis are identified, defining clear statements about the outcomes of a test. This is related to creating the hypothesis component in the VA process. Proper analysis methodologies are established, resulting in the functions H_S and U_H.

Step 1.5 Report Generation and Status Update Functions between visual exploration and hypothesis testing are established. visualization of analysis status and hypothesis tests via visual exploration is created by providing notifications and process updates, resulting in the functions V_H and H_V.

Step 1.6 Post-Analysis and Summarization Here, users focus on analysis outcomes of the implemented workflows, defining possible insight classes. Those insights are returned to the existing dataset component, resulting in the feedback loop $F(S)$.

At this point, uncertainty is not included in the design process explicitly. We acknowledge that the choices made in this step greatly impact the further design of a UAVA application. Still, this step comes with multiple benefits. First, it allows the systematic creation of the components and functions in the visual analytics process. Second, these components are also part of the UAVA process and help in the further design process as all potential sources of uncertainty are already defined. New applications should be planned using this guideline instead of implementing each step directly to avoid unnecessary changes and reduce implementation efforts.

3.4.2 Step 2: Quantify System Uncertainty of the Given Visual Analytics Process

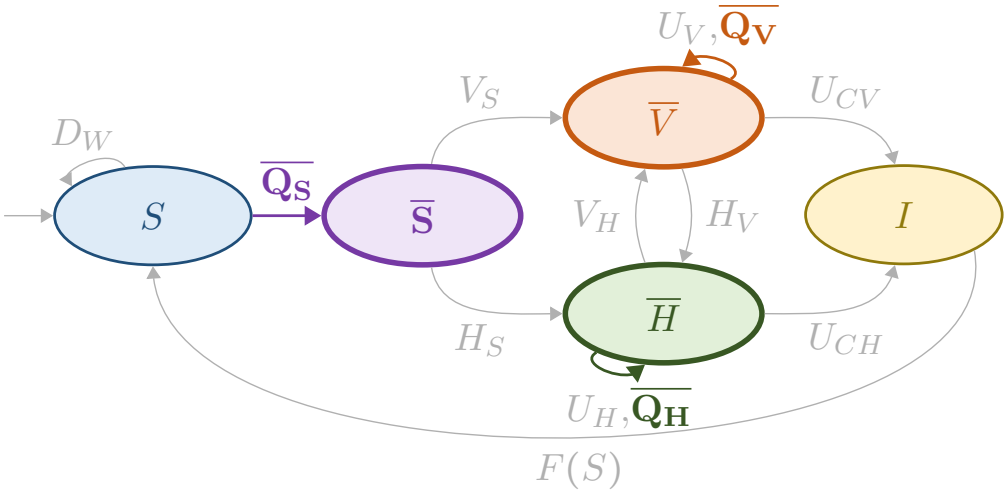


Fig. 3.4: Step 2: Quantify system uncertainty of the given VA process. This inserts the functions $\overline{Q_S}$, $\overline{Q_H}$ and $\overline{Q_V}$.

As shown in the taxonomy of uncertainties in VA by Gillmann et al. [81], sources of uncertainty in VA can be separated into system and human uncertainties. Here, the focus is on the system uncertainties. Even though this guideline explicitly uses this specific taxonomy, other sources for extracting the right quantification methods can be used.

System component uncertainty quantification is performed based on the classic VA design in Step 1. Here, the taxonomy of uncertainties [81] can be considered to extract the right methodologies. It provides a variety of information for each uncertainty event, namely, uncertainty type, level of quantifiability, and description.

For each source of uncertainty, the following steps need to be accomplished.

Step 2.1 Decide if an Uncertainty Source Affects the Considered VA Process Notably, not all sources of uncertainty may be inherent in every VA scenario. Further, there exist cases where sources of uncertainty may be present and known but are deemed irrelevant. This decision is made by the users, depending on the given task.

Step 2.2 Select Types and Levels of Relevant Uncertainty Events According to the taxonomy of uncertainty events [81], the selected uncertainty events' type and level can be collected.

Step 2.3 Determine the Uncertainty Description The uncertainty description of an event holds the following information: dimension, type, and continuity. This information depends on the data, hypothesis, and visualization the uncertainty event is attached to. Therefore, the description of the selected uncertainty events can be derived.

Step 2.4 Select a Proper Quantification Strategy for the Selected Uncertainty Events In the taxonomy of uncertainty events [81], each uncertainty event comes with one or more potential groups of uncertainty strategies. Together with the domain experts, available techniques are discussed and selected.

This creates three novel functions in the UAVA process: $\overline{Q_S}$, $\overline{Q_H}$ and $\overline{Q_V}$, as shown in Figure 3.4. Importantly, quantifying uncertainty events in the component S (dataset) leads to a novel $\overline{S}$ component. In contrast to the other components, this separation is particularly important as not all uncertainty events are quantifiable with current knowledge. When applying a feedback loop, the incompleteness of quantifiability needs to be taken into account. This issue will be further investigated in Step 4.

3.4.3 Step 3: Integrate and Connect Uncertainty-aware Solutions

This step combines the quantified uncertainty events and the selected methodologies. First, the selected methodologies from Step 1 are exchanged to achieve uncertainty-aware descriptions. For example, if a clustering approach is selected as the hypothesis, the approach needs to be exchanged with a UA clustering approach. This builds the UA components of the VA process, as shown in Figure 3.5.

To implement the functions between the components in the VA process, uncertainty quantification approaches and the chosen UA methodologies need to be connected. Two operations achieve this: propagation of uncertainty and aggregation of uncertainty. Propagation is requested to connect the components in the VA process, whereas aggregation is required when multiple sources of uncertainty need to be summarized.

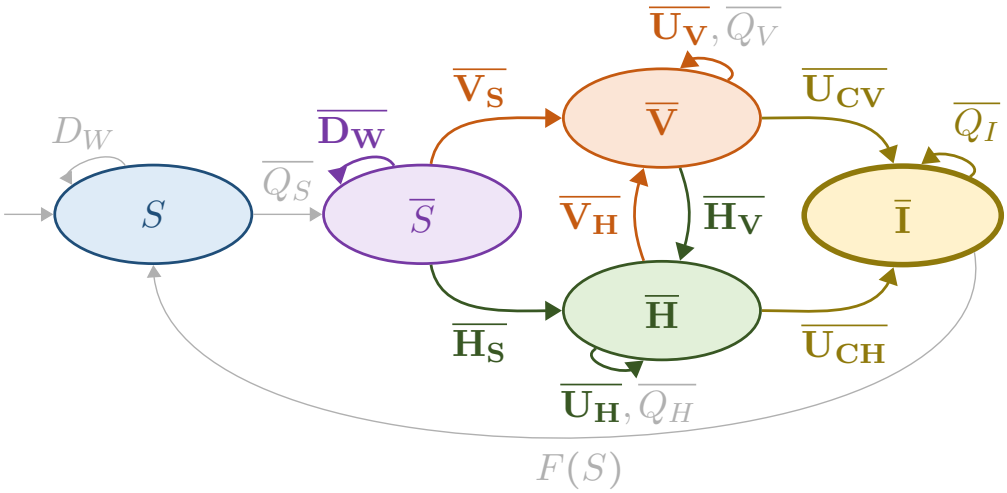


Fig. 3.5: Step 3: Integrate and connect uncertainty-aware solutions. This requires a revision of the components $\overline{V}$, $\overline{H}$, and $\overline{I}$ and all functions originating or ending in these components.

Please note that the components and functions in the VA process that have been revised to be UA are marked by a bar above their description (see Figure 3.5). This results in the following three substeps.

Step 3.1 Exchange Hypothesis H by an Uncertainty-aware Hypothesis $\overline{H}$ When selecting algorithms for exchange with uncertainty-aware variants, sensible decisions are made towards their improved effect on the workflow. By now, users can select from massive amounts of uncertainty-aware analysis tools. The first choice is recommended to be algorithms that are close to the originally selected algorithm while including uncertainty.

Step 3.2 Exchange Visualization V by an Uncertainty-aware Visualization $\overline{V}$ This step works simultaneously with Step 3.1. The state-of-the-art approaches shown in Section 3.1.2 can be considered to find proper uncertainty-aware visualization techniques.

Step 3.3 Aggregation of Uncertainty in $\overline{H}$ and $\overline{V}$ Uncertainty can be introduced into the VA process in all components, where each component can be affected by multiple sources of uncertainty.

Technically, arbitrary aggregation functions can be used to achieve uncertainty aggregation, where Cai et al. [30] provided a survey of these functions. In the VA

process, aggregation functions need to be selected so that all identified sources of uncertainty of one component can be aggregated properly. In addition, we suggest letting users select the importance of available uncertainty sources. This allows for creating awareness about the effect of different sources of uncertainty while rating them.

Step 3.4 Propagation of Uncertainty between $\overline{V}$ and $\overline{H}$ When considering a transition from one component of the VA process into another, data needs to be transferred in a specific manner. This is highly dependent on the transformation step in the VA process.

Mathematical operations propagate data throughout the VA process (at least in the system component). These operations affect the data as well as their uncertainty. In addition, the inclusion of uncertainty also affects the operator, as they usually aim to map models known to be affected by uncertainty. As a result, operators need to be adjusted to handle data and their uncertainty. Multiple approaches, usually built on the principle of error propagation [75], exist to tackle this problem.

Step 3.5 Exchange Insight I by an Uncertainty-aware Insight $\overline{I}$ The insights created by visual exploration or hypothesis testing are propagated into the insight component I together with attached uncertainty information. This uncertainty information can be directly propagated towards the insights, making the insight component UA $\overline{I}$. Also, the user could try to quantify the uncertainty about the extracted insight $\overline{Q_I}$, i.e., by annotating the data or specifying his confidence as a number.

3.4.4 Step 4: Separate the Visual Analytics process Feedback Loop into Quantifiable and Unquantifiable Feedback

The classical VA process has a feedback loop $F(S)$ in which new insights are fed back into the dataset component. This is the main advantage of the classic VA process, which aims to preserve and enhance the presented approach. Unfortunately, we showed that insight is hard to quantify, even partially, under the aspect of uncertainty. Here, the separation of new insights into two feedback loops is obtained to distinguish between both types, the classic feedback loop $F(S)$ and the UA feedback loop $\overline{F(S)}$, as shown in Figure 3.6. The UA feedback loop inserts the data into the UA dataset component $\overline{S}$. In contrast, the feedback loop that holds information that cannot be quantified is inserted into the dataset S component. As an example, this

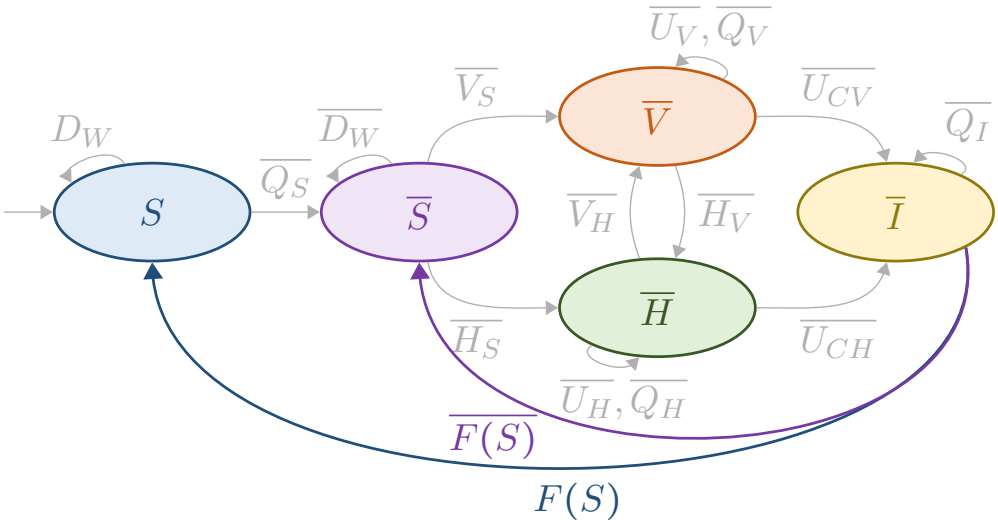


Fig. 3.6: Step 4: Separate the VA process feedback loop ($F(S)$) into quantifiable feedback and feedback that cannot be quantified $\overline{F(S)}$.

can be insights that arise from the process of exploring the shown dataset. Users might identify patterns in datasets that cannot be described mathematically. Here, the user must be incorporated to determine a proper uncertainty quantification. In this process, the identification of meaningful uncertainty quantification ($\overline{Q_S}$) is extremely important.

Step 4.1 Define Insight with Uncertainty Quantification $\overline{S}$ Insights with uncertainty quantification need to be inserted in the $\overline{S}$ component due to the missing need for uncertainty quantification. This insight usually results from the hypothesis component, which computes knowledge based on uncertainty-aware data and the uncertainty-aware hypothesis.

Step 4.2 Define Insight Without an Uncertainty Quantification S Insights without uncertainty quantification that need to be reinserted in the VA process must be fed back into the dataset component S. This justifies the existence of the original dataset component S in the UAVA process, as not all gathered information can be quantified according to uncertainty. Starting from here, a suitable uncertainty quantification needs to be found according to the data structure of the insight. As the uncertainty of insight cannot be computed directly in many cases, insight can be modeled as a normal dataset and then be transferred into a UA dataset through a suitable uncertainty quantification.

3.4.5 Step 5: Generate Provenance of Uncertainty-aware Components in the Visual Analytics process

When running a UAVA process, uncertainty will be propagated and aggregated along with the performed operations of the VA process according to the rules defined in Step 3. Varga et al. [235] highlighted the importance of provenance analysis and provided potential visualization approaches. They imply that the uncertainty information and process steps should be tracked throughout the entire VA process. This is called provenance, a novel component in the VA process that assists in implementing provenance principles (P), as shown in Figure 3.7.

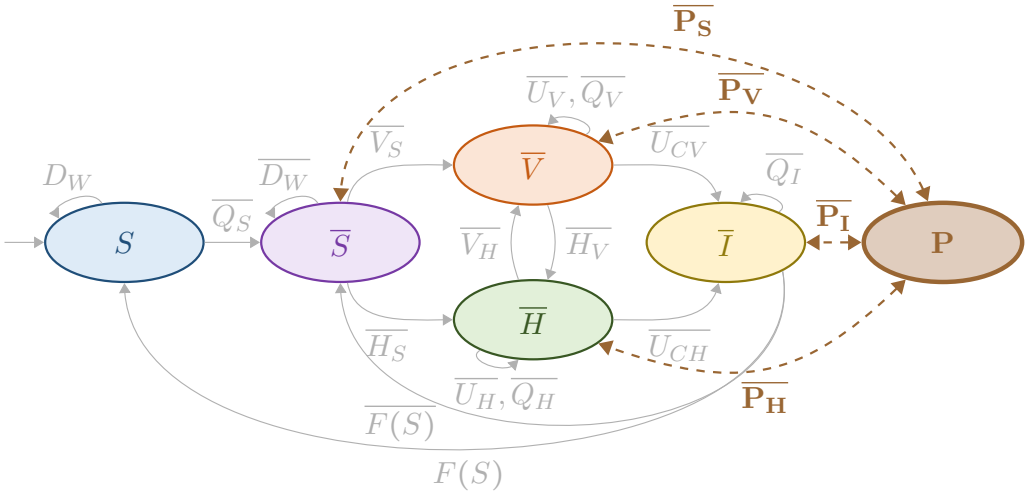


Fig. 3.7: Step 5: Generate Provenance of uncertainty-aware components in the VA process. The provenance of the system components ($\overline{P_S}$, $\overline{P_V}$, and $\overline{P_H}$), as well as the provenance of the human components are inserted ($\overline{P_I}$).

As shown in the taxonomy of sources of uncertainty in VA [81], the VA process can be separated into uncertainty sources of the system and human components. Provenance generation needs to be tackled differently in both cases.

Step 5.1 Provide Provenance of Sources of Uncertainty in the System Component
On creation of a UA dataset $\overline{S}$, a UA hypothesis $\overline{H}$ or a UA visualization $\overline{V}$, the quantification of uncertainty, the operations utilized, and their results must be stored and made accessible.

We suggest providing visualization and interaction tools to let users follow uncertainty development throughout the VA process. This development aims to provide users with a tool to measure which operators cause a high increase in uncertainty. Additionally, thresholds that indicate if the uncertainty exceeds a maximum, making the

underlying data noninterpretable, should be available. While Herschel et al. [114] provided a survey on provenance creation, selecting a suitable tool or visualization approach highly depends on the underlying use case and the data type. In addition, Ragan et al. [193] characterized the provenance in visualization and data analysis approaches, which form the basis of the following discussion.

The provenance of a dataset S focuses on the history of changes and movement of data. This type of provenance is very important in computational simulations and scientific visualization, as data is massively processed. The history of data and the respective uncertainty changes can include subsetting, data merging, formatting, transformations, or the execution of a simulation to ingest or generate new data [193]. These principles can be applied directly to $\overline{S}$ and $\overline{H}$ while creating two novel functions $\overline{P_S}$ and $\overline{P_H}$.

To achieve visualization and interaction provenance, Xu et al. [250] provided a survey summarizing existing technologies. We suggest checking for available techniques and carefully integrating them into the generated VA process, creating a novel function $\overline{P_V}$.

Step 5.2 Provenance of Sources of Uncertainty in the Human Component The provenance of UA insights must also include uncertainty. Unlike data computations, insights are not directly observable as they are subjective in many cases, so their uncertainty is not observable directly as well [193].

This means that the provenance generation of the uncertainty sources contained in the insight component can only be partially accomplished. Mechanisms exist that can be applied to tackle this problem.

As Ragan et al. [193] suggested, insight can be tracked by annotations. We suggest applying the same principle to the sources of uncertainty in the human component of the VA process. Here, two steps must be accomplished:

1. Describe sources of uncertainty in the human component

2. Provide an annotation tool to track occurring sources of uncertainty in the human component

3.4.6 Summary

Based on the discussion above, the resulting guideline for UAVA application creation assembles as follows:

1. Design a visual analytics process
2. Quantify system uncertainty of the given visual analytics process
3. Integrate and connect uncertainty-aware solutions
4. Separate the visual analytics process feedback loop into quantifiable and unquantifiable feedback
5. Generate provenance of uncertainty-aware components in the visual analytics process

3.5 Example Application

This section exhibits a post hoc mapping of an explainable artificial intelligence application for brain lesion prediction via the presented guideline. The application allows users to understand the AI models that predict lesion locations, where lesions resemble areas of tissue that suffered damage.

In the presented example of Gillmann et al. [82], a neural network is used to predict the lesion in a patient's brain that can result from stroke, as shown in Figure 3.8. The authors trained several networks, all being able to predict the lesion to a certain accuracy, but the medical collaborators wished to understand how the network made its decision [249].

In the *first step*, we start with the content gathering (Step 1.1). Here, data related to the stroke imaging process can be considered. In the given case, the clinic used computed tomography scans enriched with contrast bolus to highlight the brain's vascular system and perfused tissue (Step 1.2). The visual system was developed to compare the ground truth used for training to the neural network's prediction, as shown in Figure 3.8 (Step 1.3). As usual in such scenarios, the predicted lesion is thresholded by a value of 0.5, creating our hypothesis (Step 1.4). Further, the display represents the histogram of the currently reviewed prediction and the ROC (receiver operator curve) of the prediction. This information can be fed back into the visual analytics process (Step 1.5).

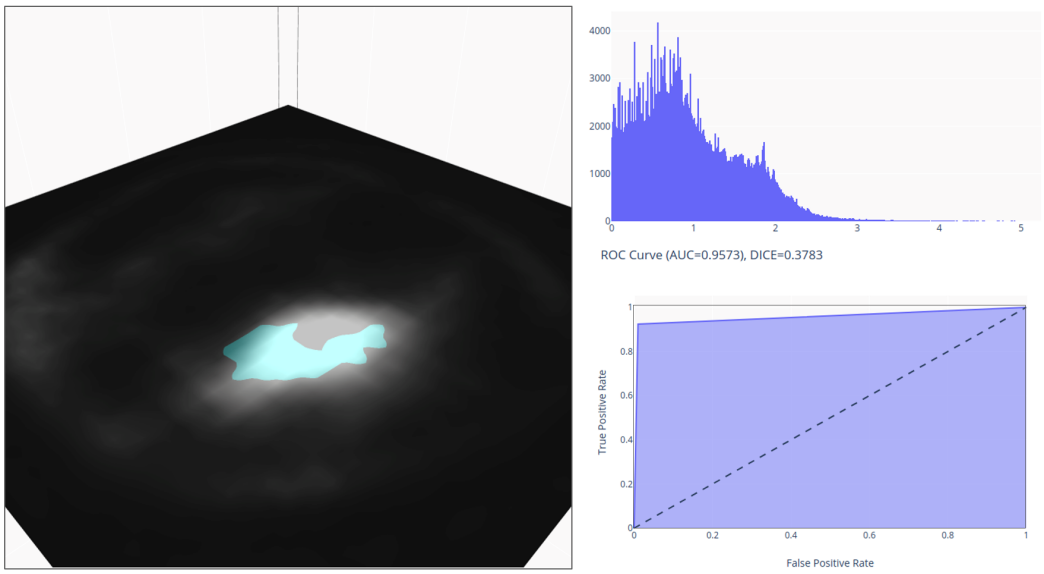


Fig. 3.8: VA for analyzing the neural network to predict a brain lesion by Gillmann et al. [82]. Left: slice view to compare ground truth and prediction. Upper right: histogram of the prediction. Lower right: ROC of made prediction.

In the *second step*, all relevant sources of uncertainty must be quantified. In this specific case, there is an inherent uncertainty in the input dataset's ground truth, as three ground truths exist per patient. This results in a level 2 uncertainty event with a dimensionality of 1 [81]. Additionally, the machine learning model introduces uncertainty into the VA process, as it is potentially incomplete. This can be expressed by a level 3 uncertainty event using sensitivity analysis (Step 2.1-2.4)

In the *third step*, the components of the VA process are exchanged. At first, the U-Net, which was used to predict the patient's lesion, is exchanged by a Bayesian U-Net (Step 3.1). Further, the evaluation of the U-Net was adapted by an uncertainty-aware computation, as shown in Figure 3.9 by Sperling et al. [221] (Step 3.2). Here, the hard threshold that decides which pixels are considered true and false positives, as well as true and false negatives, is exchanged by a PDF. Adjusting σ of this threshold leads to areas in the classification that cannot be clearly sorted into the four classes.

The feedback loop must be separated in the *fourth step*. In this case, computations such as accuracy or other metrics, considering the quality of the prediction, can be computed based on the uncertainty-aware classification of pixels from the prediction (Step 4.1). As this is the only information fed back, no insight without uncertainty quantification is being processed (Step 4.2)

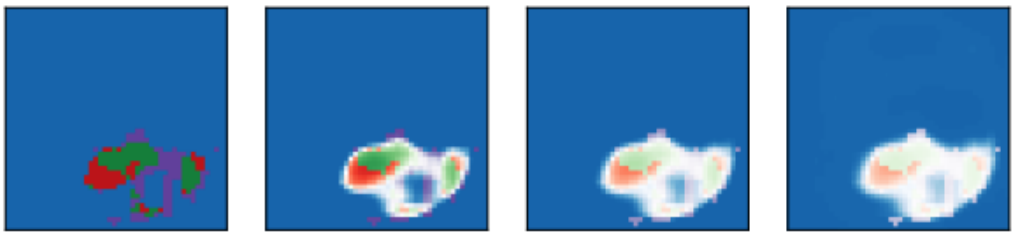


Fig. 3.9: UAVA evaluation of neural networks by Sperling et al. [221]. TP (green), FP (red), TN (blue), and FN (purple) are shown together. From left to right by manipulating the σ of the used uncertainty-aware threshold, separating the four classes becomes fuzzier (white areas).

In the *fifth step*, the implementation of provenance is implemented. This directly relates to the manipulation of σ, as users can directly relate the consequences of the used σ value.

3.6 Conclusion

This chapter forms the thesis core, providing the technical background for the applications in the following chapters. It extends the VA process of Keim et al. [136] into a UA version that considers uncertainty in every component. This is achieved by quantifying uncertainty in the system and human components, as well as propagating and aggregating uncertainty throughout the process. It forms a holistic mechanism to tackle uncertainty originating from data, models, and humans in VA approaches.

A guideline that transforms a classic VA process into a UAVA process is provided, allowing the creation of compatible applications. Here, five steps need to be accomplished. First, a VA process is designed. Second, sources of uncertainty must be detected and, if possible, quantified. Third, components in the VA process need to be exchanged for UA alternatives. Fourth, the feedback loop needs to be separated, and last, provenance mechanisms need to be implemented. This results in a holistic mechanism to tackle uncertainty originating from data, models, and humans in VA approaches. For each step, we make suggestions and provide an example application to ease the understanding of the five-step guideline.

4 Uncertainty-aware Visual Analytics Application for Proteins

Due to the limitations of existing experimental methods for capturing stereochemical molecular data and atomic movement, there is an inherent level of uncertainty present in models describing the conformation of macromolecules. This uncertainty can originate from various sources and can significantly affect algorithms and decisions based on such models. Incorporating uncertainty in state-of-the-art visualization approaches for molecular data is an important issue to ensure that scientists analyzing the data are aware of the inherent uncertainty present in the representation of the molecular data. In this chapter, we introduce an application that allows biochemists to explore molecular data in a familiar environment while including uncertainty information within the visualizations. Our application is based on an anisotropic description of proteins that can be propagated along with required computations, providing multiple views that extend prominent visualization approaches to visually encode the uncertainty of the atom positions, allowing interactive exploration. We show the effectiveness of our approach by applying it to multiple real-world datasets and gathering user feedback.

4.1 State of the Art and Requirements

The important role of visualization in molecular biology has been outlined by Olson et al. [179] and is applied to many tasks, such as folding prediction, detection of active sites, or structural examination. Several approaches have been proposed in this area in the last decades [143]. They are available in various open-source tools,

This chapter has been partially published in the following articles:

- R. G. C. Maack, C. Gillmann, and H. Hagen. "Uncertainty-Aware Ramachandran Plots". In: IEEE Pacific Visualization Symposium (PacificVis). IEEE, 2019.
- R. G. C. Maack, M. L. Raymer, T. Wischgoll, H. Hagen, and C. Gillmann. “A Framework for Uncertainty-Aware Visual Analytics of Proteins”. In: Computers & Graphics, 2021.

making visualization a state-of-the-art tool in biochemical applications. Molecular data can originate from a variety of sources, such as X-ray crystallography, Nuclear Magnetic Resonance Spectroscopy (NMR), Cryogenic electron microscopy (Cryo-EM), or atomic simulations. Due to the experimental nature of these approaches, several types of uncertainty are embedded in the acquired data, which mostly results in positional uncertainty of the captured molecular structures.

This uncertainty affects the decision-making process in visual analytics (VA) tasks and needs to be visually communicated, as shown by Sacha et al. [203]. This especially applies to molecular data as it is often used for drug development. When considering molecular data, the origin of uncertainty, the resulting positional variation, and further requirements from the biochemical domain need to be considered in order to provide uncertainty-aware (UA) visualization approaches, as shown in Section 4.2. As molecular biologists are increasingly finding it necessary to employ a wide range of computational tools in their work, an application that can be intuitively used without the need for special training is an important aspect that needs to be considered when developing novel visualization approaches in this domain.

So far, molecular visualization approaches often lack the ability to communicate uncertainty or, if they are available, they need to be included in existing visualization applications, as shown in Section 4.3.

4.2 The Role of Uncertainty in Molecular Data

4.2.1 Biochemical Basics

Proteins are the molecular machinery of biological systems. They are among the most abundant biological macromolecules and undertake a diversity of roles in living systems. Structurally, proteins consist of polymers (chains) of amino acids. There are 20 different amino acids typically found in human proteins. A single protein can consist of one or more amino acid chains, typically ranging in length from a few hundred to several thousand single entities. We would like to refer to the introduction to proteins and their structure by Buxbaum [28].

Figure 4.1 shows that a protein contains a backbone that forms when amino acids bind together. This backbone is a repetition of N , C^{α}, and C as a chain with C_i^{β} being the R-Group that defines which amino acid the residue resembles. The connections between amino acids allow for three rotational angles along with covalent bonds between connected amino acids. Such angles are called dihedral angles,

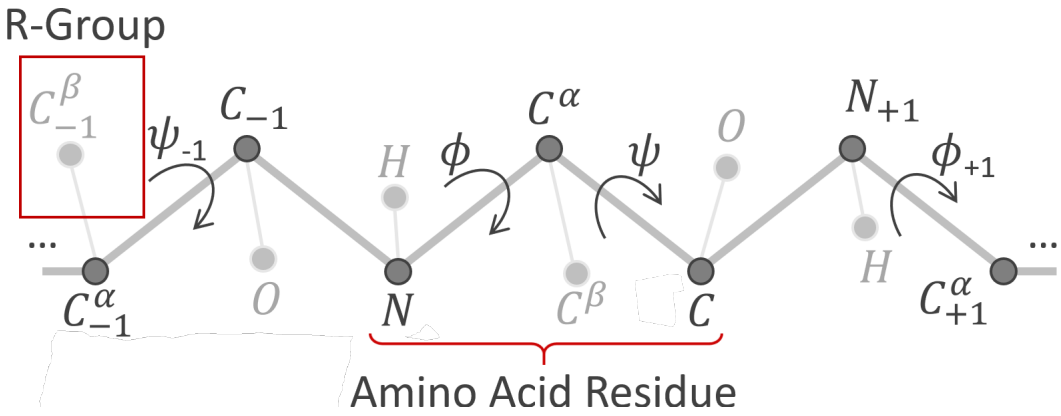


Fig. 4.1: General setup of a protein. The backbone (bold lines) is composed of repeating CCN-Atoms of amino acid residues, where each residue has an R-group attached to it.

meaning the angle between two planes spanned by four neighboring backbone atoms such that the planes intersect along the line between the two middle atoms. The three dihedral angles for each amino acid residue, where the two important ones are typically referred to as ϕ and ψ, allow proteins to adopt a wide variety of three-dimensional structures.

4.2.2 Uncertainty in Biochemical Data

Modern methods for protein structure determination, including X-ray crystallography, cryo-EM, and nuclear magnetic resonance spectroscopy, can provide three-dimensional structures of soluble polypeptides with high confidence. The resolution of these structures is sufficiently high that the location of individual protein and ligand atoms can, in many cases, be determined with precision to within a few Ångströms (100 pm or 0,1 nm). There are several sources of uncertainty related to these atomic positions. Proteins are not monolithic, rigid molecules [39]. Even within the context of protein crystals, local protein regions are subject to thermal mobility to differing extents [133].

Intrinsic disorder and local mobility have been identified as important factors in protein ligand-binding and allosteric functional mechanisms [54, 121, 120]. A variety of computational methods have been developed for the prediction of intrinsic disorder in proteins based on structural and sequence constraints [144, 167, 127], and at least one experimental method for the characterization of intrinsic disorder has been proposed [8].

In X-ray crystallography experiments, structural uncertainty arises from several sources. Thermal-related mobility of backbone and side-chain atoms within the crystal structure leads to a blurring effect on the electron density maps induced by x-ray diffraction. Additionally, the nature of the crystallization process can lead to multiple structural variants of a protein within a single crystal, possibly leading to one or more alternative locations for each protein atom. While resolving a three-dimensional structure, this uncertainty is captured in the *B-Factor* (thermal mobility) and occupancy terms for each atom. In practice, the occupancy is often constrained to a value of 1.0 while the B-Factor term is used to express the overall structural uncertainty of the atom [131, 94].

Both X-ray crystallography and cryo-EM experiments generally include one or more computational modeling steps. For X-ray structures, this step involves the minimization of unexplained observed electron densities. For Cryo-EM experiments, the modeling involves automated or manual groping and alignment of two-dimensional particle images, followed by the calculation of a three-dimensional map from the aligned images. In both cases, artifacts of the modeling process can contribute additional uncertainty to the resulting three-dimensional structure.

For NMR-spectroscopy, structural constraints are captured through the NMR experiment, and a number of structural models consistent with those constraints are generated. In this case, an atom's structural uncertainty is related to the variability of its position across the resulting model set.

It is important to note that not all conformational uncertainty can be captured and visualized. In X-ray crystallography, for example, the observed proteins may adopt a non-native structure to facilitate the formation of a crystal lattice. The difference between the native structure and the crystal structure is not known and is thus not captured in the experimental data.

As experimental methods for observing the structure of biological macromolecules have continued to advance, many sources of uncertainty in the resulting molecular models remain. A clear understanding of variability and uncertainty in protein structure is important to biochemists in a wide variety of scientific contexts, including drug design, docking, ligand screening, structural homology modeling, protein function assessment, and more.

4.2.3 Requirements for Uncertainty-aware Protein Visualization

Based on the previous state-of-the-art analysis and application analysis, we are able to determine a list of requirements, which will be presented in the following. We followed the suggestion of Lam et al. [146], where interviews were described as a proper tool for understanding the needs of users. Here, we used the requirements defined by Gillmann et al. [79] that are formulated to promote the real-world use of novel visualization approaches. The list contains 16 *low-level requirements*, which are sorted into 5 categories (usability, effectiveness, correctness, flexibility, and intuitiveness). We showed the list to a domain expert from biochemistry and a visualization expert to cover both views on the proposed topic. First, we let both experts express the importance of fulfilling each requirement. Here, a Likert scale was used (1 unimportant, 5 very important). The results can be found in Table 4.1.

It can be observed that avoidance of clutter, uncertainty visualization, different use cases, interactivity, and ease of use are highlighted as important by both experts. Further, interactivity and use time efficiency are also listed as very important or important by at least one of the experts. As these requirements are rather general and low-level, we have used them to derive *high-level requirements*, which can be found in the following. The list is further maintained by the requirements to achieve uncertainty-awareness in VA tasks developed by Sacha et al. [203].

R1: Visualization of positional uncertainty. As the captured position of atoms in a protein can have a huge impact on the computation of properties of the considered protein, the propagation of these uncertainties is required. Here, every computational step based on atom positions needs to be adapted according to the underlying uncertainty captured or computed for each atom [41].

R2: Integration into known visualizations. There exists a consensus of visualization techniques that are suitable in the biochemical domain [143]. These visualizations have been proven to fulfill the requirements of molecular visualization. A visual representation that includes uncertainty information should be an extension of known visualization paradigms.

R3: Avoidance of visual clutter. Visual clutter, in the sense of this application, refers to the numerous occlusion of objects or information by overlying objects, resulting in a tangled visualization. As many atoms are displayed in a 3D scene, potentially even displaying the superposition of various protein models at the same time, summarizing the available information is an important step in communicating the available data in a compressed manner. A suitable visualization strategy should

reduce visual clutter as much as possible to allow the user to understand the outline of the dataset.

R4: Interactive visualization application. As the visualization of proteins usually results in a three-dimensional object, users need to be able to explore the data utilizing suitable interaction paradigms [79].

4.3 Related Work

In this section, we provide a summary of related work in terms of UA molecular visualization approaches as well as open-source applications that implement these approaches.

4.3.1 Uncertainty-aware Protein Visualization

In the field of visualization, the inclusion of uncertainty was classified as one of the most important research problems by Johnson [126], as it cannot be implemented right away. Brodlie et al. [27], as well as Potter et al. [191], divided uncertainty visualization challenges using the dimension of their data and the dimension of data points. Here, we obtain a valuable starting point as we can consider molecular data as scalar data.

Molecule and protein visualizations are widely used. Therefore, a large number of projects have been addressing issues of biochemical, pharmaceutical, and medical researchers as well as their industry members. Kozlíková et al. [143] summarized a variety of visualization options in a state-of-the-art analysis. Here, the selection of a proper uncertainty representation was named as one of the main challenges [50]. In the following, the most important UA protein visualization strategies related to our approach are summarized.

Rheingans and Joshi [199] used various family members of molecules holding the same atoms and bonds. They either superimpose the members showing regions of high uncertainty by large disagreement between the confirmation states or show iso-surfaces using Gaussian splatting to indicate the likelihood of an atom being located at a set location. Although this provides a visualization of all potential protein positions, it introduces visual clutter in the resulting visualization. Instead, the reduction of visual clutter in these visualizations is a focus of this application.

Rasheed et al. [197] utilized volume rendering to show the per voxel uncertainty computed across an ensemble of slightly perturbed samples of the same molecule. It showed that the B-Factor uncertainty correlates with this distribution function uncertainty. Knoll et al. [141] provided a volume-rendered uncertainty classification based on electron density distributions using 2D transfer functions, helping to identify interfaces based on chemical bond forces. Skånberg et al. [217] used volume rendering of spatial distribution functions to visualize the distribution of selected structures over ensembles of molecules.

Schulz et al. [211] presented a model visualizing the uncertainty of secondary structure assignments on ribbon diagrams, comparing various assignment algorithms. In contrast to this work, they used various assignment algorithms as their source of uncertainty instead of the positional uncertainty of atoms. The visualization was made more squiggly in areas of high uncertainty instead of using iso-surfaces for the visualization where the original geometry can still be seen. In contrast to this contribution, we aim to focus on the positional variations of atoms in proteins as a source of uncertainty while preserving the original shape of the ribbon model.

Lee and Varshney [148] created a UA 3D visualization of the solvent-excluded surface using Gaussian distributions and a fuzzy rendering mode. Even though this approach seemed promising, the appearance of the results is comparable to blurring the original visualization.

Sasisekharan [206] developed the usage of dihedral angles to describe polypeptide conformations. This so-called Ramachandran plot was used throughout biochemical research quickly [196, 195]. A UA version of this plot, indicating the variations in the dihedral angles, is provided here.

4.3.2 Uncertainty-aware Protein Visualization Software

A major drawback of all presented visualization approaches is that they are rarely used in open-source visualization tools. As Gillmann et al. [85] pointed out, this is a desirable feature in many applications. The protein data bank provides a list of open-source visualization tools in general [13]. We use this list to extract the tools that incorporate uncertainty.

Chimera is an open-source tool for visualizing molecular structures [185]. Variations in molecular modeling can be visualized by plotting multiple proteins at the same time. Atoms can be exchanged, and a Ramachandran plot can be used to examine the effect of the deviations in atom positions. Here, the resulting visualization

can quickly become cluttered. In the presented approach, various protein models are summarized within one visualization, such that the amount of visual clutter is minimized as much as possible.

Polyview 3D [190] allows for the visualization of multiple positions of atoms in a protein using animation. Although this provides a nice visualization of different protein formations, it is only able to show one formation at a point in time. In contrast to this, our approach aims to show all confirmations simultaneously if requested while giving users the freedom to watch any formation on demand.

The molecular visualizer iMol [186] uses motion blur to indicate variations in atom positions. Although this shows the uncertainty of atom positions, it causes visual clutter if a large number of blurred areas exist.

Swiss PDBViewer [97] represents uncertainty using color-coding. Here, atoms or amino acid residues that hold high amounts of positional uncertainty are shown in red highlight color. This allows for the visualization of positional uncertainty without introducing additional visual clutter, allowing it to show the positional displacement in space. Instead, the presented approach seeks to provide a trade-off between minimal visual clutter and the inclusion of the potential positions of an atom.

4.4 Uncertainty-aware Description and Properties of Proteins

4.4.1 Uncertainty-aware Description of Molecular Data

As shown in Section 4.2, there are several sources of uncertainty and different ways to express them when capturing biochemical data. When not considering uncertainty, atom positions are usually treated as fixed Cartesian coordinates. As this ignores the fact that atoms have a certain range of movement, captured by the B-Factor or multiple models of the same protein, the uncertainty of atoms will be described in an isotropic and anisotropic way.

To make use of the available uncertainty information, we utilize an extended description of points that allows the visualization to capture positional uncertainty, introduced by Gillmann et al. [86], referred to as probabilistic points. It describes every point by a center Σ and three orthogonal vectors $\sigma_1, \sigma_2, \sigma_3$ to describe the available movement in each direction. Figure 4.2 shows how each atom in a protein

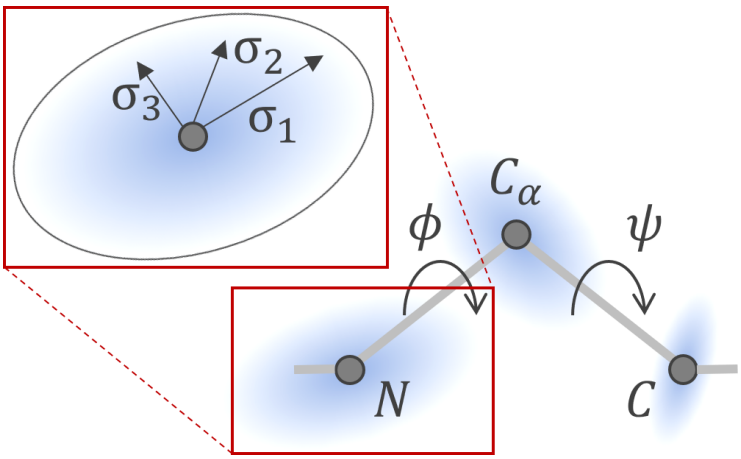


Fig. 4.2: Uncertainty-aware description of atom positions in a protein using three eigenvectors of the standard deviational ellipse.

can be modeled as a probabilistic point. To achieve quantification of uncertainty for the underlying data, two types of uncertainty descriptions for molecular data have to be distinguished: *isotropic* and *anisotropic*. Those orthogonal vectors do not have to be axis-aligned and either depend on the distribution of the same atom in multiple models (anisotropic) or the B-Factor (isotropic).

Isotropic Model When using a single model with a B-Factor value attached to each atom, the root-mean-square displacement can be retrieved from the B-Factor, yielding an isotropic model for the atom's movement. To highlight regions of high or low uncertainty, the normalized B-Factor, or the root-mean-square displacement, can be utilized to visualize the isotropic model. Normalized uncertainty can be scaled using a parameter to visually encode differences better. Our collaborators highly recommended this when working with isotropic models, helping to distinguish uncertainty information better.

In the isotropic case, the measured position of an atom is its equilibrium position, used as the center of a probabilistic point (Σ). This means that the model used to create data estimated this position using the most likely conformation simulated. Additionally, the σ-values of each probabilistic point will be set to the root-mean-square displacement (u) retrieved from the B-Factor of the considered atom as shown in Equation (4.1) [34].

$$B = 8\pi^2 u^2 \iff u = \sqrt{\frac{B}{8\pi^2}} \tag{4.1}$$

This is a suitable assignment, as the root-mean-square displacement captures the available movement of the respective atom in each direction. Although probabilistic points can be modeled with anisotropic σ-values in each dimension, the available B-Factor is an isotropic description of the positional uncertainty. Therefore, each dimension gets the same σ-value assigned. In total, Σ and σ can be utilized to define a three-dimensional distribution function that is able to output the Gaussian probability density for an atom to be located at an arbitrary point in a three-dimensional space.

Anisotropic Model On the other hand, multi-model data, like *Nuclear magnetic resonance* model ensembles, capture multiple positions of each atom. This leads to an anisotropic distribution of points around their average. Therefore, an anisotropic approach can be used instead. For this purpose, the average position of each atom over all available models has to be calculated first. Those average positions are needed to find the covariance matrix for each atom thereafter. In this way, the distribution of each atom over all models can be described by retrieving the eigenvectors and corresponding eigenvalues of said covariance matrices using a process called *Eigendecomposition*. The eigenvector with the biggest eigenvalue describes the direction with the most substantial standard deviation, while the eigenvector with the smallest eigenvalue gives the direction of the least substantial standard deviation. This way, the normalized eigenvectors can be multiplied with the square root of their eigenvalues to create $\sigma_1, \sigma_2, \sigma_3$ [243]. As the covariance matrix is symmetric, the eigenvectors are orthogonal, which can be used to create a *Standard Deviational Ellipse* (SDE) around each Σ. It should be noted that the SDE is actually no ellipse [71], but a special type of curve [91]. This fact was ignored to preserve the directions of the eigenvectors better and to save time while computing uncertainty contours.

4.4.2 Uncertainty-aware Computation of Dihedral Angles

Besides the challenge that the representation of UA proteins varies from the classic representation of a protein, the UA description of proteins also adds additional information to their dihedral angles. This information contains the available change of said angles induced by the available movement of their atoms. In the following, both an isotropic and anisotropic representation of dihedral angles will be described.

Isotropic Model In the isotropic case, dihedral angles depend on the position of four atoms each. If the considered atom uncertainties describe a lot of movement, the corresponding dihedral angle can also strongly change. This should be reflected in the computation of the dihedral angle uncertainty. Therefore, the dihedral angle uncertainty consists of the average normalized B-Factors of the considered atoms. This directly links the angle uncertainty to the uncertainty of its atoms while removing any influence of the angle value itself. Also, the uncertainty can be directly scaled to show highly uncertain regions in the Ramachandran plot.

Anisotropic Model For the computation of dihedral angles using anisotropic uncertainty, the distribution of a dihedral angle can be extracted from its distribution over all models. Therefore, the average dihedral angle of each amino acid residue has to be calculated first. Then, the covariance matrix of each amino acid residue can be created. It should be noted that the distance metric has to be chosen with respect to angles, e.g., the distance between -179 degrees and 179 degrees is 2. Similar to the representation of probabilistic points, the eigenvectors and eigenvalues can be extracted from the covariance matrices. This way, the average dihedral angle and its distribution can be shown.

A consistent view of the underlying uncertainty of dihedral angles is given in both cases, as they are handled in a similar fashion to their corresponding atoms. The isotropic case directly shows the average uncertainty of the underlying atoms, whereas distributions of atoms or dihedral angles are shown in the anisotropic case. This visually and conceptually connects the 3D visualization with the Ramachandran plot.

4.5 An Application for Uncertainty-aware Visual Analytics of Proteins

We propose a visualization application for molecular data that is affected by uncertainty and provide a UA description of atom positions to show how this knowledge can be inserted in arbitrary computations based on these positions (see Section 4.4). To incorporate this knowledge, we propose a multi-view system that is composed of prominent visualization approaches in molecular biology, such as volumetric visualization, Ramachandran plots, and statistics views. We adapted each of the visualization approaches such that they are able to indicate the positional uncertainty

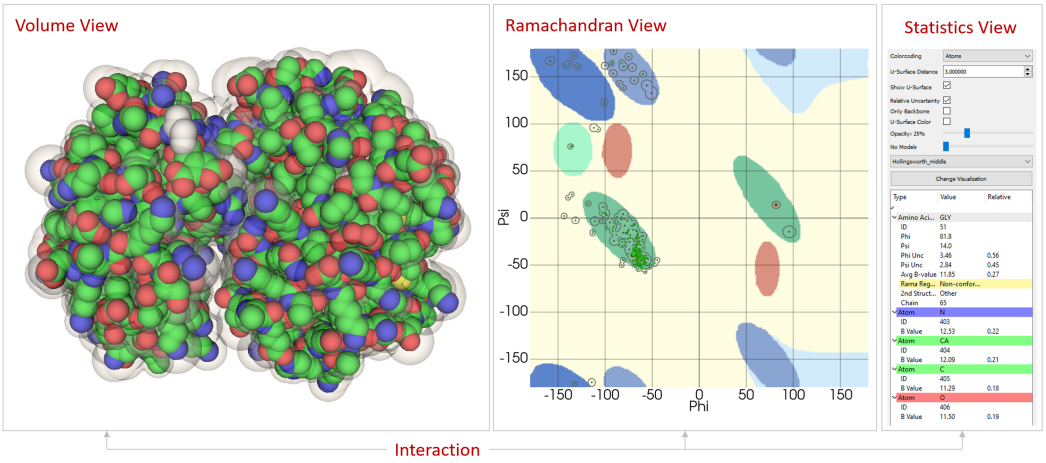


Fig. 4.3: Overview of the presented visualization application. The application consists of a volume view, a Ramachandran plot view, and a statistics view. The views are linked with a hover interaction methodology. Here, the 1H97 protein is shown.

of atoms in specific proteins visually. The linked views are highly interconnected to provide user interaction for exploratory tasks (see Section 4.5).

We show the effectiveness of the presented application by applying it to real-world molecular data and demonstrate how the UA visualization approaches perform (see Section 4.6). Our results will be discussed in Section 4.8.

In order to devise a VA tool for researchers in biochemistry that allows users to review uncertainty in protein data, we created an interactive multi-view application. The application expands on well-established views in the biochemical community. According to our domain experts, volume views and Ramachandran plots are the most important visualization approaches that are used in daily tasks. Our system builds upon these two visualization strategies of proteins while including uncertainty information and allowing interaction.

4.5.1 Uncertainty-aware Volume Visualization

One of the major goals of this work is to enhance the current visualization capabilities by including uncertainty information. The 3D visualization methodologies utilized are presented here. The inclusion of uncertainty in the field of proteins consists of showing possible movements of atoms in the examined protein. Therefore, a mutable transparent isosurface around known visualizations is used to provide the original and UA version in a mixed display, shown in Figure 4.5. Using transparency and color controls, the user can blend in the uncertainty information as much as

desired. The goal is to create a three-dimensional barrier that indicates the potential mobility of protein atoms considering specific possible freedom of movement. Throughout discussions with our collaborators in the biochemical domain, we determined that uncertainty contours should be computed based on the underlying geometric representations of proteins.

3D Protein Visualization Types

The Van der Waals surface visualization draws a sphere for each atom with the Van der Waals force of the corresponding chemical element as its radius. The sphere representation is stored as a geometry, which will be referred to as g hereafter. For any geometry, $b(g)$ is defined as the boundary of g, including minimum $b(g)_{min}$ and maximum $b(g)_{max}$ of its points in space. The Ball-and-Stick visualization and the Solvent Accessible Surface use either constant radii or are set to the Van der Walls radii plus a constant solvent radius.

Apart from these spherical visualizations, the Ribbon model is also implemented, drawing a spline curve using the backbone atoms of a protein as their control polygon. The spline is represented by triangles, receiving their uncertainty information from the closest backbone atoms. This is needed as the Ribbon model only consists of triangles. To achieve a mapping between a triangle to a B-Factor, the closest atom to each vertex of a triangle is found and their uncertainty information is averaged. This allows for sufficiently fast computation while still retrieving a well-suited estimate. Without averaging, one of the vertices would have to be picked using the uncertainty information of the closest atom, leading to high inaccuracy in some cases. Another approach would be to average the positions of triangle vertices and then find the closest atom to the middle.

Creating the Uncertainty Scalar Field

In order to create a surrounding contour, a structure that represents the distance of each point in space to the closest primitive in an atom representation is required. Although this task can be solved analytically, it would result in significant computational effort. Also, it would have to be completely recomputed when choosing a different σ-distance from the geometry. In order to reduce the computational effort and to allow isosurface scaling, a scalar field F, using the boundaries mentioned in Section 4.5.1, is overlaid with the original protein representation. Depending on the resolution of this scalar field, we are able to discretize the distance of points

in space to the closest point of the protein geometry with relatively high accuracy. The distance to the nearest piece of geometry, measured in a directional σ-value, is evaluated for each scalar in F. Measuring the distance in units of σ is needed as each atom has a different σ-value. This approach presents a clear advantage in that the scalar field has to be computed only once, such that isosurfaces can be created at any σ-value chosen by the user.

Distance of a Voxel to an Atom under Uncertainty

To compute the distance of an atom to a voxel center measured in a directional σ_d unit, first, the point of intersection I of an SDE with the line between the voxel center S_c and the ellipse center E_c has to be found. Then, σ_d is found by taking the distance between I and E_c and subtracting the radius R.

$$\sigma_d = \|E_c - I\| - R \tag{4.2}$$

σ_d is a measure of one standard deviation in a direction given by E_c and S_c. Therefore, the distance between E_c and S_c minus the radius has to be divided by σ_d to retrieve the distance between the atom and the voxel in units of σ_d, allowing the user to later scale the contour to their liking.

$$D_\sigma = \frac{\|S_c - E_c\| - R}{\sigma_d} \tag{4.3}$$

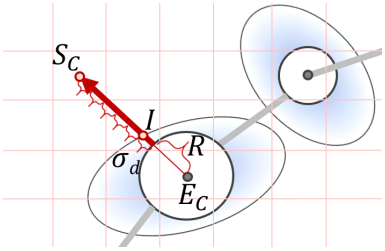


Fig. 4.4: Computation of a scalar field yielding discretized values of differences to a considered geometry in σ scale. The example shows the scalar at the tip of the red vector is 6σ away.

In Equation (4.2) and Equation (4.3) the radius R always has to be subtracted as it is a property of the representation itself, thus it is not changing when the user chooses a different standard deviation threshold. It should also be noted that using the isotropic model results in a simpler calculation of σ_d, as the uncertainty is the same in each direction, i.e., $\sigma_d = \sigma_1 = \sigma_2 = \sigma_3$. This function, calculating the distance

between voxel center S_c and atom center E_c, with ellipse axis $\sigma_1, \sigma_2, \sigma_3$ and radius R, will be called $\sigma(S_c, E_c)$ for later computations. Figure 4.4 depicts the computation of the distance between an atom with anisotropic uncertainty and a cell center of the scalar field.

As described in Section 4.5.1, the Ribbon model is represented by triangles such that the distance computation needs some modification. Obviously, the distance of a point to a triangle has to be found first [128]. Then, a sigma value has to be assigned to the triangle for the isotropic case. In this case, we chose to average the contribution of the closest atoms to each vertex. In the anisotropic case, the center of the triangle is used to represent the triangle as the center of an ellipsoid, using the properties of the closest atom to that point.

Iso-surfaces are utilized to extract the uncertainty contour based on the computed field. As the values in the scalar field represent the distance measured in a directional σ distance from the next geometry, a surface (e.g., with distance 1σ from the geometry) can be created showing the possible mobility of the atoms in their equilibrium position under a statistical model. Therefore, changing the iso-surface threshold is a simple operation and can be done seamlessly on modern hardware.

Filling the Scalar Field

For each of the considered geometries, the boundary of the scalar field has to be calculated with an as-small-as-possible extent, allowing enough space to fit the iso-surface but small enough to keep the resolution high and the size of voxels low. Therefore, the boundary $b(g)$ of the original geometry is offset in each direction by a constant value (c) that depends on the expected maximum offset and maximum uncertainty, resulting in a bounding box $b(F)$ of the scalar field F.

To calculate the scalar field for the sphere visualization, a Kd-Tree saving all atom positions is generated. This structure allows for an efficient search of close points in space. Then, for all voxels in a discrete scalar field, created as described above, the n atoms closest to its center are saved to a list L, efficiently created by the aforementioned Kd-Tree K. Then, for each atom position on this list, the distance, measured in a directional σ-value, is calculated from the voxel center to the surface of the current representation. Therefore, the $\sigma(S_c, E_c)$ function, calculating the directional σ distance, is used as described in Section 4.5.1. The lowest σ-distance is then assigned to the voxel.

The number n is the amount of closest atoms that are examined to determine the probability of a protein being located in a specific cell of F. It is a value that influences the quality of the scalar field, especially if the minimum and maximum σ-values strongly deviate throughout the dataset. As the radius and σ-values often strongly differ from each other when scanning neighboring atoms, this value should be kept at roughly 10% of the atoms. Otherwise, the iso-surface may lose its elliptical form when using high-scaling factors.

To achieve an uncertainty contour based on the scalar field F, a scaling factor f is chosen. This factor influences the isovalue at which the isosurface is created. Users can define this factor to set the distance in standard deviations (anisotropic) or root mean squares (isotropic). In principle, this factor can be set to an arbitrary number greater than zero, but in reality, 3σ usually encapsulates all available variations. Thus, higher choices are often not needed when ignoring extreme outliers. Figure 4.9 shows an example using $f = 1\sigma, 2\sigma, 3\sigma$, while 3σ encapsulates even the strongest outliers of the dataset.

Protein Representations

As shown in Section 4.2, there exist several ways to visualize proteins in 3D. We provide four protein representations: the space-filling model, the ball-and-stick model, the ribbon model, and the SAS (solvent-accessible surface). Figure 4.5 provides all four visualization types with a B-Factor color-coding. Each of the visualization types can be colored according to various features of a protein. This helps to encode important aspects of the amino acid residue that users are interested in. Usually, color is set per atom, but as the ribbon model does not have single atoms, each vertex is colored the way the closest atoms would be colored. In between points, the color is interpolated to receive smooth color-coding throughout the surface.

The resulting uncertainty contour can either be colored consistently or according to the relative B-Factor of the closest atom according to uncertainty. A consistently colored contour is shown in gray, preserving the color-coding underneath. Otherwise, the B-Factor color-coding (green to red) is used. Additionally, any transparency level can be chosen. Figure 4.6 compares the gray contour with B-Factor colored atoms and the color-coded uncertainty contour with gray atoms.

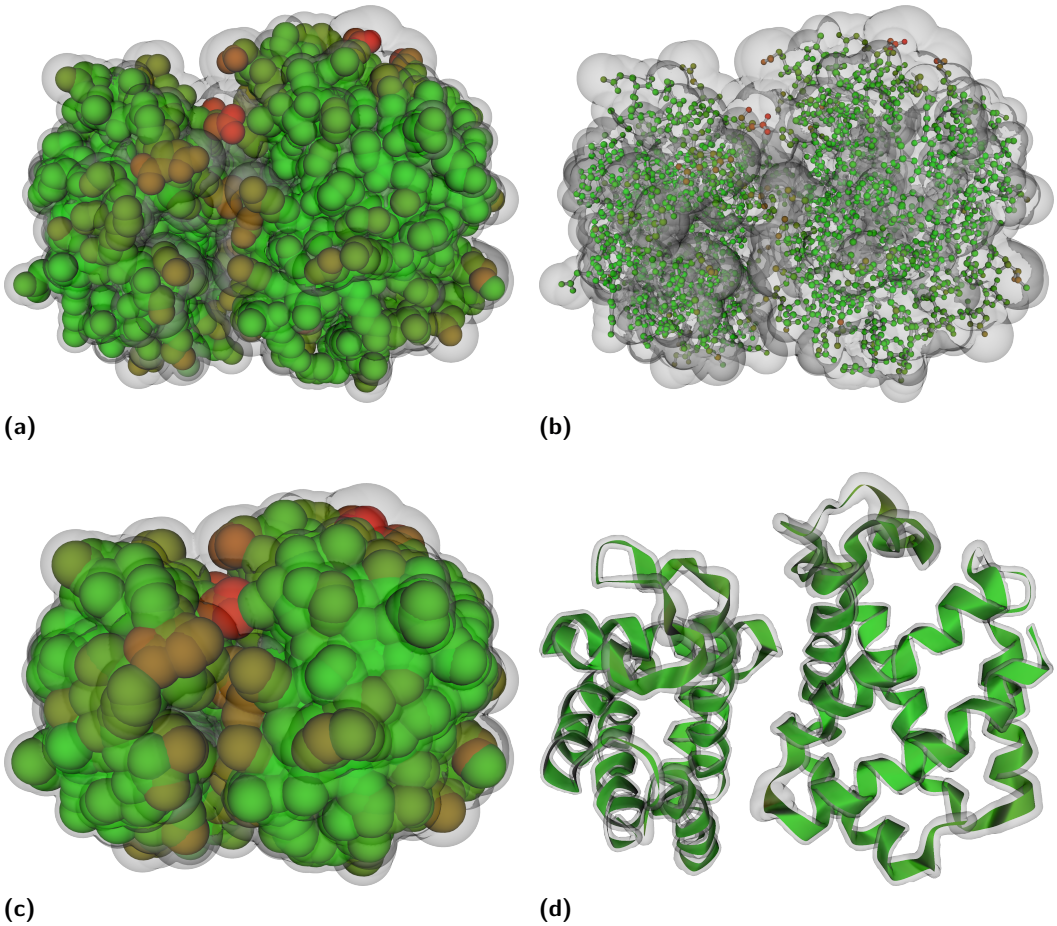
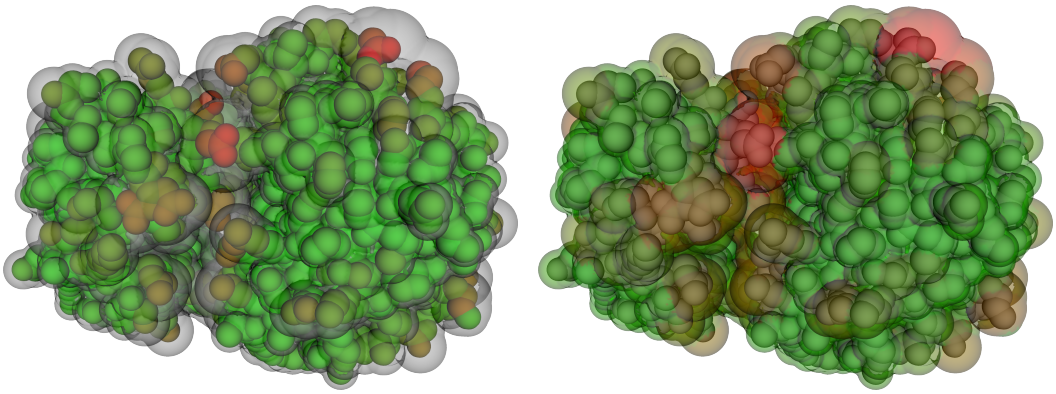


Fig. 4.5: Uncertainty-aware volume visualization of the 1H97 dataset [183], using different geometries for visualization. a) Sphere visualization. b) Ball-and-Stick visualization. c) Solvent Accessible surface visualization. d) Ribbon model visualization.

4.5.2 Uncertainty-aware Ramachandran Plot

A Ramachandran plot displays the distribution of dihedral angles in a protein. Throughout the years, by an empirical analysis of dihedral angle distributions, many nomenclatures of this plot, delivering the background maps, have been found. As scientists nowadays use a lot of different background maps, a suitable application does need to support multiple Ramachandran nomenclatures. Our tool is able to load any nomenclature provided in a given format, allowing the user to change the color of regions and also mapping each nomenclature to a color-coding for the 3D visualization of any geometry type.

The creation of isolines in the isotropic case works similarly to the isosurface creation for 3D geometries. Therefore, a 2D scalar field is being created, filling it with the



(a) (b)

Fig. 4.6: Uncertainty-aware volume visualization of the 1H97 dataset[183] at 35% uncertainty transparency. Gray uncertainty contour with B-Factor color-coded atoms (a) and B-Factor colored uncertainty contour with gray atoms (b).

nearest dihedral angle combination while considering its uncertainty in the ϕ and ψ direction, called $phiu$ and $psiu$. Each point of the dataset is modeled as an ellipse with (ϕ, ψ) as its center and ($phiu$, $psiu$) as its axis (distance to the center in ϕ and ψ direction). In the end, the marching squares algorithm implemented in VTK creates the isolines at a distance of 1 to draw the ellipses directly, removing intersecting parts. As the ellipses do not visually clutter the image in any way, they are always shown. An example is provided in Figure 4.3.

In the anisotropic case, the distribution of each dihedral angle over all models is shown. As described in Section 4.4.2, the average dihedral angles, eigenvectors, and eigenvalues are used to display average positions and uncertainty of these positions. The average dihedral angles are shown as points on the plot while displaying the eigenvectors and eigenvalues as non-axis aligned ellipses using the eigenvectors for directions and the eigenvalues for scaling of each direction. Similar to the isotropic case, intersections are omitted using a scalar field and the marching squares algorithm. Figure 4.9 shows this using the 1G03 dataset.

4.5.3 Uncertainty-aware Statistics View

Besides the volume and Ramachandran view, biochemists also need to be able to look at the raw data. Therefore, the statistics view is shown next to the Ramachandran plot and the 3D Visualization, as shown in Figure 4.3. It includes values such as amino acid, atom type, and radius. In addition, we provide the amount of uncertainty

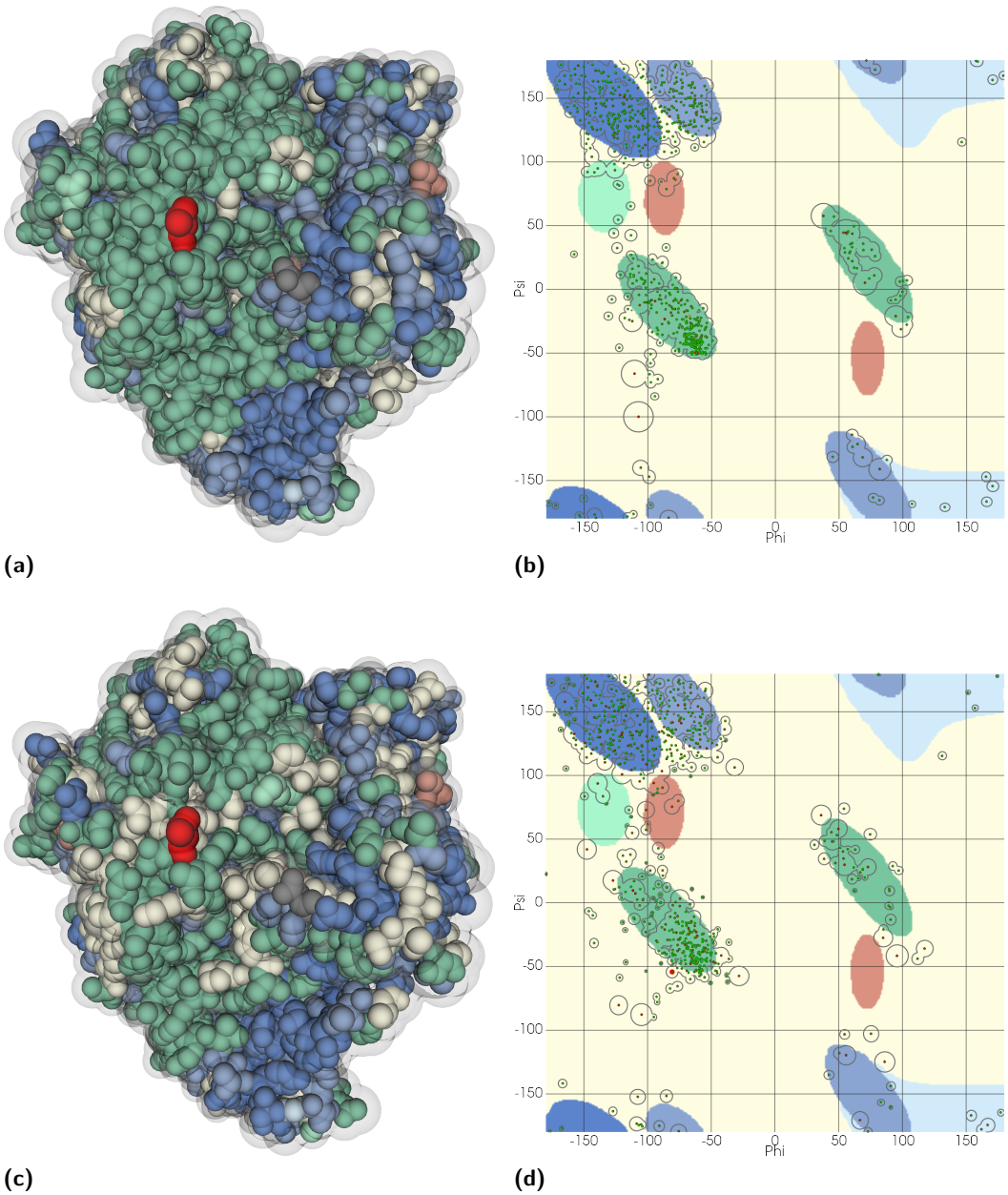


Fig. 4.7: Cyclodectrin glycosyltransferase with and without mutagenesis. a) Uncertainty-aware visualization of 1CGT with mutagenesis in ball view style containing the uncertainty contour and a selection of a peptide on the active site. b) Uncertainty-aware Ramachandran plot of 1CGT with selection highlighted according to a). c) and d) show 1CGT without mutagenesis. The selected amino acid residues in both proteins are identical, considering their ID, i.e., their index in the amino acid chain.

in each atom, showing a raw view of the values captured for each atom and residue. Here, two different modes are available.

If no amino acid residue is selected, the detail view shows a number of important features of the dataset, like the ID code, a unique identifier of the Protein Database, and general uncertainty information for the dataset. A reference is needed to correlate B-Factors with each other. Here, the B-Factor range and its average value are displayed. The same is provided for the averaged B-Factors of the amino acid residues, while it should be noted that this average does not equally consider all atoms, as amino acids have different amounts of atoms in them. For the analysis of dihedral angle uncertainty, their uncertainty range and average are also provided.

When an amino acid residue is being picked by the user, the detail view changes to an elaborate display of the amino acid residue and its atoms. General information like its ID and type of amino acid is provided next to the dihedral angles, including their uncertainties, average B-Factor, Ramachandran region, and secondary structure type. The chain ID and Residue ID might be interesting for identifying certain amino acid residues directly. Each peptide is provided with detailed information about the involved atoms, displaying the element, ID, and B-Factor. An example is shown in Figure 4.3.

Normalized B-Factors are often used for comparison [253]. Therefore, normalized values for all B-Factors and dihedral angle uncertainties are provided next to their absolute value in the statistics view. This allows users to get a sense of high and low uncertainty in the viewed dataset.

4.5.4 Interaction

Our presented system is designed to consist of multiple views that are highly interconnected and linked.

The interaction between the 3D view, the Ramachandran plot, and the detail view is a point-to-show implementation. An example can be seen in Figure 4.3. When hovering the cursor over either an atom in the 3D view or a data point in the Ramachandran plot, the corresponding amino acid residue is shown in the 3D view along with the Ramachandran plot and the detail view. This enables a fast correlation of different visual aspects of a protein to be examined. The 3D view displays the amino acid residue by drawing its atoms as red spheres, whereas the Ramachandran plot highlights the corresponding data point with a red disk around it. All important information is provided by the detail view. When desired, the selection can be fixed to a certain amino acid residue by clicking on it using the right mouse button. Right-clicking again resets the selection, highlights another residue,

or shows the general dataset information, depending on the mouse cursor position. Through this mechanism, a set of amino acids can be depicted such that they can be examined in their entirety.

The uncertainty contour controls are another important feature. Users are allowed to toggle the contour on and off, use the relative or absolute B-Factors, toggle between the gray and colored contour, and set the transparency of the contour. Transparency is especially important, as surrounding a geometry with any transparent surface always partly blocks the view of some features of the object. In this case, the color-coding of the original geometry might be harder to see. A transparent contour better preserves the features of the original geometry, making it harder to see the details of the contour itself. An opaque contour shows depth information in a clear way but blocks the view of the underlying geometry.

4.6 Results

In the following section, the presented approach is used to create UA visualizations of real-world proteins. The use cases have been selected by a domain scientist who considered them interesting in terms of uncertainty analysis. All use cases are known, as this allows us to check if the uncertainty visualization indicates the aspects of uncertainty visualization that are relevant in the respective case. The datasets were used in conjunction with our approach, and the findings were discussed with our collaborator. The presented approach was implemented in C++ using the VTK [209] library with the OSPRay [241] rendering backend, using Qt [227] for GUI design.

4.6.1 Monomeric Hemoglobin from the Trematode Paramphistomum Epiclitum

As a first example, we present a small but interesting protein selected by our collaborator to obtain a first understanding of the presented visualization approach. Fasciola is a type of fluke commonly known as liver fluke. It is a parasitic organism that infects the liver tissue of a wide variety of mammals, including humans, in a condition known as fascioliasis. Scientists became very interested in monomeric hemoglobin (1H97 [183]) of this organism as it could be the key to developing a vaccine against the parasite.

We use this protein to provide an overview of the available visualization approaches. The underlying computational model of uncertainty is isotropic as we consider the B-Factors assigned to each atom in the protein originating from the PDB file. Figure 4.5 shows a variety of UA volume visualization approaches to represent 1H97. The different types of visualization (sphere visualization, ball-and-stick visualization, solvent-accessible surface visualization, and ribbon visualization) are color-coded with the respective B-Factor (green: low B-Factor, red: high B-Factor). In addition, we show the uncertainty contour indicating the potential spatial displacement of atoms. Here, we can see that the surface differs from the original visualization when the B-Factor of nearby atoms is high. This confirms the computational setup of the presented approach.

Figure 4.3 shows the UA Ramachandran plot of 1H97 in the middle. We can directly see that the computed dihedral angles are, in general, very stable according to the spatial movement of captured atom positions. This can be seen by the relatively narrow uncertainty bounds around the visualized points which indicates that the spatial movement of atoms is small. Our collaborator confirmed these findings of our visualization approach. In general, most angles lie in desired areas, and the consideration of potential changes in the atom position does not change this impression.

It can be seen that the proposed visualization approach helps to confirm the stability of a protein considering uncertainty information. We also showed throughout this section that a variety of visualization approaches can be adapted using our proposed visual metaphor of uncertainty contours.

4.6.2 Cyclodextrin Glycosyltransferase

The second example is cyclodextrin glycosyltransferase. The example was chosen by our collaborator as the protein is one protein he is interested in examining the variability of simulation results. This protein is able to produce cyclodextrins from starch, which is an important process in the production of drugs, as cyclodextrins help transport certain molecules in an efficient manner. In the presented example, the original cyclodextrin glycosyltransferase (1CGT [140]) is compared with cyclodextrin glycosyltransferase that is affected by mutagenesis (1CGU [139]) in the active site of the protein. During the process of mutagenesis, certain amino acids are exchanged, which can affect the function of the protein. As the active site of the protein is the main catalyst of a chemical reaction, this part of the protein needs further

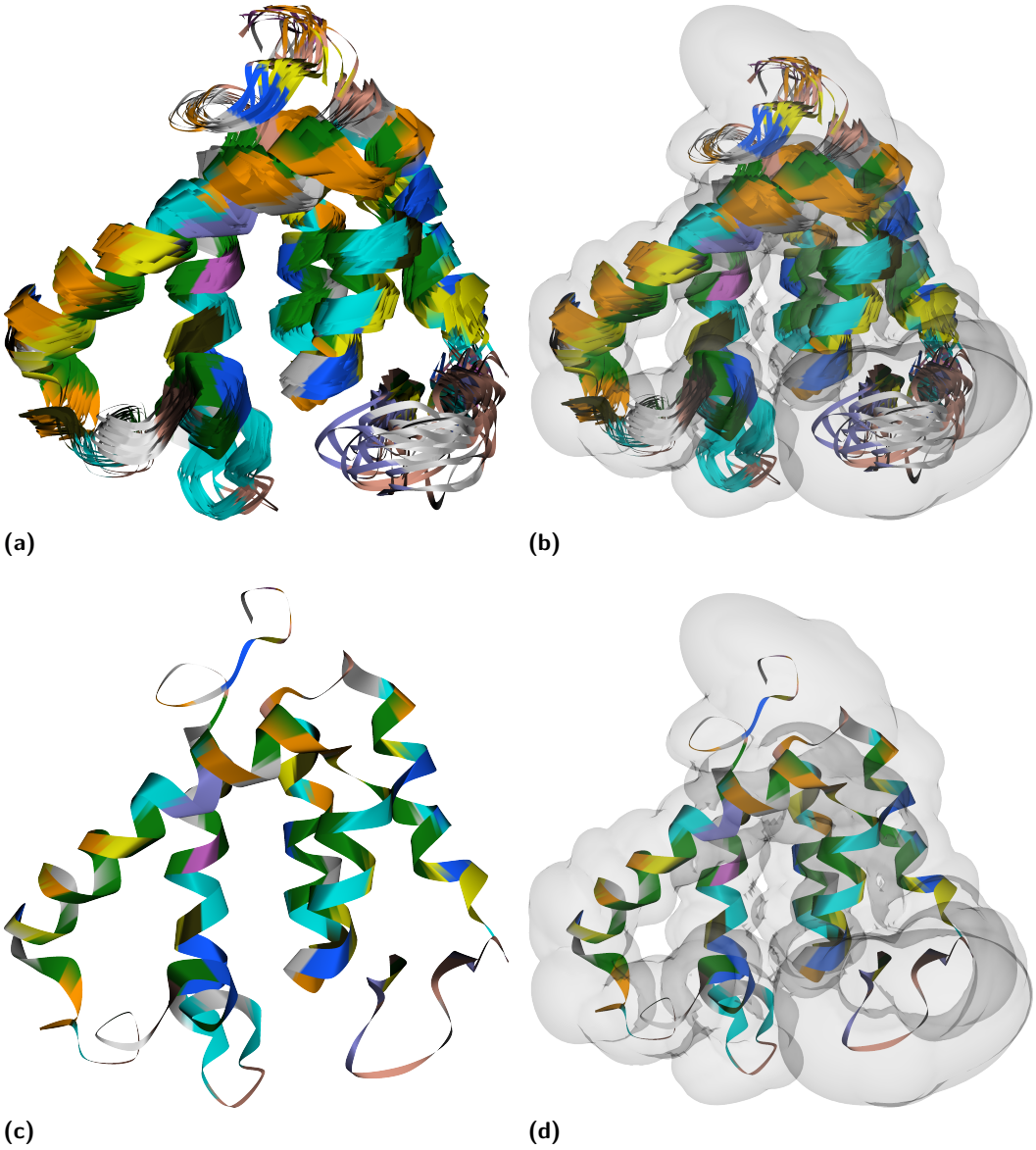


Fig. 4.8: 20 Models of 1G03 protein shown with different styles of visualization. a) Superposition of all models as normally used in the biochemical domain. b) Superimposed model with 3σ uncertainty contour. c) Average model. d) Average model with 3σ uncertainty contour.

examination in terms of stability and uncertainty. The data is provided as an isotropic model of uncertainty in this example.

Figure 4.7a shows the cyclodextrin glycosyltransferase (1CGT) without mutagenesis using the ball view. Color-coding reflects the area in the Ramachandran plot where the respective backbone amino acids are located. We selected an amino acid residue on the active site to review the thermal stability of the protein. Showing the

uncertainty contour of the protein, it can be seen that the contour is displaced equally throughout the three-dimensional space without larger outliers in the spatial displacement. Figure 4.7b displays the UA Ramachandran plot of 1CGT. Here, it can be seen that most amino acid residues are located in desirable areas of the Ramachandran plot. There exist outliers in the lower-left corner, where we figured out that they are not included in the active site of the protein and, therefore, are not of main interest. The uncertainty contour around the remaining amino acid residues shows that although spatial displacement of the proteins can be observed, most amino acid residues will not leave the desired areas in the Ramachandran plot. This helps to determine that the current composition of the protein seems to be stable. When considering the active site of 1CGT, we can see that all residues are located in the dark green areas, which are stable.

In contrast to the finding of 1CGT, Figure 4.7c shows 1CGU, a cyclodextrin glycosyltransferase that is affected by mutagenesis. The color scheme, uncertainty contour, and performed selection are identical to the ones in Figure 4.7c. When reviewing the volume views it can be seen that the uncertainty contour does not change optically, meaning that the spatial displacement of the atoms in the protein is not affected by the mutagenesis. On the other hand, we can directly see that the color-coding of several amino acid residues changed in the volume view. This means that several amino acid residues are now located in another area of the Ramachandran plot compared to the original protein. Overall, we can detect more amino acid residues that are located in undesirable areas (light yellow). When reviewing the active site (selection made in red), the selected amino acid residue in the active site is no longer located in a stable region (see Figure 4.7d). Although the UA Ramachandran plot represents the spatial displacement of atoms, the selected amino acid residue will not be located in a stable area.

In general, when reviewing the Ramachandran plot of 1CGU, we can identify more amino acid residues that are not located in a stable area of the Ramachandran plot. This becomes more critical when considering the uncertainty spheres around each amino acid residue in the Ramachandran plot. Here, a variety of amino acid residues could leave stable areas when considering the spatial movement of atoms. Our collaborator from the biochemical domain supported these findings and highlighted how the visualization identifies this easily.

Overall, this example shows that 1CGT is a more stable protein (especially in terms of the active site) than 1CGU. As a result, our approach is suitable for understanding the stability of proteins in direct comparison while considering the uncertainty included in each of the computational models.

4.6.3 N-terminal Domain of the Human T-cell Leukemia Virus Capsid Protein

HTLV-I is a virus that binds in the human body and can cause leukemia or neural disorders. An important structure that is involved in this process is the 1G03 protein [45]. Our collaborator chose the example as it provides large positional uncertainty that influences the research conducted with this protein. In order to understand the structure and function of the protein, researchers aim to synthesize it. The peptide chain that builds this protein is pretty clear, but its three-dimensional folding in space may vary according to a variety of factors, such as temperature or other binding proteins. In this context, 20 simulations exist that capture the three-dimensional folding process of 1G03.

Figure 4.8 shows different visualization approaches for the 20 models of 1G03. Biochemists usually review these datasets using a superposition visualization. Although all models can be reviewed at the same time, the visualization is very cluttered. Especially inside areas where the depicted models disagree, it is hard to determine the different models and how they are located in space, as shown in Figure 4.8 a). Adding the proposed uncertainty contour helps biochemists to examine the potential space where a protein can be located in. Figure 4.8 shows the superposition visualization with the 3σ uncertainty contour. Although we still use the superposition models in this visualization, we can clearly show the user where proteins can be located in space. We allow this visualization in the current application in order to provide a mechanism to relate the presented visualization approach with already existing approaches. Figure 4.8 c) shows the computed average model of the 1G03 protein. Here, the visualization is less cluttered as only one model, which is composed of all 20 existing models, is displayed. This visualization is almost free of visual clutter but reduces the information captured in the 20 models. Based on the average visualization, we provide the final visualization approach that allows us to show the average model in combination with an uncertainty contour (Figure 4.8 d). Here, it can be seen that the average model is covered by the 3σ uncertainty contour. The contour helps to indicate areas in the protein that hold high amounts of positional uncertainty. For example, the top region of the protein holds high amounts of uncertainty, which is indicated by the large displacement of the uncertainty contour. Other regions, such as the center of the protein, hold a rather tight uncertainty contour, indicating a low variability in the underlying models. This shows that the presented visualization allows for an easy-to-understand representation of disagreement in computed models of proteins.

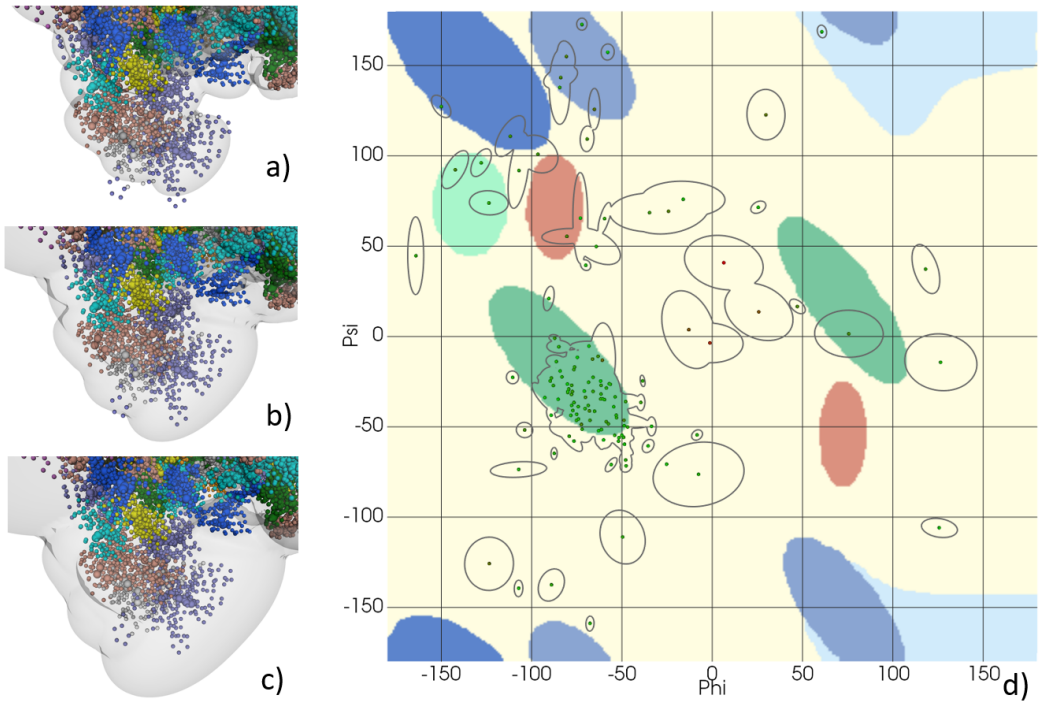


Fig. 4.9: Closeup of the region in the 1G03 protein with high amounts of uncertainty. a) 1σ uncertainty contour. b) 2σ uncertainty contour. c) 3σ uncertainty contour. d) Uncertainty-aware Ramanchandran plot.

Figure 4.9 shows the top part of the 1G03 protein that was identified to hold large amounts of positional uncertainty. In this example, the visualization of the protein was changed to a ball visualization, indicating the different types of amino acids. Here, we are interested in the amino acid TRP. Figure 4.9 a) shows that the uncertainty contour does not capture several points. When increasing σ to 2 (Figure 4.9 b), most of the existing models are included in the uncertainty contour. For 3σ Figure 4.9 c), all models are included in the uncertainty contour. The contour indicates the center of the existing distribution of amino acids and shows their spread in space. Here, it can be directly seen which direction is the most uncertain. Our collaborator confirmed that the visualization correctly indicates this distribution, helping him understand the positional uncertainty.

The example shows that our approach suits understanding the difference in multiple computational models. The number of models is not limited in terms of visual clutter as we are able to reduce the number of visual primitives due to the used average model.

4.7 Perspective on the Uncertainty-aware Visual Analytics Process

This section relates the presented application to the UAVA process described in Section 3.3 by following the proposed guideline of Section 3.4. Each step is described in its own paragraph, describing the design process without prior knowledge of the presented application. This simulates the design process of this tool to ease the creation of similar applications, providing an example for the execution of the UAVA workflow proposed in Section 3.4.

Design the VA Application At first, researchers in this scenario work with PDB (Protein Data Bank) files to check the stability of new simulated data. Other tools like Chimera [185], and Polyview 3D [190], iMol [186], either become too cluttered, show only one instance of a protein at a time, or use too simplistic depictions for their analysis.

The data provides atom positions and secondary structure information (helix-shaped, sheet-shaped, or random coil), but the dihedral angles, i.e., are not present yet and are calculated by the researchers in a preprocessing step. The combined information forms the *Dataset* component.

The researchers want to visualize their data using 3D renderings of the protein geometry and the 2D Ramachandran plot to control the stability of the protein. For 3D, the atom positions and secondary structure information are processed into various forms like Ball-and-Stick, solvent-accessible surface, or ribbon-style visualizations. The Ramachandran plot uses the preprocessed dihedral angles. Both visualization types fill the *visualization* component.

Most researchers' queries revolve around the stability of molecules, as newly simulated variants often exhibit instabilities that lead to defects and unattainable designs. The Ramachandran plot and color mapping of the B-value are available in this scenario to test the stability, providing a visual hypothesis test for the *Hypothesis* component and connecting it to the *visualization* component.

When analyzing the data, the researchers find unstable regions that must be optimized as their *Insight*. Those regions are enhanced in further simulations to generate new designs to be tested, powering the feedback loop.

Quantify System Uncertainty In designing the VA application, the researchers used the B-value, representing the isotropic movement of atoms, via color maps or analyzed multiple simulation results of the same protein by superposition. Still, they want to understand the extent and effect of the B-value or spread of the ensemble of molecules.

They find that the B-value partially stems from finite instrument resolution providing a fixed standard deviation per atom, and the ensemble resembles variations in observations due to minor changes in the starting conditions, providing an anisotropic distribution. Both types of uncertainty are 3D and quantifiable, allowing them to be represented in the 3D visualization. Those properties also hold true for the respective dihedral angles in each scenario (isotropic and anisotropic). Also, non-representative sampling could be a problem but is discarded as the software is not meant to integrate simulation software at this point. Mapping, interpolation, and shading are minor sources of uncertainty and are therefore disregarded in this application. It is simply known that they are schematic representations of atoms and proteins, so the deviations in their exact form or slight variations of the color-coding do not strongly influence the decision-making.

Both selected types of uncertainty are 3D or 2D with respect to the atom position or dihedral angle distribution. The B-value gives a discrete standard deviation, whereas the eigendecomposition defines a continuous ellipsoid of deviation around each atom or angle combination.

As the B-value directly corresponds to the standard deviation of movement per atom, uncertainty quantification is present via a given formula and propagated straight to the computation of dihedral angles. Eigenvalues and vectors must be computed via eigendecomposition for the ensemble of molecules. Similarly, the distribution of the dihedral angles in the Ramachandran plot is decomposed into eigenvalues and vectors per amino acid residue.

Integrate and Connect Uncertainty-aware Solutions When testing a molecule's stability, researchers need to find statistics to support or reject their hypothesis. Here, in addition to standard descriptions of atoms, their uncertainty in each eigenvector direction is provided. Alternatively, the standard deviation extracted from the B-value can be shown.

To ensure accurate data visualization, it's essential that the visualization component can effectively process the uncertainty information provided by the data component. This crucial step will help ensure that the insights gleaned from the visualizations

are reliable and can be used to make informed decisions. The researchers want to understand how far the atoms can move under the underlying uncertainty models. As blurring seemed to clutter the visualization too much in other applications, they want a contour around their visualizations that can be set to a user-defined threshold in standard deviations. Therefore, a scalar field is created that describes the distance of each point in space to the respective visualization in standard deviations. The researchers can interactively change this distance using the marching tetrahedra/triangles algorithm with different thresholds given in standard deviations. The procedure is similar for both 3D and 2D visualizations in this application.

As the researchers are interested in finding amino acid residues that make the protein unstable, the uncertainty of individual atoms can be summarized and displayed per amino acid residue.

Also, individual amino acid residues can be marked by the researchers to connect hypothesis testing with the visualization, delivering details on demand.

Feedback Loop Separation In defining the VA process feedback loop, the user's interaction with the application must be considered. As shown in Figure 4.3, the UA visualizations of the protein and the Ramachandran plot are shown next to each other, providing a statistics view with numerical data. When a user selects an atom in the 3D visualization, the Ramachandran plot highlights the corresponding dihedral angle pair and vice versa. Also, numerical information about the angles and considered atoms is shown, including their uncertainty information. The impressions obtained by the user cannot be directly quantified at this stage. Still, the uncertainty information from *UA Insights* can be used first to improve the regions most affected by uncertainty. This way, the most problematic configurations are handled, still giving the user more regions to look at and ordering them by uncertainty. This feedback can be used to generate new simulations with other software solutions that can be analyzed again in the present application.

Provenance Generation Implementing *Provenance* is carried out by the possibility to view all protein models in a single visualization while providing the distribution of models, showing them as iso-surfaces around the average model. To keep track of the individual models, the researchers can interactively scroll through them while maintaining the uncertainty representation, allowing them to find strong outliers. Here, Provenance uncertainty is only possible when using anisotropic models. In the isotropic case, the uncertainty directly stems from the uncertainty of the average model with the eigenvectors attached, created by summarizing all models.

Insight provenance, for example, could be recorded by the average uncertainty of the dataset in each iteration. It would be helpful to implement such a procedure to integrate simulations into the present application, tracking the uncertainty in each step. This would allow users to find occurrences where changes to certain areas resulted in higher overall uncertainty levels. The tracked uncertainty levels could then be made explorable by using a provenance exploration tool.

4.8 Discussion

4.8.1 Check of Low-level Requirements

Category	Requirement	Visualization Expert				Domain Expert			
		Imp.	PyMol	Protoshop	Our Tool	Imp.	PyMol	VMD	Our Tool
Usability	Collaborative	3	4	4	4	3	2	2	3
	Interactivity	4	4	4	4	5	3	3	3
	Avoidance of Clutter	5	4	3	5	4	2	1	4
	Minimized Input Parameters	4	4	4	4	3	2	1	3
	Compatibility	5	4	4	4	3	4	4	2
Effectiveness	Runtime Efficiency	4	3	3	4	2	3	3	3
	Memory Efficiency	4	3	3	4	2	3	3	3
	Use Time Efficiency	4	3	3	4	4	3	3	3
Correctness	Precision	5	4	4	4	3	5	5	5
	Quantification	5	4	4	4	3	4	5	4
	Uncertainty	5	1	1	4	5	2	2	5
Flexibility	Use Cases	5	4	4	5	4	3	4	3
	Different Datasets	5	4	4	4	4	3	4	4
Intuitiveness	Feedback Loop	5	4	4	5	3	3	3	3
	Easy to use	5	4	3	5	4	3	1	3
	No background knowledge	5	4	3	4	2	3	1	3
	Weighted Average		3.65	3.38	**4.27**		2.83	2.68	**3.42**

Tab. 4.1: User evaluation performed for the presented approach. 16 low-level requirements have been evaluated with a Likert scale from 1 to 5 in terms of importance by two experts (visualization and domain). High ratings in importance are highlighted in gray. Our approach has been evaluated against two known approaches by each of the experts. The results are color-coded in red (if our approach is rated worse than the known visualization tools), yellow (if our approach is rated similar to known visualization tools), and green (if our approach is rated better than the known tools).

As this project was developed in close collaboration with domain scientists in the biochemical domain, we continuously included the feedback of the users originating from an informal interview where we presented the current status of our research. The presented visualization approaches are the final version of this continuous review process, which was approved by the domain scientists.

To understand the improvements that the presented approach makes, the list of requirements that assisted in generating is used for evaluation. Here, the experts

were asked to rate the presented approach against the 16 requirements formulated by Gillmann et al. [79]. In addition, the experts were asked to rate two molecule visualization tools that they frequently use. The visualization expert chose PyMol and Protoshop, whereas the domain expert selected PyMol and VMD. Again, a Likert scale was used for the rating (1 meaning that the requirement is not fulfilled, 5 meaning that the requirement is absolutely fulfilled). The results can be found in Table 4.1. The color-coding shows in which categories the presented tool was not able to fulfill the requirements, such as the remaining tools (red), fulfill the requirements equally well as the remaining tools (yellow), or fulfill the requirements better than the remaining tools. Here, the presented tool shows a clear improvement.

First, the importance of the requirements has been used to create a weighted average to rate each tool. The visualization expert rated PyMol with a weighted average of 3.65, Protoshop with 3.38, and our approach with 4.27 points. Here, our tool outperforms both tools that were known to the user. On the other hand, the domain expert rated PyMol with 2.83, VMD with 2.68, and our tool with 3.42. Again, the presented tool outperforms the used techniques so far. When having a closer look into the ratings of the single requirements, it can be observed that the visualization expert rated seven requirements with the same points as the known tools. He rated the presented tool better than the standard tools for nine requirements. In addition, the domain expert found one requirement that we were not able to fulfill as well as the standard tools (compatibility). The expert justified this with the further need to promote the presented tool as an open-source tool such that it can be used in the biochemical community. For nine requirements, our tool was able to perform equally well as the standard tools. Further, 6 requirements are fulfilled better in our tool than by the tools chosen by our collaborators. This is the first indicator that our tool provides an overall improvement for the visualization of protein data under uncertainty.

In addition, our collaborators provided us with very motivating comments on the approach that we summarize below:

- "*I could use the application right away. It provides me with the most important visualization types that I need*"
- "*I like the easy-to-interpret visualization of the uncertainty in all views. It does not require a massive amount of time to learn them.*"
- "*I would like to encourage you to include this visualization style in an already existing visualization application for biochemical data.*"

We would like to take these comments as the first basis of a user evaluation. The last comment on the integration of the tool will form a basis for further development. Here, we aim to gather further user feedback when integrating our approaches into an already existing application.

4.8.2 Check of High-level Requirements

This work provides a UA interactive application for protein visualization. The system was designed according to the requirements we agreed on with our domain experts. This work provides a UA description of proteins that is able to represent anisotropic positional uncertainty. In the case where all dimensions hold the same quantification of uncertainty, the model degenerates into an isotropic model. Still, the distributions along one axis are assumed to be equal in both directions. Although this may result in simplifications of atom position distributions that are not equal in both directions, it allows us to propagate the described uncertainty along computational paths. We achieved this propagation for the computation of ϕ and ψ angles within an amino acid residue, revealing important information about a protein's structure. Although we did not use the propagation for further properties that can be computed based on the atoms of a protein, the general mathematical setup is not restrictive. Further measures such as curvature or quality of surface can be computed in a UA manner within the presented application (R1).

The described visualization procedures are integrated into visualization approaches actively used in the biochemical domain. We showed that commonly used visualization methods for proteins can be extended using our method. Although further representations exist, the presented application allows their inclusion and the mathematical setup does not restrict the underlying visualization approaches. Solvent-accessible surfaces and Ramachandran plots, as well as other geometric representations, are part of the presented visualization application that was designed flexibly so that further visualization schemes can be included if requested. We see this as a strong benefit, as biochemists are able to use the provided visualization methods without requiring a long training phase (R2). This is possible as the well-known protein representations, simple interaction modalities, and well-ordered controls create a familiar environment for users of the biochemical domain.

The uncertainty in most biochemical datasets results in a large number of potential atom configurations. A variety of approaches exist that aim to superimpose visual representations of these atom configurations, resulting in a large amount of visual clutter. In the presented application, we allow for an average visualization of all

potential atom configurations while outlining the potential spatial distortion of atoms. This massively reduces visual clutter while keeping the information of atom movement (R3). Due to validation reasons, we also enabled the application to show the generated uncertainty contour around all potential models.

An important aspect of a proper visualization approach for the biochemical domain is an interactive visualization application. Often, multiple views are required to understand the functionality of a protein in its entirety. Here, we provide a uncertainty-aware visual analytics (UAVA) application that allows biochemists to freely explore protein datasets that are affected by uncertainty (R4). During the development of the presented visualization approaches, we highly focused on achieving a minimal time-consuming computational process in order to avoid waiting times and allow real-time interaction.

4.8.3 Further Potential Applications

Our application could also be used to understand the interaction between proteins. In biochemical applications, the surfaces of proteins are compared in order to examine whether a ligand is able to bind to a protein. Here, our methodology could be beneficial in examining potential displacements in atom positions, resulting in possible binding sites.

Another application beyond chemistry could be ensemble visualization of geometries in general. This is an important issue in many applications, such as industrial manufacturing or path computations in simulations. Here, our visualization approach can be of great benefit in order to examine differences and common grounds of geometries in general.

At last, high-dimensional data analysis is an important topic. In high-dimensional data analysis, data points can be affected by uncertainty as well. The high-dimensional data points are usually simplified using dimension reduction applications in order to review these points. When doing this, the uncertainty of the original points propagates along with the performed computation and needs to be examined in the resulting reduced dataset. Our approach could be of benefit in indicating the uncertainty of the resulting points.

5 Images Measures as Uncertainty Predictors

Image measures can be used for uncertainty quantification by analyzing the statistical properties of images. By examining the distribution of image measures across many images, insights can be retrieved by analyzing them systematically. For example, if we observe that a particular image measure tends to be highly variable across different images, this may suggest that the measure is not a reliable predictor. On the other hand, if we observe that a measure is consistently low across many images, this may suggest that the measure is a good indicator of low uncertainty. Combining image measures by aggregation can create a more accurate and reliable uncertainty quantification method for various applications. Therefore, the following presents two applications using image measures for uncertainty prediction and similarity analysis.

First, a tumor detection scenario examines the use of image measures for uncertainty prediction. A framework for evaluating uncertainty-awareness in medical image analysis algorithms is proposed, and the effectiveness of image measures for finding regions of similar tumorous tissue is evaluated. Specifically, image measures are used to identify cases where an algorithm is uncertain about a particular classification, and this information can be used to improve the overall performance of the algorithm.

Then, the authors explore the use of image measures for uncertainty prediction in the context of cinema databases that allow the analysis of simulations, too large to watch on a regular computer, by compressing the scenes into 2D images. A method for exploring those simulation databases based on multi-dimensional image measures is proposed, demonstrating the effectiveness by visually identifying similar

This chapter has been partially published in the following articles:

- R. G. C. Maack, H. Hagen, and C. Gillmann. “Evaluation of the Uncertainty- awareness principle using a tumor detection scenario”. In: Proceedings of the VisGuides: 2nd Workshop on the Creation, Curation, Critique, and Conditioning of Principles and Guidelines in Visualization (IEEE Vis). 2018.
- R. G. C. Maack, D. Rogers, and C. Gillmann. “Exploring Cinema Databases using Multi-dimensional Image Measures”. In: LEVIA. OSF Preprints, 2020.

images and drastic changes. The image measures can be used to identify outliers in the database, which can help identify errors or anomalies in the data.

Both works utilize image measures to predict uncertainty in different domains. While medical image analysis and cinema databases are very different fields, both applications demonstrate the potential of using image measures for similarity analysis and uncertainty prediction.

5.1 Evaluation of Tumor Detection under Uncertainty

The acquisition process of real-world data is usually affected by uncertainty, which can significantly impact decision-making processes. The UA workflow tries to address this problem and gives guidelines to quantify, propagate, and communicate the uncertainty of a dataset. Although various works address this principle, the UA workflow is not evaluated thoroughly. This work selects a tumor detection scenario from the medical area, applying the uncertainty workflow. Based on this, we performed a user evaluation that tries to identify the quality of the UA workflow compared to a standard visualization approach. We summarize the pros and cons of the UA workflows according to different measures. Our study shows that the UA workflow helps users detect the boundary more accurately and understand the quality of the underlying image data.

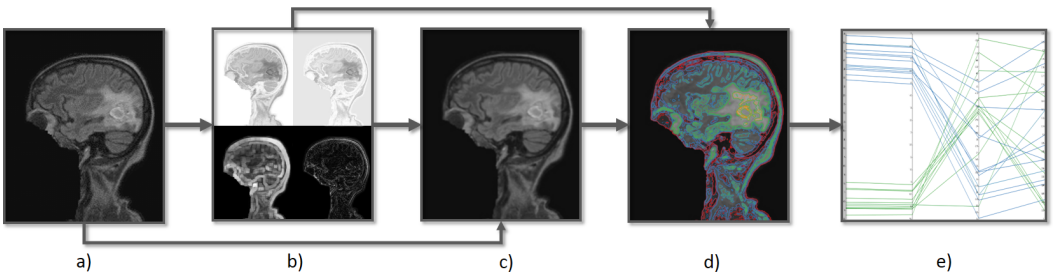


Fig. 5.1: UA pipeline used for an evaluation of the UA workflow. Using the original dataset (a), four different uncertainty measures (b) are being computed. The uncertainty measures and the original dataset are used to smooth the image with a UA Gaussian filter (c) to reduce noise. The smoothed image and the underlying uncertainty measures cluster the image, obtaining a weight for each pixel. For these weights, an iso-line visualization bordering regions with similar uncertainty behavior can be computed (d). In this visualization, the user can select multiple points to show the respective uncertainty values for each measure in a parallel coordinates plot (e), allowing intuitive exploration of the dataset.

5.1.1 Introduction

In various real-world applications, state-of-the-art image processing techniques are barely used [79]. This is caused by multiple factors, where the communication of uncertainty was identified as crucial [85]. For most of these applications, the considered data is affected by uncertainty, which means that measures may vary around the actual measured point [3]. This data uncertainty significantly impacts the utilized image processing techniques and can cause wrong or misleading results in a decision-making process.

Clinical image data, such as CT or MRI scans, are examples of real-world datasets containing uncertainty. Here, a model reconstructs the captured signals to image pixels, thereby introducing uncertainty [208]. Medical doctors must consider these image data in their daily clinical routine to determine the disease and select a suitable treatment.

Sacha et al. [203] formulated guidelines that need to be fulfilled to promote a UA visualization. In this paper, we aim to evaluate these criteria utilizing a dataset from the medical domain. This evaluation compares the slice-by-slice reviewing methodology used in clinical routine to a UA image processing approach. The approach comprises UA image processing operations, visualization, and interaction techniques. This allows a comparison of the UA workflow with a state-of-the-art visualization.

Therefore, this paper contributes:

- An evaluation of the uncertainty-awareness guidelines based on a real-world scenario
- A summary of advantages and disadvantages of the implementation using the UA workflow

The evaluation shows that the UA image processing approach helps users identify interesting areas in the dataset. The tool uses a parallel coordinates plot to let users select distinct points to examine their uncertainty properties. This helps to find out which areas are certain or uncertain.

5.1.2 Guideline Description

Uncertain data is defined as data that may contain values within some margin of error, which deviates the measured point from the intended or correct result [27].

The reason for this phenomenon is the data acquisition process. Typically, this process involves reconstructing sensor-captured values using a model.

This work evaluates the uncertainty-awareness criteria originally defined by Sacha et al. [203]. They proposed four requirements that need to be fulfilled to achieve uncertainty awareness in a visualization, namely:

- **C1** Quantify uncertainty in each point
- **C2** Visualize uncertainty information
- **C3** Enable interactive uncertainty propagation
- **C4** Propagate and aggregate uncertainty

Based on this work, Gillmann et al. [27] identified the apparent lack of implementations for these requirements in open-source solutions. Therefore, they proposed a novel data transformation pipeline aware of the uncertainty-awareness criteria C1-C4.

An input dataset must be extended by an uncertainty measure (C1) to achieve this. This extended dataset, namely U-dataset, is the novel input of an adapted data transformation pipeline. The U-Pipeline is necessary to transform the data values and alter the underlying uncertainty information. (C4). Throughout each data transformation step, the uncertainty needs to be visualized and interactively explorable by users while the uncertainty gets propagated to the next pipeline step (C2, C3).

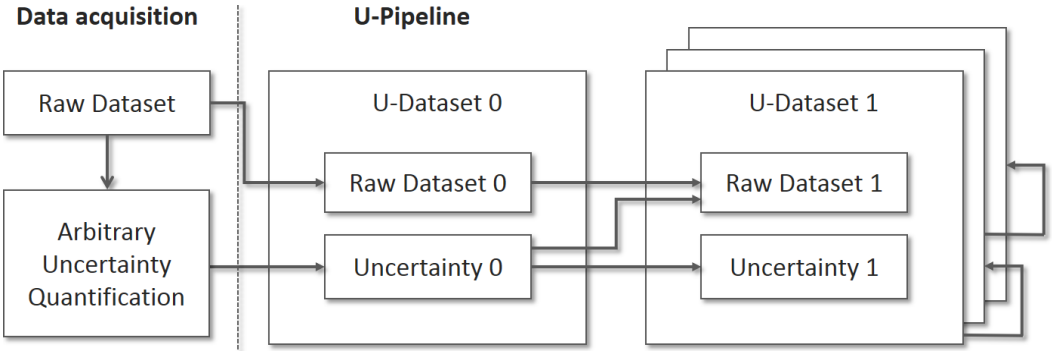


Fig. 5.2: Overview of the U-Pipeline used for the presented evaluation. At first, a raw dataset is utilized as an input. Using the raw dataset, the user can select an arbitrary amount of uncertainty measures to create an uncertainty quantification. Both the raw dataset and the uncertainty quantification are used in the U-Pipeline to apply an operation forming a new dataset. The uncertainty measures the outcome influence of the new dataset so that uncertain pixels are considered less.

Figure 5.2 shows the extended data transformation pipeline according to Gillmann et al. [80]. Here, UA image pre-processing operations such as histogram equalization, brightness thresholding, clearing areas, and detecting edges are presented. Each of these operations can transform a U-Dataset in its entirety. These operations can be applied arbitrarily, where the output of one operation can be used as an input in the next operation. It is also possible to add, subtract, and compare two datasets. This is enabled by the used pipeline, which uses the U-Pipeline to propagate and aggregate the uncertainty measures. This gives every step in the pipeline a distinct U-Dataset, allowing the user to visualize every intermediate step to gain a broad overview of the whole dataset.

5.1.3 Experiment

An evaluation was conducted to verify if users will accept and use the UA workflow and to determine if this principle improves the decision-making process. In this experiment, the U-Pipeline is utilized for a medical dataset. Then, the results are compared to a method used in daily clinical routines utilizing a questionnaire. This section presents the evaluation results and summarizes the pros and cons of implementing the UA workflow.

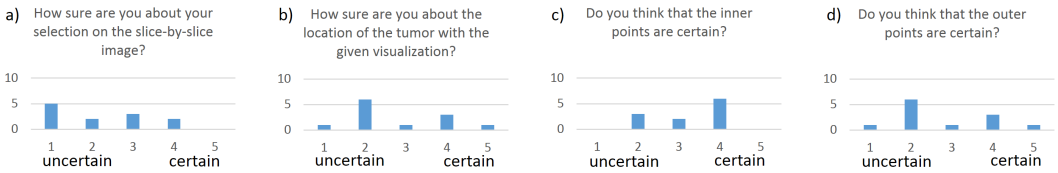


Fig. 5.3: Analysis of the questionnaire result. At first, the participants of the questionnaire were uncertain about the location of the tumor when solely considering the grayscale picture from Figure 5.1 a) is shown (a). For the proposed iso-line visualization in Figure 5.1 c), significant improvements can be achieved (b) where more people were certain about the location of the tumor. The parallel coordinates view, as seen in Figure 5.1 e) showed that the points on the interior of the tumor boundary are more certain (c) than the one lying outside the boundary (d).

Application Scenario

This experiment uses a human brain dataset with an internal tumor [207, 40]. The intention of using that dataset is that a clear border between the tumor and healthy tissue does not exist. In many cases, tumors do not separate clearly from their surrounding tissue. This increases the difficulty for doctors when determining the tumor's size, shape, and location [151]. The dataset also includes two zones of

tumorous tissue with a core segment and a surrounding bright area that should both be examined by the user.

As the tumor does not show clear borders, finding all areas and defining a separation between healthy and tumorous tissue is the main challenge in the presented example.

Experimental Setup

The presented dataset evaluates the UA workflow, defined by Sacha et al. [203]. We performed a study asking users to detect the tumor and its boundary.

We restrict the presented use case to the one slice of the given data set in Figure 5.1 a) to exclude interferences of the evaluation results by three-dimensional effects such as visual clutter. In this questionnaire, the user is asked to draw the border of the tumor in two different scenarios. First, users are requested to fulfill the task when provided with the slice-by-slice reviewing technique utilized in daily clinical routines. Second, users are asked to perform the same task while using a UA visualization approach from Figure 5.1 d) capable of including all uncertainty awareness requirements (C1-C4). The participants of the presented evaluation were asked to rate the ability of each visualization technique to indicate the tumor boundary. This directly compares the slice-by-slice visualization that is often used in medicine with the novel UA visualization.

To achieve a UA visualization, an iso-line representation (Figure 5.1d) was utilized. The method requires an uncertainty quantification of the input dataset. An arbitrary amount of uncertainty image measures output an uncertainty vector for each image pixel. In the presented case, the Gaussian error [67], the brightness measure [23], the local range error [229], and the salt and pepper noise [122] were used, as seen in Figure 5.1 b). These image measures were used to create a U-Dataset. This U-Dataset quantifies the uncertainty in each point, achieving the first criteria (C1) for UA visualization techniques. We applied a Gaussian filter in Figure 5.1 c) to the U-dataset to filter the contained noise in the dataset. Here, the underlying uncertainty was utilized to refine the Gaussian filter to consider uncertain data points less. This allows the propagation of the uncertainty quantification in the original dataset. While filtering the noise in the original dataset, the uncertainty vectors for each pixel also change, propagating and aggregating the uncertainty vectors for the given transformation (C4). This result is utilized to create the UA visualization presented by Gillmann et al. [78]. The technique utilizes multiple iso-lines, bordering regions with the same uncertainty behavior, dividing the dataset into similar areas while

showing the original slice in the background (C2). In addition, the technique allows visual exploration of the uncertainty space by enabling users to select specific image pixels and represent the respective uncertainty vectors in a parallel coordinate plot (C3).

Evaluation Results

We asked 12 persons to participate in our questionnaire-based evaluation. The results of this evaluation can be found below.

The age of the attended participants was an average of 27, and we had a ratio of two males to one female. Their working fields ranged from computer science and mechanical engineering to human biology.

At first, the participants were asked to draw a border around the tumor on a slice and to rate their certainty about their result. For most participants, it was not clear if they selected the correct tumor border or if they could capture all parts of the tumor. This resulted in a mean of 2.2 of 5 points utilizing the Likert scale. Figure 5.3 a) shows the exact distribution of values. The results show that defining the tumor border is not an easy task. This is an important feature for medical doctors to determine, highlighting its importance.

The iso-line representation from Figure 5.1 d) was presented to our participants to compare the slice-by-slice method to the UA visualization. When considering the UA visualization approach, the users were prompted to tell how certain they were about the tumor's location. The mean of the resulting scores given by the users is 3.1. This means an average improvement of 0.9 points to the slice-by-slice method. The exact distribution of the given scores can be seen in Figure 5.3 b). Since a Shapiro-Wilk test revealed that both data series are not normally distributed (p-value of 0.03528 (Figure 5.3a) and 0,01507 (Figure 5.3b), a t-test could not be conducted. However, the advantage of this visualization still becomes apparent when looking at the average improvement of 0.9 points.

In the last step of the questionnaire, points on the iso-line picture were selected, and their uncertainty measures were shown in a parallel coordinates plot. The points are depicted in such a way that they are located close to the isoline-indicated tumor border. Some points are located outside, and some are inside the border. Supported by the resulting parallel coordinate plot, users had to state if they think that the inner points (Figure 5.3 c) and the outer points (Figure 5.3 d) are certain. For the inner points, people found that they are mostly confident. On the other hand, the

outer points were seen as somewhat uncertain. Here, it becomes clear that the parallel coordinates view with explicit points can indicate an image point's exact uncertainty image measure values.

In conclusion, it can be seen that the proposed UA visualization supports the users to understand the dataset and its uncertainty better, as they can focus on areas with the same behavior under the selected uncertainty measures from Figure 5.1 b). For the parallel coordinates plot, users found that points inside the tumorous tissue are more certain than the ones outside this region. This information can help clinicians find areas that could be affected by the tumor but don't show a clear appearance of tumor tissue.

The proposed use case showed a significant improvement when using a UA visualization. The user study showed that the participants can perform more certain decision-making, which is also supported by the following very motivating statements:

- Areas were grouped, which are not correlated
- Faster decisions can be made
- Objective, absolute, and discrete information can be seen
- Similar regions can be seen better

In the medical field of application, tumors and other degenerated tissue often have fluent transitions to healthy tissue. By utilizing uncertainty visualizations, transition zones can be determined, and the probability of a data point being affiliated with one or the other tissue can be specified.

The utilized pipeline enables users to filter noise with different kernels, detect edges by various operators, and cut away nonessential information in an easy-to-use system. Each step in the pipeline can be used by simply applying an operation to one or two U-Datasets. The used functions in the pipeline can be extended as the need for new procedures arises, leaving the user unrestricted.

5.1.4 Discussion

Although a medical dataset is analyzed in this work, other data can be processed with the presented visualization by selecting and connecting different pre-processing steps. Most kernels and operations can be implemented and used in the pipeline,

creating various intermediate steps, which can be combined and compared, resulting in a processed image. This processed image can help to analyze the dataset faster.

Uncertainty image measures for uncertainty quantification can be selected arbitrarily to find the ones giving the most support for the specified task. To find the most helpful measure, this high-dimensional space can be visually inspected by looking at every measure individually. Measures directly related to the given use case can be neglected that way. Also, no uncertainty measures can be selected to adapt to circumstances where uncertainty propagation is unnecessary. That means that the underlying U-Dataset can be effectively created and used as desired by the user, as shown in Figure 5.2.

A dedicated infrastructure is required to use the tool on high-resolution datasets originating from new scanning devices. The used computers have to be capable of computing the required steps sufficiently quickly while also giving a smooth representation of the dataset. Since computer hardware is becoming cheaper, more compact, and more powerful, this issue will be automatically solved over time. In addition, data systems in the respective area need to be capable of saving the additional information generated throughout the UA pipeline (uncertainty-quantification, intermediate computational steps, selections, etc.).

Another aspect of the needed infrastructure is that people using the software must learn how to use it. Users need time to adapt to modern medicine's new software and techniques. Here, the educational system needs to be adapted. Novel approaches to teaching people these new skills exist. One of them is being suggested by Gillmann et al. [87], presenting a concept lecture aiming to impart image processing and visualization principles to students in medicine.

To determine if the tool is powerful and can be used in day-to-day operations, the criteria for successful visualizations defined by Gillmann et al. [79] are being tested.

For the tool's usability, a low degree of difficulty in using the visualization and data pipeline is achieved through easy data manipulation. Here, the user has to select an operation and a dataset to create the following dataset in the pipeline. For any of these datasets, the iso-surface visualization can be created.

For tool efficiency, namely the consumption of resources, main memory consumption is one of the most significant issues. An arbitrary amount of uncertainty measures and steps in the pipeline must be created for every raw data volume. This should be kept in mind when buying dedicated hardware. However, since hardware is

also becoming cheaper and more powerful over time, this will be less important for future applications.

The Correctness of a program shows how accurate a visual representation is, while the communication of uncertainty is also required for a good visualization. For the used tool, the communication of uncertainty has been shown before. The iso-surface visualization itself uses this uncertainty information, creating a good representation of the underlying information. For the parallel coordinates view, it becomes clear that it shows the exact values of the underlying uncertainty measures.

Flexibility means that different tasks of an application can be executed. This is not solely valid for medical image data but for every image that can be loaded as a volume. Any volume can be analyzed and processed since the user can select the most useful uncertainty measurements for the given application. The pipeline is also completely flexible, as any operation can be applied anytime.

The last criterion to be discussed is the tool's intuitiveness, which declares that the user does not need more profound background knowledge about the program's procedures. This criterion is fulfilled since the utilized tool enables the user to play around with the loaded volume to find the best visual representation. There is a specific way of representing uncertainty measures, and each pipeline step can be visualized to show its current state.

In the given experiment, only medical image data were examined to investigate if improvements can be made in the daily clinical routine using UA visualizations. Given the error introduced by data acquisition, these techniques can also be applied in other application fields. Every image reconstructed from the real world contains uncertainty, so field experts can use uncertainty information to improve decision-making. Natural disaster forecasts could be made with plans for every scenario, material testing could show points of failure more confidently, and the certainty of simulation data could be displayed.

5.1.5 Implications and Considerations

Our evaluation showed that including uncertainty information is crucial in gaining user acceptance for novel visualization approaches. We showed that the implementation of all four uncertainty criteria provides a significant improvement in the decision-making process. However, requiring a suitable infrastructure and teaching effort, utilizing a UA visualization is beneficial.

The evaluation of the UA workflow also showed that all four defined requirements for uncertainty awareness are required and sufficient to achieve a UA visualization.

When implementing the UA workflows to further application scenarios, the following guidelines need to be kept in mind:

- An uncertainty quantification of the utilized dataset needs to be found (e.g. tree datasets, high-dimensional data, time-dependent data...)
- A UA visualization that can address the defined requirements needs to be found
- A suitable data management and teaching infrastructure needs to be employed

5.2 Image Filtering Using Uncertainty Measures in Cinema Database

Exa-scale simulations can be challenging to analyze because storing all computed timesteps and other parameters is nearly impossible. The Cinema Database provides a storage-saving solution that captures images of each simulation timestep from various camera angles. Still, the resulting number of images can be overwhelming, and it is hard to find interesting images and features for further analysis. Here, a zoom-based approach where users can utilize arbitrary image measures to explore interesting images and further analyze their behavior in detail. In this chapter, we show the effectiveness of our approach by providing two real-world cinema datasets.

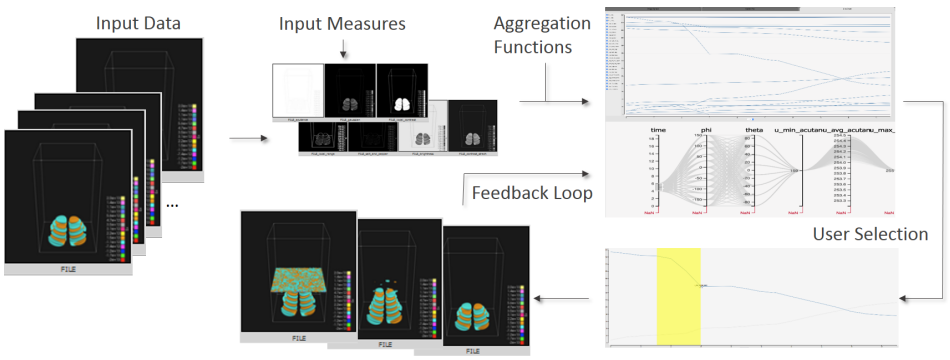


Fig. 5.4: Zoom-based Querying in Cinema Database. Input images, input measures, and aggregation functions form the input of the presented approach. Users can select interesting timesteps where image features change dramatically. The selected images can be reviewed and form a feedback loop for users to create new selections.

5.2.1 State of the Art and Open Problems

Large image databases recently have been used more frequently, either as commercial tools such as Bynder [29] or pixx.io [187] as well as in research such as the Cinema Database [51, 2, 240] and interactive websites [89]. Their general purpose is to allow users to browse, select, and analyze large collections of images. Cinema, in particular, aims to assist in analyzing exa-scale simulations, with multiple timesteps that are too big to be stored on a computer or cluster. It consists of multiple projects that can create Cinema Databases from simulations, alter and add additional information to Cinema Databases, visualize those databases in viewers, and provide building blocks to build up a new viewer.

Here, Cinema provides a variety of images per timestep that can be stored and reviewed. This enables the user to watch the simulation as often as needed to analyze problems and find interesting results while not requiring a lot of computing power or massive storage solutions. Although Cinema Database assists users in storing, managing, and browsing their image collection, it lacks the ability to provide an overview of the image collection, an entry point for image searches, or a search function via interesting attributes. As one can imagine, not every image in the Cinema Database provides the same amount of information. Usually, timesteps are sampled equidistantly in each direction, leading to a variety of images that can show equivalent or irrelevant information. When considering these image collections, users face the problem of finding interesting images or overseeing interesting ones for their analysis, as shown in Section 5.2.2.

5.2.2 Related Work

This Section summarizes the state of the art for the visualization and exploration of large image databases.

Visualization of Large Image Databases

As Camargo et al.[32] stated, most image databases aim to map a high-dimensional feature space into a two-dimensional map. This statement highlights the ability to represent the proximity of images in a more visual manner. A variety of approaches are available, namely Multidimensional Scaling [254], Principal Component Analysis [202], Self-Organizing Maps [175], Curvilinear Component Analysis [149],

Laplacian Eigenmaps [226], Shannon's Mapping [205], Stochastic Neighbor Embedding [115] or Kernel Isomaps [37].

Although these approaches allow users to gain a first overview of the considered image database, they cannot guide users to interesting images that contain most of the information. The presented approach allows users to select different features of images and identify images that match these features the most.

Exploration of Large Image Databases

There has been a large number of works addressing the navigation of large image collections. The most important issues are discussed below.

Deng et al. [154] presented a similarity-based visualization of large image collections that should allow users to identify images that contain similar content. Although this allows users to identify similar images, it does not provide an entry point where users are guided to interesting images. In contrast, we provide a measure-based entry point where users can identify interesting images and further examine them.

Further approaches utilize hierarchical visualizations of images to allow users to identify correlations between the reviewed images [12, 4, 53]. Here, images are ordered in terms of their hierarchical correlation, which can be computed with different measures. Although this provides important information about the hierarchy within an image database, users need to be assisted in finding the most significant images. In contrast to the presented approaches, we provide a zoom-based approach where users can select interesting images and allow further examination.

Image databases can additionally be browsed and sorted according to metadata information, such as segmentation information [159], machine-learning-based content examination [231, 11], Thumbnails [25] and image locations and time [90, 238]. Here, users can search for interesting images as a starting point. Still, the meta-data must be computed, which can be computationally expensive. In the presented system, we provide a measure-based approach, where users can insert and pre-compute arbitrary measures to query interesting images in the considered image database.

5.2.3 Contribution

To solve the problem of "finding interesting images" or "overseeing important keyframes", a novel visual guidance approach is used to explore large image

databases. A visual workflow embedded in the Cinema Database is used where users can determine arbitrary image metrics that capture features of interest. In most cases, multiple features are of interest to the user. Determining their importance is part of the exploration process. Multiple image measures are implemented to address this issue, providing users with a visual tool to explore these metrics, apply aggregation functions, and explore the resulting selection based on the user settings. When focusing on selected images, a detail view of the selected features within the images is available.

This application contributes:

- Requirements for intuitive exploration of large image databases
- A context view of feature distribution within a Cinema Database
- A detail view of feature distribution within images

We show the effectiveness of the presented approach by applying it to two different Cinema Databases (see Section 5.2.7).

5.2.4 Requirements Analysis

To develop an interactive framework where users can find and examine interesting images in an extensive image collection, we formulate requirements based on the conducted state-of-the-art analysis that needs to be fulfilled.

R1: Feature Selection

The selection of images in a database highly depends on the user's goal. Here, users must be enabled to determine features they are interested in and utilize these feature measures to select interesting images.

R2: Overview of Feature Distribution

Besides the ability to freely decide on image features, users need to be able to browse the overall distribution of a feature in the considered image collection. Solely when reviewing the overall distribution of a feature in a set of images, they can recognize important images where specific features are dominant.

R3: Detail View of Interesting Images

Although the selected images may represent a suitable set of images that contain the desired feature, users still need to be able to review single images and explore how features are distributed in these images.

R4: Low Computational and Usage Costs

As the Cinema Database is designed to decrease the computational costs of reviewing simulation results, a navigation approach should not increase the complexity of the Cinema approach [79].

5.2.5 Methods

This section describes the visual guidance presented in Cinema Databases using multi-dimensional image measures. An overview of the workflow we present is shown in Figure 5.4. From the input dataset and the selected image measures, aggregation functions are used to guide users to interesting images in the database interactively. Following that, users can further analyze selected images in detail by reviewing all selected image features. The single steps will be shown below.

Input and Preprocessing

Input datasets that fit the Cinema Database scheme can be very versatile. The specification (spec D[2]) requires a CSV file containing all images and additional information that is contained in the database. It can state additional metadata, such as the angle positions of the camera, timestep information, and even additional data files.

In addition to the input Cinema Database, the presented approach requires one or multiple image measures representing the features of interest in the given database. These features can range from image quality measures [7] over uncertainty measures [78] to entropy measures [35]. Computing image measures can be time-consuming, especially when considering large image databases. As these computational outputs remain the same, we pre-compute the selected image measures and store them beside the actual images. When starting the presented system, the

[2]https://github.com/cinemascience/cinema/tree/master/specs/dietrich

computed measures can be directly loaded without additional computational time needed. As this adds a gray-scale image per measure and original image, theoretically, only the aggregates of those measures could be saved and safe on disk space.

The image measures used vary in their goals but are all local to a pixel rather than the whole image. All the measures used will be briefly described to give a short overview. Acutance [6] is an image measure that considers pixels on an edge as certain, resulting in a low output value for those pixels. Contrast Stretch [188], on the other hand, aims to have an equal distribution of intensity values, giving pixels with very low or very high occurrence rates a high value. Gaussian Noise [67] is very similar but uses a Gaussian distribution instead. Brightness [23] is a simplistic measure that considers dark pixels as certain and gives them a low value. Local Contrast [124] views pixels, which are often used as trustworthy, resulting in a low value. Local Range [229] uses the highest absolute distance to its neighbors as the output. Finally, Salt and Pepper [122] uses the average distance to the neighboring pixels for its output.

Usually, image measures work on a local level, which means that they output a result for each pixel contained in an image. As we are interested in the overall rating of an image to find interesting ones, we need to aggregate the single outputs of each pixel in an image. Here, the presented system utilizes different aggregation functions, such as minimum, maximum, or average, to create an overall impression of single images. An overview of different aggregation functions can be seen in the work of Cai et al. [31].

In summary, the image collection, feature measures, and aggregation functions form the input of the presented system.

Feature Aggregation Visualization

Based on the pre-processed input of the presented system, we built an interactive visualization where users can search for interesting images in the database and further examine them. Here, an intuitive visualization is provided to allow users to find interesting images along different computational timesteps of a simulation. The presented visualization consists of multiple components. First, a line chart that represents the aggregated feature functions across the considered timesteps or other dimensions. Second, a parallel coordinates view that represents the distribution of aggregated features throughout the captured images. Third, an image viewer shows

each image with its corresponding image measure images. The first two views can be seen in Figure 5.5.

Figure 5.5 a) shows the line chart of the aggregated feature functions. The x-axis presents the selected dimension. Here, the timesteps are shown, and the y-axis shows the aggregated feature values. Users can enable and disable all feature functions and search for interesting trends in one or multiple features. When detecting interesting feature trends at specific timesteps, users can select these timesteps by dragging along the x-axis, and the corresponding images will be displayed, making them exploreable.

Figure 5.5 b) represents an example of the parallel coordinate view. Here, axes like the time and angles of the captured simulation runs and the computed aggregation functions are displayed. Users can manipulate the view by sorting the axis as well as selecting interesting feature accumulations or timesteps. The selected images will be displayed according to the user selection.

Both views allow users to examine the image database according to the defined image feature functions and the selected aggregation functions. The selections made throughout this interaction process will be displayed to the user and can be further examined in a detail view.

Image Features Visualization

After selecting interesting images based on the aggregated measures, users are enabled to review the selected images and their features. Contrary to the aggregated views, we aim to allow users to examine the features of the selected images.

Here, all computed features are presented side by side. Users can review each output of the computed measures to determine areas in the image that contributed the most to the aggregation function output. Although not available yet, a further visualization approach that helps users review all measure outputs at once can be included in the presented system by utilizing an embedded isosurface visualization [77].

5.2.6 Implementation

For this project to be implemented, three repositories had to be altered. First, Cinema Lib [51] is used to apply image measure functions, as well as aggregate functions to the images and save them to the database. Second, the Line Chart component is added to Cinema Components to be embedded into Cinema Explorer.

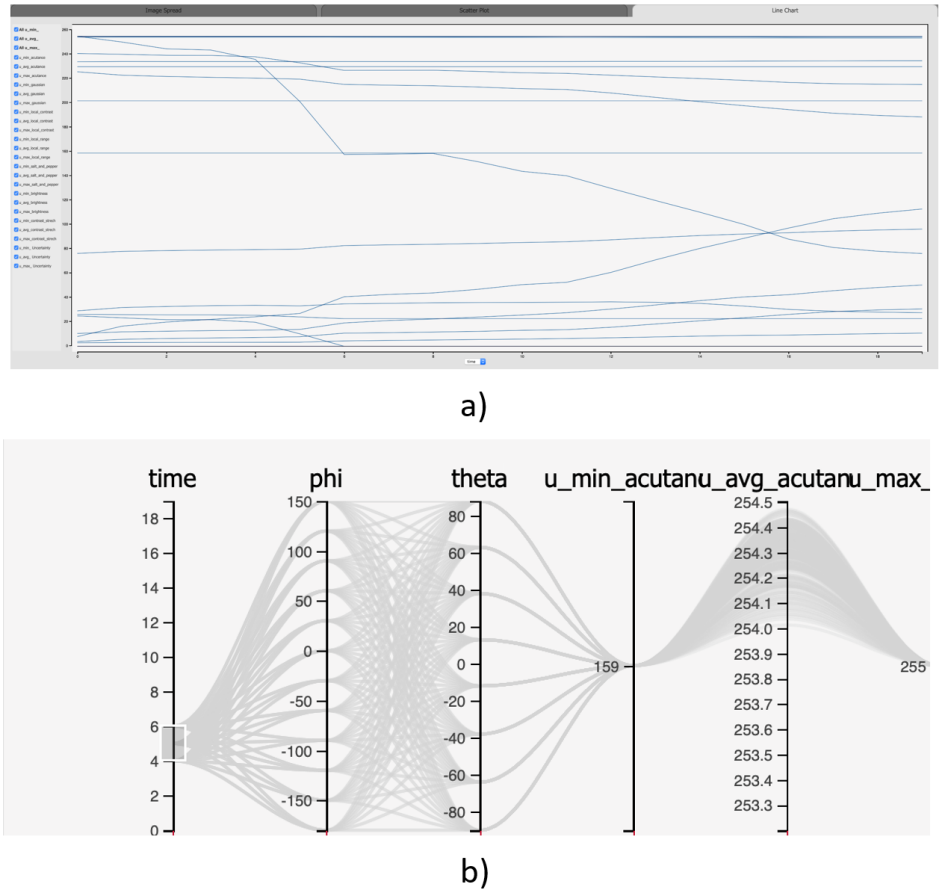


Fig. 5.5: Feature aggregation visualization. a) Line plot with the x-axis showing the timestep and the y-axis representing several aggregated feature values. b) Parallel coordinates plot with aggregated feature values, time, and image angles.

Cinema Lib

Cinema Lib is a command-line tool written in Python that can alter and check cinema databases. In this case, seven image measures, shown in Section 5.2.5, were implemented next to the existing image measures. These measures capture different properties of the images. They can be applied using a simple command that can be found in the Cinema Lib documentation.

Aggregate functions like min, max, and average are automatically applied to these resulting measure images, giving them a single value per timestep to be displayed.

Expanding the functions in the Cinema Lib project can add other image measures and aggregate functions. The functions for applying image measures to Cinema Databases already exist, so own implementations of image measures just have to

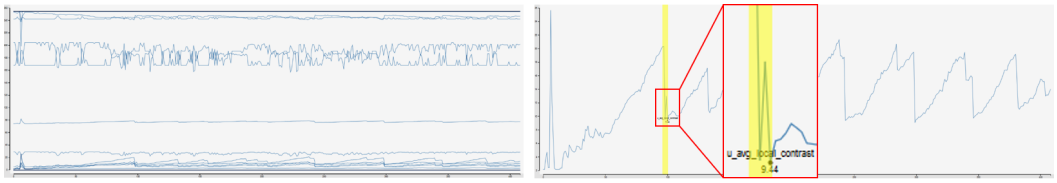


Fig. 5.6: A snapshot of the Line Chart with the loaded Bubble Bed simulation. Figure 5.6 a) shows all the available aggregate functions. In Figure 5.6 b) a very characteristic function was selected, zooming into the range of that function. The yellow bar shows a selection of the user, which is viewed in detail in Figure 5.7. The zoomed part of the Line Chart shows a second rise and drop of the line after the huge drop.

be made compatible with these functions. Thus, it is relatively easy to add new measurements to the system. The same holds for aggregate functions.

Cinema Components

Cinema Components is a collection of tools and visual components written in JavaScript. They are designed such that they can easily be embedded into any HTML viewer. We added a Line Chart component and interaction methods for other components to this repository, enabling the user to explore the computed aggregate functions from the altered Cinema Database. The Line Chart component consists of the chart and various user interface functions to create an easy-to-embed component.

Cinema Explorer

Cinema Explorer is an advanced viewer that allows for exploring cinema databases. It consists of a Parallel Coordinates view at the top and a tab container at the bottom. The tab container has three tabs: an image viewer, a scatter plot, and the new line chart.

The Parallel Coordinates (Figure 5.5 b) view gives the user an overview of the dataset, enabling the user to view parts of the data by selecting a range for each dimension.

The image viewer shows all images corresponding to an entry in the dataset, giving the user a detailed view of the dataset. Here, all the image measures can be seen next to the original image, allowing the user to view all the dataset details together with the properties of each entry.

Scatter plots can be used to plot two dimensions against each other to explore the interdependence of two dimensions.

The Line Chart shows the computed aggregate functions averaged over the entries for the selected x-axis. For example, if the timesteps are selected, all those entries with the same timestep are averaged in regard to their aggregate function and then displayed. A resulting line might show that a certain measure is rising or falling on average, depending on the viewed measure. This can mean that the number of edges, homogeneous regions, or overall brightness of the images is rising or falling. The chart is embedded so that selections are also applied to the Image Viewer and the Parallel Coordinates view, granting the user easy control over the whole tool. Therefore, a part of the x-axis can be selected on the Line Chart to be viewed in detail in the Image Viewer. This gives the user a zoom exploration experience.

General Workflow

The general workflow can be summarized into a few steps:

1. Convert a Cinema Database to support aggregated measurements of any kind using the Cinema Lib project.
2. Add the column names from the CSV file of the aggregated measurements to the Cinema Explorer database file using the image_measures entry.
3. Use the Line Chart checkboxes and filter for interesting-looking or strongly changing lines, zooming in on them.
4. Select a part of the x-axis with interesting line changes by dragging the mouse with a left click.
5. Explore the causes of those changes by examining the detailed information from the Image Viewer.
6. Repeat steps 3-5 or 4-5 to fully explore the important steps in the dataset.

5.2.7 Results and Discussion

Now, two successful Cinema Databases are reviewed by the presented workflow. Additionally, the presented approach will be compared against the defined requirements.

Results

In order to prove the applicability of the presented system to real-world datasets, this chapter provides two use cases that will be discussed in the following. For both use cases, we provide the aggregation functions minimum, maximum, and average as well as seven different image measures, discussed in Section 5.2.5.

The first example is a series of timesteps comparing the evolution of isosurfaces based on topological analysis optimized to find the most significant isosurfaces versus regularly spaced isosurface values. The images are acquired by the WarpX system [236], which is an advanced particle accelerator simulation code that models plasma-based accelerators. The image database contains 1920 images in total. Scientists are interested in finding interesting timesteps where novel and large iso-surfaces occur, meaning that they are interested in big changes in the iso-surfaces. We use this example to show the resulting workflow of the presented approach, as can be seen in Figure 5.4.

When applying our algorithm, the user can review both aggregation function visualizations. In the line aggregation view, most values appear to be stable independently from the reviewed timestep. Contrary to this, maximum and minimum contrast stretch are increasing (or decreasing) when considering different timesteps. Here, users can select the most significant "jump" in the aggregation function. When reviewing the resulting images, it can be seen that the images change dramatically. The shown iso-lines grow massively until they finally build a novel large iso-surface. This example shows, that the presented system helps users identify interesting timesteps and review the content of these images.

Bubble Bed Simulation The second example is a small, dense bubbling bed simulation result with over 3M particles. The physical dimensions of the domain are about the size of a business card. This simulation contains 408 timesteps and 3 Million particles and displays the formation and evolution of bubbles expected in a full-looping reactor. The data was generated by the MFIX-Exa system, which is a multiphase flow simulation code based on the MFIX code. Detailed information on the MFIX simulation can be found on the National Energy Technology Laboratory (NETL) website [234].

To analyze this dataset, it is important to find timesteps that mark important steps in the simulation. Therefore, seeing how much area is covered by bubbles at a time is a good insight for scientists.

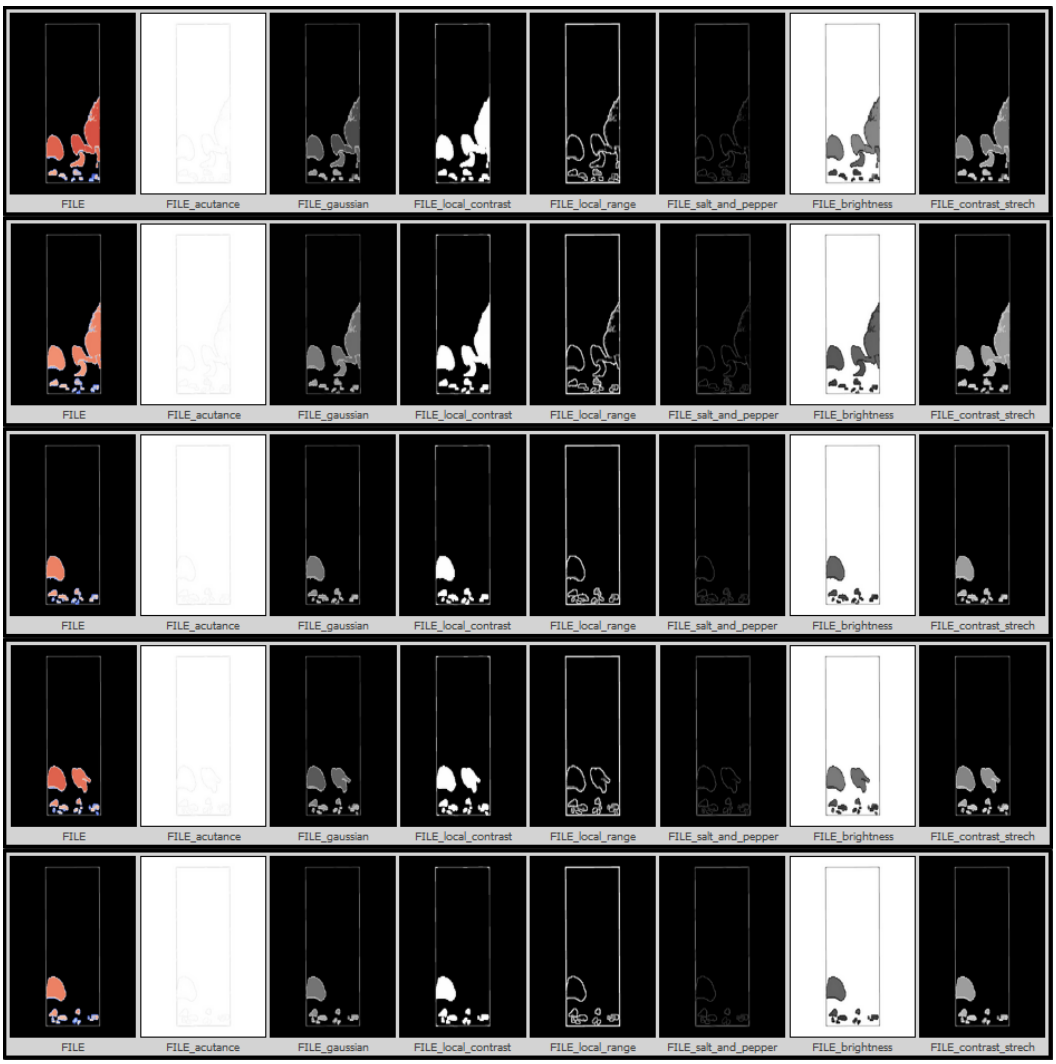


Fig. 5.7: A snapshot of the Image Viewer tab with the loaded Bubble Bed simulation showing the images selected in Figure 5.6b. On the left, the original images, each with 7 image metrics next to it, can be seen. Each row corresponds to one timestep. The sharp drop with the smaller spike afterward in the selected region corresponds to a big bursting bubble followed by a smaller bubble building up and bursting again.

Figure 5.6 a) shows all the available aggregate functions and considered image measures. Now, very characteristic lines at the bottom of the chart show repetitive behavior. When the topmost line is selected, as seen in Figure 5.6 b), a line building up slowly and dropping very fast can be seen. Therefore a part of the x-axis is selected (yellow) and is viewed in the Image Viewer tab in Figure 5.7. When considering the selected images, the cause for the sharp drops can be found in the images, as these show how big bubbles burst and grow over time. Bubbles building up and bursting according to the selected aggregated measurement can be seen.

Now the scientist can characterize the dataset into growing stages and bursting stages by utilizing the selected image measure.

Discussion

This section will discuss the requirements of Section 5.2.4 and the benefits and drawbacks of the presented method.

First, it is designed to allow a variety of image measures for the selection of interesting images. Seven different image measures are newly available to users together with the already existing one of Cinema Database. Additionally, the code is designed to be flexible to allow users to further implement their own image measures (R1).

In order to provide a context visualization of all images contained in a Cinema Database, we allow the use of different aggregation functions and represent the results with two different line views. In the current system, three different aggregation functions are implemented. Again, users can implement further aggregation functions due to the flexible code design (R2).

After deciding on interesting features, users are enabled to review selected images of the database. The images and computed measures can be reviewed for every measure utilized for the selection process. This forms a feedback loop as users can check which part of the image contribution forms the outcome of the image measure (R3).

The presented system allows an arbitrary amount of image measures to be added to classify images. Although these can be computationally expensive, the runtime of the presented workflow is not affected as the image measures are computed in a pre-processing step and loaded when required (R4). Also, the task of computing the image measures can be parallelized per image or even per pixel.

Overall, the presented system is capable of fulfilling the defined requirements.

5.3 Perspective on the Uncertainty-aware Visual Analytics Process

This section maps both applications to the UAVA process described in a post hoc analysis. Here, the design process is depicted for the creation of similar applications. The design can exhibit additional features not present in the original design to

fully explore the possibilities of the UAVA workflow and as an outlook for future applications. As those applications were developed before the creation of a UAVA workflow, the original design misses some described features that can be seen as recommendations for future work.

5.3.1 Turmor Detection

In the described scenario of tumor detection, radiologists are interested in extracting areas of tumorous tissue from CT scans. After the CT images are taken from a patient, the radiologists usually view them to identify tumors, as they appear glowing. Still, the extent of the tumors is not apparent right away. The slice-by-slice visualizations make this process worse, as the correlation between slices is often lost.

Therefore, radiologists want to create a UA workflow that allows them to extract tumorous tissue more efficiently throughout the slices, still keeping the slice-by-slice viewing method they are used to. The CT image data is the starting point for every analysis that can be preprocessed with blurring or sharpening filters to filter noise or enhance edges and transitions between tissue types. From here on, contours can be created to catch similar areas, leading to many uncorrelated areas. Hypothesis tests often depend on the prior knowledge of the radiologist to extract all tumorous areas. Those areas are annotated, improving the dataset. This annotated dataset can be enhanced in further iterations to detect more tumorous tissue.

There are several sources of uncertainty when capturing CT images, including image noise, patient motion, beam hardening, and partial volume effects. Image noise is caused by the random nature of the X-ray photons that penetrate the patient's body and can lead to a loss of image detail. Patient motion can occur due to breathing or other involuntary movements, resulting in blurred images. Beam hardening occurs when the X-ray beam passes through different materials in the body, leading to artifacts and inaccuracies in the image. Partial volume effects happen when a voxel contains more than one type of tissue, leading to uncertainty about the actual density and composition of the tissue. All these sources of uncertainty can affect the accuracy and reliability of CT images, which can have significant implications for patient diagnosis and treatment. As most uncertainty sources are hard to evaluate or unavailable, the researchers use image metrics to model the uncertainty in the capturing process.

At first, the Gaussian blur and sharpening filters are adapted to allow preprocessing of the images, considering uncertain pixels less. These preprocessing steps can diminish

noise or enhance features to ease further processing. The enhanced images are visualized using multiple contours, where the space of image measures is aggregated to utilize them as a similarity metric. This enables the radiologist to remember areas of similar behavior by highlighting them in the same color throughout all slices. By visually exploring the contours on various slices of the image, radiologists can detect tumorous regions, receiving an uncertainty quantification from the underlying uncertainty space of a selected region. Areas of unclear behavior can be explored in detail using a parallel coordinates plot to explore the uncertainty space.

When selecting an area, the physician can annotate it, also saving the underlying uncertainty space with that annotation. This gives the insights an explicit uncertainty quantification. This annotation can also be seen as the sole opinion of a radiologist, being transferred to the Dataset component, where an uncertainty quantification could be created using another set of image metrics.

Throughout the process, radiologists can express their degree of (un)certainty in each decision. Tracking those expressions of uncertainty with all annotations can be viewed in a line chart that shows a cumulative product through time. The radiologists can edit unsure annotations and review the areas they see as real tumorous regions with high certainty.

5.3.2 Cinema Databases

In the case of cinema databases, compressed simulation data, created by capturing 2D images from various perspectives per timeframe, is to be analyzed. The researchers want to find interesting time steps in the plethora of images present in each dataset. This drives the need for a visual exploration tool. Images can be filtered to show the same perspective for the present timesteps to allow scrolling through the dataset. The images are shown together with the timestep and perspective information (in camera angles), allowing users to zoom into intriguing images on demand. When finding such important timesteps, researchers can view nearby timesteps and camera angles to find anomalies or features of interest. This may lead to a change in the simulation to enhance the reported issues, resulting in a new dataset to be analyzed.

Uncertainty in simulations can be introduced in multiple ways. Variations in observation can be present by simulating with slightly varying starting conditions. Choosing the set of starting conditions is another source of uncertainty, as it can lead to Non-representative sampling of the parameter space. Still, uncertainty quantification has

to be conducted on the simulation side for meaningful results. Therefore, to allow the analysis of arbitrary sets of images, the researchers estimate the uncertainty using image metrics. This step may be skipped when uncertainty quantification is given directly through the capturing process.

The uncertainty quantification can be directly displayed next to the images, providing details about extracted features. Also, aggregation functions summarize the uncertainty per image and image measure, displayed as a line chart, enabling to find interesting time steps that might exhibit important keyframes in the simulation, improving query processes in a focused manner.

The image measures can also be used to quantify the insight uncertainty, where selecting the right measures is important. The researchers can annotate their findings and express their conclusions in the form of text. This annotation process can also be used when ignoring the uncertainty information completely, such that insights created without quantification are also rated uncertainty-wise.

Finally, provenance creation and tracking can be implemented in multiple ways, depending on the available uncertainty information. When including simulation software in the process, the uncertainty of each measure or the aggregation functions can be tracked, allowing the researchers to understand how key features change throughout iterations. Also, annotations tracked while improving the simulations can be displayed in a calendar-style visualization to give the researchers an overview of their findings, allowing them to find loopholes in their development or highlight reoccurring issues.

5.4 Conclusion

In conclusion, the presented evaluation of a uncertainty-aware (UA) data transformation pipeline and the flexible zoom-based querying system for image databases demonstrate the potential of using advanced visualization and querying techniques in medical and scientific fields. The evaluation results showed that embedding UA data transformation pipelines and visualizations into the daily clinical routine is possible nowadays and can help medical doctors examine data faster and create better treatment plans. The effectiveness of the zoom-based querying system was demonstrated with two real-world scenarios. Image measures are reliable estimators of uncertainty and similarity for image data. Those image measures built the basis for the following analysis in both applications, where they can significantly improve data analysis and decision-making processes in various fields.

6 Topological Data Analysis and Potential Use-cases for Uncertainty

Topological data analysis (TDA) provides a family of effective feature characterizations, including the well-studied Morse-Smale (MS) complex. The MS complex is a central tool in TDA for feature-driven data analysis and visualization, as it segments the domain of a scalar field into regions with equivalent gradient flow behavior (see Figure 6.1 and Section 6.2). This rather abstract feature characterization based on gradient flow has been applied successfully in several domains, including chemistry [178, 16], material science [237, 117], physics [147, 106], and cosmology [220].

However, due to the high computational complexity of MS complex generation, it often becomes a time-consuming bottleneck. Especially in the case of interactive analysis and visualization, users expect to quickly retrieve a visual output, where long wait times can interrupt their workflow. Furthermore, many applications do not require the computation of the full MS complex but only two of its central features: 1) the MS segmentation of the domain that assigns extrema to each vertex by following the gradient along the steepest descend and ascend; and 2) the interfaces between different regions in the segmentation.

In this paper, we describe a scalable implementation of these two tasks with high parallel efficiency, further referred to as the piecewise linear Morse-Smale segmentation (PLMSS) algorithm (Section 6.3). As the name suggests, the input of PLMSS is a scalar field defined on the vertices of a piecewise linear domain, i.e., a simplicial complex. PLMSS utilizes path compression to derive the MS segmentation of the domain and a multi-label marching tetrahedra procedure to derive the interfaces between different regions of the MS segmentation. We chose these two underlying

This chapter has been partially published in the following article:

- R. G. C. Maack, J. Lukasczyk, J. Tierny, H. Hagen, R. Maciejewski, and C. Garth. “Parallel Computation of Piecewise Linear Morse-Smale Segmentations”. In: IEEE Transactions on Visualization and Computer Graphics, 2023.

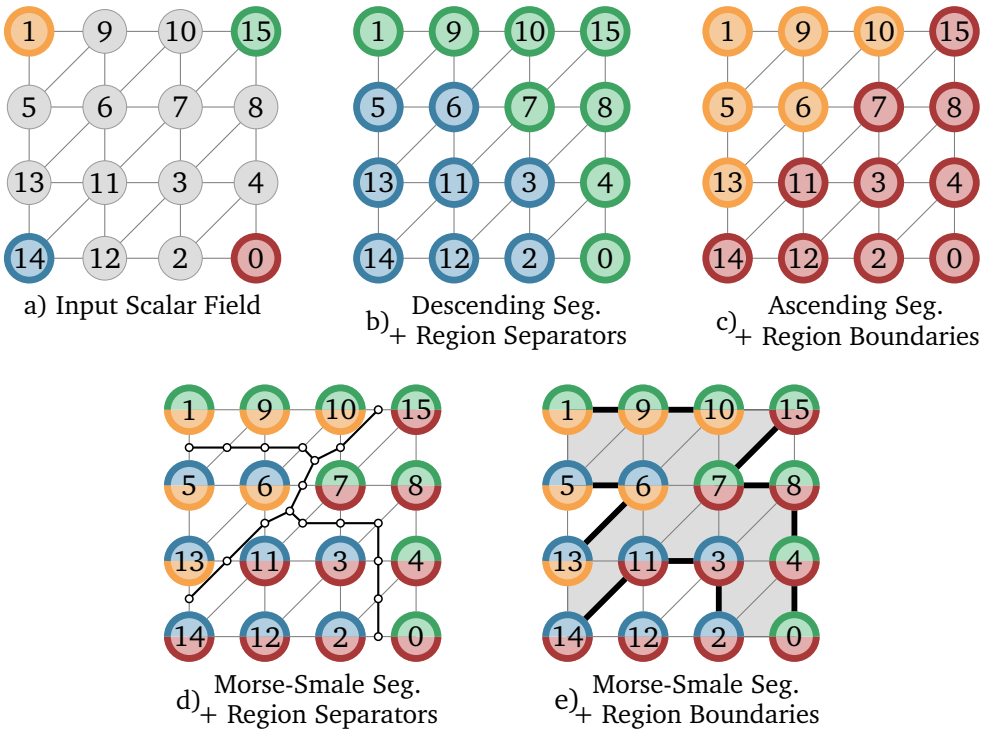


Fig. 6.1: A 4×4 example triangulation showing the input field (a) and available segmentations and region-separating geometries (b-e). Each color represents the influence area of a specific extremum (the areas of minima 0 and 1 are shown in red and yellow, while the areas of maxima 14 and 15 are shown in blue and green). The descending segmentation (b) represents the influence area of maxima and the ascending segmentation (c) the ones of all minima. In the case of MS segmentations (d-e), the nodes show the influence of minima-maxima combinations, leading to nodes being colored by the minima color at the bottom and the maximum color at the top. Subfigures (b) and (d) show region separators (thin black lines), while (c) and (e) show region boundaries (thick black lines).

algorithms since they are known to be embarrassingly parallelizable and, therefore, expected to scale well. In Section 6.4, we demonstrate the benefit of using PLMSS for effective data analysis and visualization on four datasets.

We compare PLMSS against the corresponding subprocedures of two state-of-the-art MS complex software libraries, i.e., the implementations available in the Topology ToolKit (TTK) [18] and MSCEER [101]. Specifically, we performed two strong scaling studies (Section 6.5) that show that MSCEER outperforms TTK, but PLMSS is still up to an order of magnitude faster than MSCEER while also providing superior parallel efficiency. However, the qualitative and quantitative comparison between the region separators computed by PLMSS and their counterpart of MS complex 2-cells is more challenging. Although they are both separate regions with different gradient flow behavior, they are defined and computed differently (Section 6.2).

Yet, for many applications, including the scenarios presented in Section 6.4, the computation of region separators is sufficient without the need for expensive discrete gradient field and dual mesh computations.

In short, the contributions of our work are:

- A scalable algorithm with high parallel efficiency for the computation of PLMSS;
- A detailed performance benchmark that compares PLMSS with the corresponding subprocedures of two state-of-the-art MS complex software libraries (TTK and MSCEER); and
- The integration of PLMSS in TTK to facilitate future benchmarks and reproducibility.

6.1 Related Work

The MS complex subdivides a given scalar field into regions of uniform gradient flow behavior, segmenting the domain such that each point in the same MS manifold will flow towards the same critical point pair, considering forward and backward integration. Two approaches to computing the MS complex arose from Morse theory [171], where either a discrete gradient vector field is defined on the whole domain [69], or piecewise linear Morse theory is used to define a segmentation [9, 10]. For an in-depth comparison of both approaches, we refer to Lewiner's work [153].

Algorithms based on piecewise linear Morse theory are divided into boundary-based and region-growing algorithms. Boundary-based algorithms trace lines of steepest descent/ascent seeded at the saddles, such that every vertex on a line of steepest ascent/descent belongs to the same region. Region-growing algorithms grow sets of top-level cells, e.g., cubes or tetrahedra in 3D, located at the minima and maxima of a scalar function, iteratively enlarging regions. One representative of boundary-based algorithms is Edelsbrunner et al. [61], which first introduced the MS complex for piecewise linear 2-manifolds, recording steepest ascent and descent paths. They also introduced the notion of the quasi MS complex that was extended to 3-manifolds [60] and later improved in geometric accuracy by Bremer et al. [24]. Concerning region-growing algorithms, Danovaro et al. [49] started growing regions by using triangles incident on maxima at vertices, adding edge incident triangles iteratively. They extended this approach by appending

additional seeding points at the initialization, also enabling the processing of higher-dimensional scalar fields [48]. Gyulassy et al. [108] implemented a region-growing algorithm that labels the vertices to extract 3-, 2-, and 1-cells, also extendable from 3D to higher dimensional scalar fields.

Discrete Morse theory was developed by Forman [68], applying Morse theory to any type of simplicial cell complex. Many algorithms build upon the discrete gradient vector field to effectively compute the MS complex [100, 200, 138]. Several parallel and distributed memory implementations were introduced to compute the MS complex efficiently for large datasets. Gyulassy et al. [102] first proposed splitting a dataset into subsets called parcels, extracting the MS complex from each parcel to later merge them in a cancellation-based step. This allowed for the computation of datasets that do not fit into memory and gave rise to distributed memory approaches [109, 184]. Further parallel optimizations were achieved by merging gradient paths, enabling the computation of the gradient assignment and extrema traversals on the GPU [215, 214]. Subhash et al. [223] then accomplished computing all steps of the MS complex computation on the GPU. Even though some algorithms improved the steepest descent line tracing [24, 108] by allowing the traversal to use edges and triangles, still all presented algorithms often produce incorrect connectivity and inaccurate geometry due to the refinement of the underlying discrete domain [113]. Here, Gyulassy et al. [103] implemented a probabilistic algorithm to extract the correct geometry and connectivity. MS complexes were already used in many applications such as material science [104], chemistry [99], and medicine [105], allowing for fast and consistent analysis of the data. Cancellation-based simplification [158, 65, 247] is often used in applications where pairs of critical points are removed to simplify the MS complex and eliminate noise in the dataset. It counteracts over-segmentation of the domain and enables the extraction of persistent features.

The Watershed transform, originally defined by Beucher and Lantuéjoul [14], is another approach to Morse theory, segmenting the domain, usually gray-scale images, into catchment basins that represent the zone of influence of minima and watershed lines, separating catchment basins from each other. Beucher [15] described catchment basins as areas where each drop of water ends up at the same minimum when flowing down the surface. In contrast to Morse theory, minima don't have to be distinct but can rather consist of multiple vertices of the same function value. Those definitions were further improved to be rigorous [168, 173] and extended to the discrete case [169, 239]. De Floriani [52] distinguishes three types of watershed algorithms that are either based on topographic distance, simulated immersion, or rainfalling simulation. Algorithms based on topographic distance compute the short-

est paths to find corresponding catchment basins [170, 169]. Gabrielyan et al. [72] and Yeghiazaryan and Voiculescu [252] provide GPU implementations using an approach similar to path compression, speeding up the shortest path computation. Simulated immersion approaches seed catchment basins at minima, extending them by processing vertices in increasing function value order [239, 218]. Rainfalling simulation algorithms use an inverted logic, finding and labeling minima first, then decreasing in function value for each vertex by steepest descent until a labeled vertex is found [164, 222].

Marching tetrahedra [176] is an algorithm to extract boundary surfaces from a tetrahedralization separating differently labeled vertices from each other. It is a generalization of the marching cubes algorithm [156, 177] that initially allowed the creation of iso-surfaces at a selected isovalue by subdividing voxels such that areas with values above and below the isovalue were separated. Yet, in contrast to the original marching cubes algorithm, it allows multiple labels to be present at each tetrahedron and always extracts a distinct triangulation at each tetrahedron, eliminating ambiguous cases. The effect of various simplicial subdivisions on the quality of the resulting surfaces has been studied by Carr et al.[33]. Also, various applications use this approach for its simplicity and performance [38, 172, 248, 255].

6.2 Preliminaries

This section describes the formal setting of our work. It contains definitions adapted from the Topology ToolKit (TTK) [230, 18]. We refer the reader to textbooks [59, 256] for comprehensive introductions to computational topology.

6.2.1 Input Data

The input is a piecewise linear (PL) scalar field $f : \mathcal{M} \to \mathbb{R}$ defined on a d-dimensional simplicial complex, with $d \leqslant 3$ in our applications. The *star* $St(\sigma)$ of a simplex σ is the set of simplices of $\mathcal{M}$ which contain σ as a face. The *link* $Lk(\sigma)$ is the set of faces of the simplices of $St(\sigma)$ which do not intersect σ. The input field f is provided on the vertices of $\mathcal{M}$ and interpolated on the simplices of higher dimensions. f is assumed to be injective, which is achieved in practice by substituting the f value of a vertex by its position in the non-ambiguous, global vertex order (by increasing f values).

6.2.2 Critical Points

The sub-level set $f^{-1}_{-\infty}(w)$ of an isovalue $w \in \mathbb{R}$ is defined as $f^{-1}_{-\infty}(w) = \{p \in \mathcal{M} \mid f(p) < w\}$. As w continuously increases, the topology of $f^{-1}_{-\infty}(w)$ changes at specific vertices of $\mathcal{M}$, called the *critical points* of f. Let $Lk^-(v)$ be the *lower link* of the vertex v: $Lk^-(v) = \{\sigma \in Lk(v) \mid \forall u \in \sigma : f(u) < f(v)\}$. The *upper link* of v is defined symmetrically: $Lk^+(v) = \{\sigma \in Lk(v) \mid \forall u \in \sigma : f(u) > f(v)\}$. A vertex v is *regular* if and only if both $Lk^-(v)$ and $Lk^+(v)$ are simply connected. Otherwise, v is a *critical vertex* of f [10]. A critical vertex v can be classified by its *index* $\mathcal{I}(v)$, which is 0 for minima, 1 for 1-saddles, $(d-1)$ for $(d-1)$-saddles and d for maxima. Vertices for which the number of connected components of $Lk^-(v)$ or $Lk^+(v)$ are greater than 2 are called *degenerate saddles*.

6.2.3 Integral Lines

Integral lines are piecewise linear curves on $\mathcal{M}$ which locally describe the gradient of f. They can be used to capture and visualize adjacency relations between critical points. Given a vertex v, its *forward* integral line, noted $\mathcal{L}^+(v)$, is a path along the edges of $\mathcal{M}$, initiated in v, such that each edge of $\mathcal{L}^+(v)$ connects a vertex v' to its highest neighbor v''. Then forward integrals are guaranteed to terminate in local maxima of f. A *backward* integral line, noted $\mathcal{L}^-(v)$, is defined symmetrically (i.e., integrating downwards towards minima).

Moreover, we define a *forward extremal integral line* as a forward integral line started at a connected component of upper link $Lk^+(s)$ of a saddle s. Backward extremal integral lines are defined symmetrically. We say that a saddle s is a *forward separating saddle* if there exist at least two forward extremal integral lines starting at s, which terminate in distinct local maxima. Backward-separating saddles are defined symmetrically. In practice, extremal integral lines help capture adjacency relations between critical points.

6.2.4 Morse-Smale Segmentation

In this section, we formalize the notion of MS segmentation computed by our approach.

For a given vertex v, let m and M be its *integration extremities*: m is the local minimum reached by the backward integral line started in v, while M is the local

maximum reached by the forward integral line started in v. We now introduce an equivalence relation $v_1 \sim v_2$ between two vertices v_1 and v_2, which holds if their integration extremities are identical. The MS segmentation is then a decomposition of the set of vertices of $\mathcal{M}$ into maximal subsets $\mathcal{M}_i$, called MS regions, such that for all pairs of vertices $(v_1, v_2) \in \mathcal{M}_i$, we have $v_1 \sim v_2$.

Let τ be a d'-simplex of $\mathcal{M}$ (with $0 \leq d' < d$), which only contains vertices belonging to a single MS region $\mathcal{M}_i$. If the link $Lk(\tau)$ includes vertices which do *not* belong to $\mathcal{M}_i$, we say that τ is a boundary simplex for $\mathcal{M}_i$. Then the *region boundary* of $\mathcal{M}_i$ is the simplicial complex formed by the union of all the boundary simplices of $\mathcal{M}_i$ (and their faces). Each *region boundary* separates $\mathcal{M}_i$ from the remaining dataset. The *region separators* separate all regions $\mathcal{M}_i \in \mathcal{M}$ from each other. To create them, every d-simplex of $\mathcal{M}$ that contains vertices belonging to at least two distinct MS regions spawns $(d-1)$-simplices inside its convex hull, as depicted in Figure 6.4 and Figure 6.5.

6.2.5 Discrete Morse Theory

We now conclude this section of preliminaries with notions (adapted from [98]) of discrete morse theory (DMT) [69], as it has become a central component in modern implementations of the notion of MS complex. We discuss the key differences between the MS complex and the structures extracted by our approach (formalized in Section 6.2.3 and Section 6.2.4).

A *discrete vector* is a pair formed by a simplex $\sigma_i \in \mathcal{M}$ (of dimension i) and one of its co-facets σ_{i+1} (i.e., one of its co-faces of dimension $i+1$), noted $\{\sigma_i < \sigma_{i+1}\}$. σ_{i+1} is usually referred to as the *head* of the vector, while σ_i is its *tail*. Examples of discrete vectors include a pair between a vertex and one of its incident edges or a pair between an edge and a triangle containing it. A *discrete vector field* on $\mathcal{M}$ is then defined as a collection $\mathcal{V}$ of pairs $\{\sigma_i < \sigma_{i+1}\}$, such that each simplex of $\mathcal{M}$ is involved in at most one pair. A simplex σ_i, which is involved in no discrete vector $\mathcal{V}$ is called a *critical simplex*.

A *v-path* is a sequence of discrete vectors $\left\{\{\sigma_i^0 < \sigma_{i+1}^0\}, \ldots, \{\sigma_i^k < \sigma_{i+1}^k\}\right\}$, such that *(i)* $\sigma_i^j \neq \sigma_i^{j+1}$ (i.e., the tails of two consecutive vectors are distinct) and *(ii)* $\sigma_i^{j+1} < \sigma_{i+1}^j$ (i.e., the tail of a vector in the sequence is a face of the head of the previous vector), for any $0 < j < k$. A *v-path* can be interpreted as the discrete analog to the notion of PL integral line introduced in Section 6.2.3. We say that a v-path *terminates* at a critical simplex σ_i if σ_i is a face of the head of its last vector $\{\sigma_i^k < \sigma_{i+1}^k\}$.

Symmetrically, we say that a v-path *starts* at a critical simplex σ_{i+1} if σ_{i+1} is a co-facet of the tail of its first vector $\{\sigma_i^0 < \sigma_{i+1}^0\}$. Then, the collection of all the v-paths terminating in a given critical simplex σ_i is called the *discrete stable set* of σ_i and is noted $\mathcal{M}(\sigma_i)$. Symmetrically, the collection of all the v-path starting at a given critical simplex σ_i is called the *discrete unstable set* of σ_i and is noted $\mathcal{M}'(\sigma_i)$.

A *discrete gradient field* is a discrete vector field such that all its possible *v-paths* are loop-free. Several algorithms have been proposed to compute such a discrete gradient field from an input PL scalar field (see [200] for instance). The *discrete Morse complex* is then defined as the complex formed by the discrete unstable sets of all the critical simplices. It is a cell complex made of d'-dimensional cells (with $d' \in \{0, 1, \ldots, d\}$), such that each d'-dimensional cell is the discrete unstable set of a critical d'-simplex. The *opposite discrete Morse complex* is defined symmetrically, i.e., it is the cell complex formed by the discrete stable sets of all the critical simplices. Finally, the discrete MS complex is defined as the complex formed by the intersections of the cells of the discrete Morse complex and the opposite discrete Morse complex.

Several conceptual differences exist between the MS complex and the MS segmentations in our work. First, as their name suggests, MS segmentations only provide vertex-based decompositions of the input domain, not a cell complex that exhaustively and precisely captures all possible adjacency relations between integral lines (formally v-paths). Thus, MS segmentation targets a subset of the applications enabled by the MS complex (specifically, involving data segmentation). While the separatrices of the MS regions (Section 6.2.4) resemble the 2-dimensional cells of the MS complex, they only correspond to the unstable sets of *separating* saddles (Section 6.2.3), which constitutes a subset of all the saddles (i.e., saddles, where isosurfaces change their genus, are not considered). Finally, note that in DMT, local maxima (critical d-simplices) cannot strictly occur on the boundary of $\mathcal{M}$, which only includes d'-simplices (with $d' < d$).

6.3 Method

In this section, the algorithms for the computation of the PLMSS are described in detail. First, necessary preprocessing steps and data structures are presented. Then, the ascending and descending segmentation of the domain is described, followed by the computation of the MS segmentation.

6.3.1 Preprocessing

To prevent ambiguity during the computation of integral paths, we apply a variant of *Simulation of Simplicity* [62] on the input scalar field f. We first sort all vertices of the domain according to their scalar value, where we resolve ties based on the indices of the compared vertices. Then, we derive the so-called order field $\bar{f}$ that records for each vertex its index in this sorted array. Note that each critical point of f is also a critical point of $\bar{f}$, but $\bar{f}$ might exhibit additional critical points that result from the disambiguation. However, these spurious critical points can be removed via topological simplification, which we apply to remove non-persistent critical points from the scalar field. For a detailed discussion on topological simplification and its implementation in TTK, we refer the reader to the work of Lukasczyk et al. [158].

The advantage of processing an order field over the original input scalar field is that $\bar{f}$ is injective, i.e., every vertex has a distinct largest and smallest neighbor in the order field. It is only possible that a vertex has no neighbor with a larger or smaller order value, in which case the vertex is a maximum or minimum, respectively. Hence, there is always a distinct direction of steepest ascent and descent, which is essential for the computation of the ascending and descending manifolds, described next.

6.3.2 Segmentation and Extrema Retrieval

The segmentation of the domain is a two-step process. In the first step, the ascending and descending segmentations are created, representing areas of influence of minima and maxima, respectively. These segmentations are intersected to create the MS segmentation, representing the areas of influence of minimum-maximum pairs.

Ascending and Descending Segmentation

MS segmentations subdivide a domain into areas of similar flow behavior, meaning that forward and backward integration for any vertex in the same region leads to the same extremum pair. This means that each MS subset of the domain corresponds to all steepest descent/ascent paths that terminate in the same pair of extrema. To achieve this, every vertex must first be assigned to its minimum and maximum. Therefore, the domain's ascending (asc) and descending (dsc) segmentations are computed. As the process is the same for both directions, without loss of generality, it will be described for the descending segmentation.

Maximum assignment for each vertex can be achieved by iteratively finding the largest neighbor's largest neighbor. As this process is lengthy, taking many steps to converge to the maximum, path compression [212] is used to double the step size in each iteration. Figure 6.2 gives an example path compression run using 7 ordered vertices.

The segmentation computation starts by assigning each vertex v to its largest neighbor in the triangulation according to the order field function value $dsc(v) = argmax_{x \in N(v)} \bar{f}(x)$, allowing for a fast lookup later in the process. $N(v)$ is the set of all vertices that are connected to v via an edge in the triangulation.

At the same time, maxima can be extracted by recording cases where no larger neighbor is found. For further processing, each vertex that is not a maximum is written to a list of vertices L_0 that have not found their maximum yet.

In the second step, the maximum for each vertex in L_0 is found using path compression. Here, the value of each vertex gets assigned to the largest neighbor's largest neighbor $dsc(v) = dsc(dsc(v))$. This allows doubling the step size towards the maximum in each iteration. If the corresponding maximum is not found $dsc(v) \neq dsc(dsc(v))$, the vertex does not converge to its maximum and is written to a second list L_1. After fully iterating over L_0, the process starts again using L_1, i.e., $L_0 = L_1$. If $L_1 = \varnothing$ after iterating over L_0, the maximum $dsc(v)$ is found for every vertex v.

In parallel environments, each vertex can be evaluated independently with little communication in between iterations, as both steps iterate over a set of vertices. Here, the first step of finding the largest neighbor is equally distributed so that every thread executes the same number of vertices. Still, every thread t keeps a local list of active vertices, i.e., L_{0t}, L_{1t}, executing the following iterations independently for each thread. Computing the ascending and descending segmentations simultaneously is also possible, further improving performance. To do this, both the largest and smallest neighbors are found simultaneously in the first step, and a vertex is added to L_1 if it did not converge in both directions in the second step.

Morse-Smale Segmentation

The ascending and descending segmentation obtained by the algorithm above can be combined into an MS segmentation. As the descending segmentation assigns a maximum to each vertex and the ascending segmentation assigns a minimum to each vertex, vertices with the same minimum and maximum are assigned to the

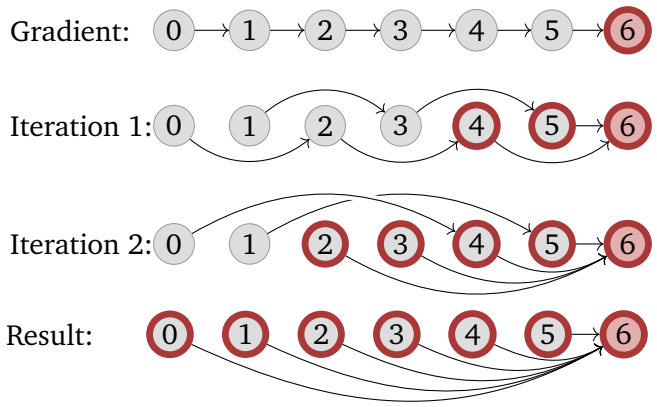


Fig. 6.2: Path compression example showing vertices as circles and current vertex assignment as arrows. The number attached to a vertex is the order field function value, and the outer ring shows if a vertex converged (red). The gradient step assigns the largest neighbor to each vertex, and each following iteration sets the neighbor's neighbor for active vertices. Please note that the order in which every iteration is executed matters. In this example, it starts with the smallest active vertex and continues in an increasing fashion. If iteration 1 starts with the largest vertex in decreasing order, the assignment will already terminate after the first iteration.

same MS ID. Therefore, the extremum pair combination is written to each vertex as a tuple, allowing access to the involved extrema. This process is trivial to parallelize, as each thread can independently write the MS IDs for its vertices.

6.3.3 Multi-Label Marching Triangles/Tetrahedra

To visually divide MS regions from each other, region-separating geometries can be created between the regions. As the triangulation consists of triangles in the 2D case and tetrahedra in the 3D case, both cases must be treated slightly differently. In 2D, region-separating geometry is created using edges that split triangles with multiple labels, whereas in 3D, triangles are utilized to separate the vertices of multi-label tetrahedra. Like marching tetrahedra [57], each tetrahedron or triangle is evaluated independently, considering the labels at its vertices for generating the bisecting geometry.

Triangles

In the 2D case, a triangle can either have 1, 2, or 3 unique labels at its vertices. In the case of 1 label, no edges must be generated as the vertices belong to the same region. When two different labels are present, one vertex a has a different label than

the other two vertices b, c. Here, as shown in Figure 6.4, the centers of the edges connecting a to b and a to c are used as the endpoints for the edge that splits the labels. In the case of 3 unique labels, an edge is created from the triangle center to all three of its edges.

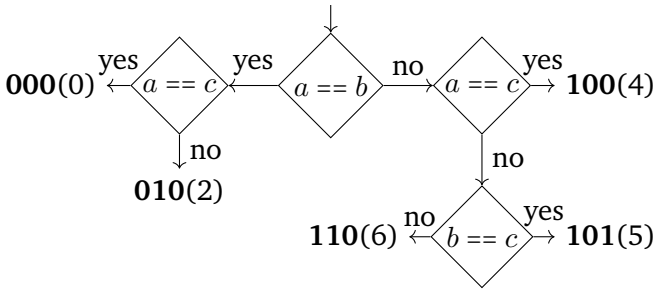


Fig. 6.3: Decision tree for the triangle binary code creation. Each rhombus represents a decision, and each node shows the resulting code with the bit representation in bold and integer representation in brackets.

Computationally, this is achieved using a lookup table describing every possible triangle configuration. Therefore, a 3-bit binary code with values in the range of $\{0, .., 6\}$ is created to describe the current triangle configuration, utilizing the labels at the three vertices a, b, c of the considered triangle, converted to a dense local representation such that $a = 0$, $b \in \{0, 1\}$, $c \in \{0, 1, 2\}$. The label a is always considered to be 0, b can either be 0 or 1 depending on the equality to label a, and c can either be 0, 1, or 2 depending on the equality to labels a and b. Therefore, $b \in \{0, 1\}$ determines the left bit and $c \in \{0, 1, 2\}$ determines the last two bits. Figure 6.3 provides the decision tree. It should be noted that some binary codes cannot appear, as $c = 1$ can only be the case if $b = 1$. Also, codes like $011(3)$ are not generally possible, as there is no label 3.

Each of the five valid binary codes corresponds to a triangle configuration, as shown in Figure 6.4. This allows retrieving the triangle edges that need to be connected. The whole procedure is well-scaling as it is executed per triangle.

Tetrahedra

In 3D, the domain is subdivided into tetrahedra. Therefore, up to 4 unique labels a, b, c, d can be present at the vertices of a tetrahedron. Like the triangle case, a 5-bit binary code is created and translated into a tetrahedron configuration. This configuration creates a consistent triangulation that separates unique labels from each other.

An extended logic creates the binary code for tetrahedra. a is considered to be 0 again, $b \in \{0, 1\}$ determines the left bit, $c \in \{0, 1, 2\}$ determines the next 2 bits and $d \in \{0, 1, 2, 3\}$ determines the last two bits. If the label of a vertex with lower index matches, its label is used as the resulting label, otherwise the index of the own vertex is used. All valid configurations are provided in Table 6.1. Some binary codes are invalid, as some labels might not exist and can not be assigned to a vertex of a higher index. E.g. $00010(2)$ is not possible as d would have label 2, but c has label 0, making label 2 non-existent in this configuration.

Tab. 6.1: All valid binary codes, the codes converted to an integer, the number of unique labels of the tetrahedron, and the value of each label.

Binary code	Case	#Labels	a	b	c	d
00000	0	1	0	0	0	0
00011	3	2	0	0	0	3
01000	8	2	0	0	2	0
01010	10	2	0	0	2	2
01011	11	3	0	0	2	3
10000	16	2	0	1	0	0
10001	17	2	0	1	0	1
10011	19	3	0	1	0	3
10100	20	2	0	1	1	0
10101	21	2	0	1	1	1
10111	23	3	0	1	1	3
11000	24	3	0	1	2	0
11001	25	3	0	1	2	1
11010	26	3	0	1	2	2
11011	27	4	0	1	2	3

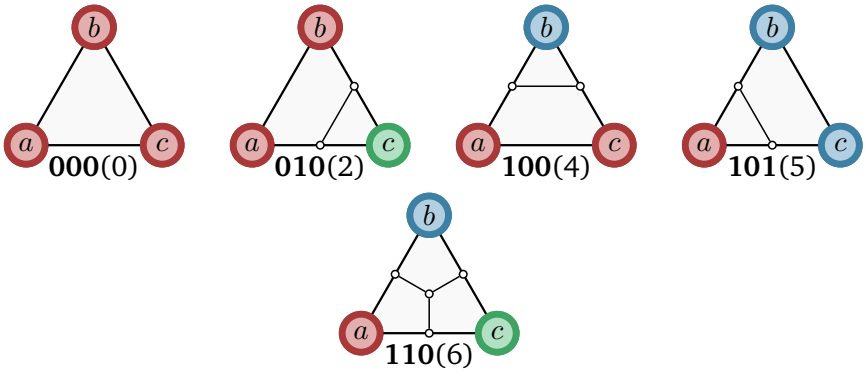


Fig. 6.4: All five valid cases for splitting a triangle. The label at the vertices of the triangles is drawn by color (red = 0, blue = 1, green = 2), showing binary code and integer representation at the bottom and the resulting separating edge(s) in the middle. White circles mark the intersection points on the edges of the triangle.

After the binary code is determined, a lookup table retrieves the resulting separating triangles utilizing the edge, triangle, and tetrahedra centers to be connected. This allows for a fast triangulation of the tetrahedra, as the resulting triangle vertices are directly retrieved. Figure 6.5 shows the resulting triangulation for all cases, ignoring permutations and rotations. The triangulation across tetrahedra is always consistent, as the triangle labels of the tetrahedron mimic the 2D case, i.e., triangles are either split by connecting their edge centers to their triangle center or two triangle edges are connected. Therefore, the triangle connecting two incident tetrahedra is always split in the same way, and the resulting triangulation separates labels from each other without any holes in the geometry.

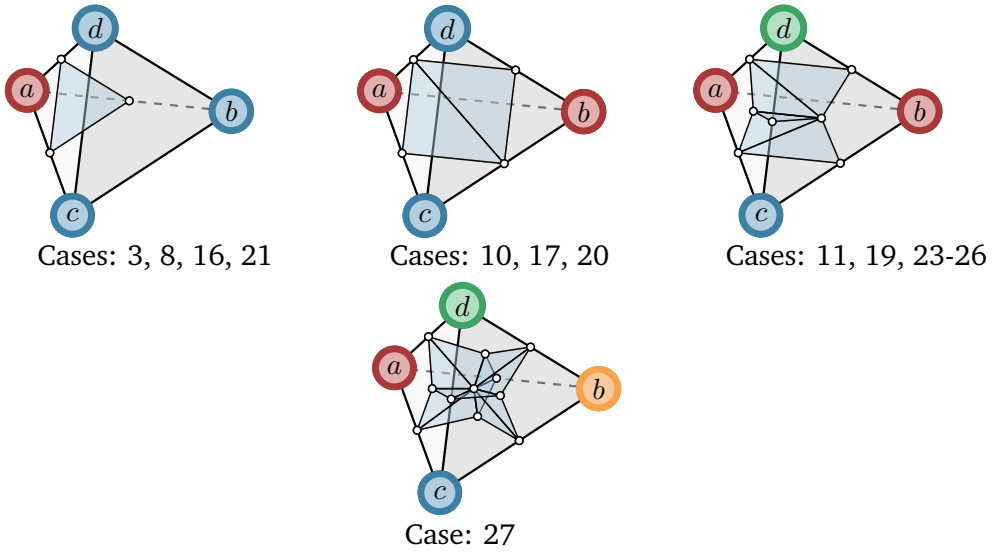


Fig. 6.5: Triangulation for tetrahedra with more than 1 unique label ignoring permutations and rotations. The case number below the tetrahedra refers to Table 6.1.

Triangulation Options

As the information requested slightly varies from user to user, two region-separating geometries are available. The user can utilize any segmentation (ascending, descending, and MS) together with either region boundaries or region separators. Figure 6.1 showcases a closeup view of the segmentations (b-e), where (b) and (d) show region separators, and (c) and (e) show region boundaries.

Segmentation Selection: For some applications, a certain segmentation will be of great interest. Here, either the MS, ascending, or descending segmentation can be chosen to deliver the labels for the marching tetrahedra algorithm. Instead of using the MS segmentation, it is also possible to show the union of the ascending and

descending segmentation, producing intersecting geometry at the meeting points of both region-separating geometries.

Region Separating Geometries: Both the region separators and the region boundaries have slightly different use cases and allow the mapping of different information to its geometry. The region separators divide all regions, or areas of influence of minima-maxima pairs, from each other by building geometry between them. Therefore, information about the minima and maxima involved in each separating triangle can be displayed to the user. Each subset of the surface that separates the same two regions can be extracted here. Still, some overhead is involved in computing the region separators, as many triangles have to be used to separate the regions from each other, depicted in Figure 6.5.

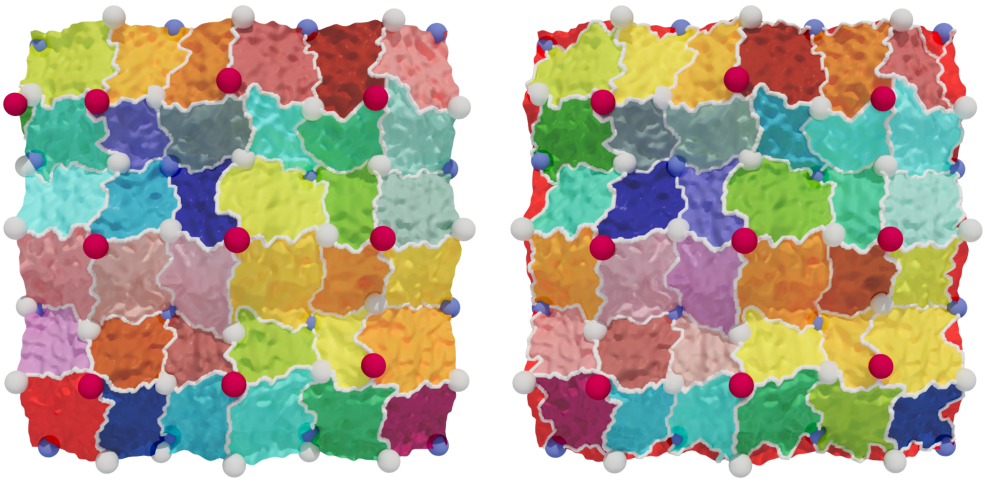
Region boundaries allow the extraction of the hull of regions or areas of influence of minima-maxima pairs. They use all tetrahedra with three vertices of the same label. Those three vertices form a triangle in the input simplicial complex that can be directly used as a separating geometry. This option will result in faster computation times and allow the extraction of geometry for each region.

6.4 Results

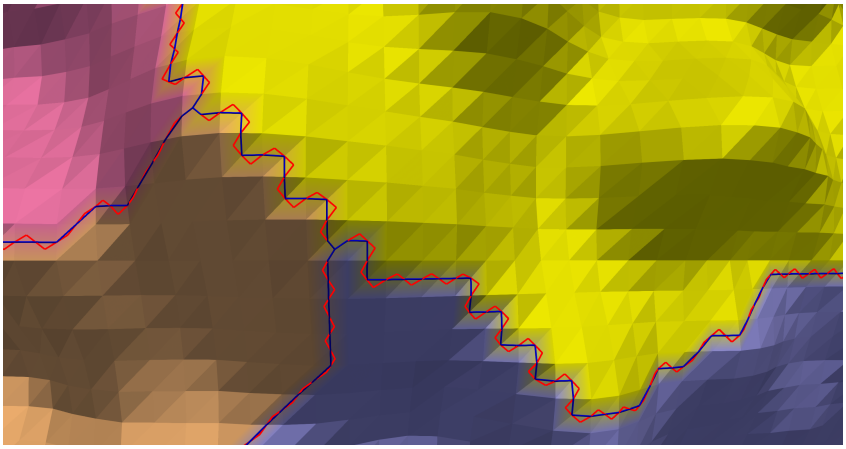
This section provides three example datasets and their region-separating geometry outputs for both algorithms. The Noisy Terrain dataset in Section 6.4.1 highlights differences between both algorithms in the 2D case. Section 6.4.2 shows that both algorithms produce a very similar output when no saddle-saddle 2-cells are present. The last two datasets indicate that the MS complex provides more geometry than necessary to extract useful information effectively in many cases, while the PLMSS can extract the areas of influence without additional preprocessing.

6.4.1 Noisy Terrain

The Noisy Terrain dataset is a triangulated surface with elevation scalars attached, chosen as an illustrative example. It has a 300x300 resolution and can be found in the TTK data repository [232]. Generally, the dataset consists of hills and valleys on a regular grid, where the hills are getting smaller the closer they are to the border. Additionally, noise was added to the terrain to showcase topological simplification.



(a) PLMSS **(b)** MS complex



(c) Region separators

Fig. 6.6: Noisy Terrain dataset [232] showing the PLMSS and MS complex next to each other. Both show critical points (blue = Minima, white = Saddles, red = Maxima), separating geometries in white and the surface colored by MS complex IDs. The border region of the MS complex is not fully segmented as a vanilla implementation of the expansion-based discrete gradient computation algorithm [200] may miss maxima on the boundary, whereas the boundary of the PLMSS is fully segmented. (c) visually compares the region-separating geometry of the PLMSS (blue) and the MS complex (red). The MS complex produces geometries with a characteristic step-function shape (red), whereas the PLMSS produces separating geometries with a linear-slope shape (blue).

Figure 6.6 compares the PLMSS with the MS complex, using the same color coding for critical points, separating geometries, and segmentation in both versions. Slight differences can be detected regarding the separating geometries, where the MS complex separating geometries are defined on the dual graph and are connecting triangle centers in the 2D case. Concerning the PLMSS, triangles are split according

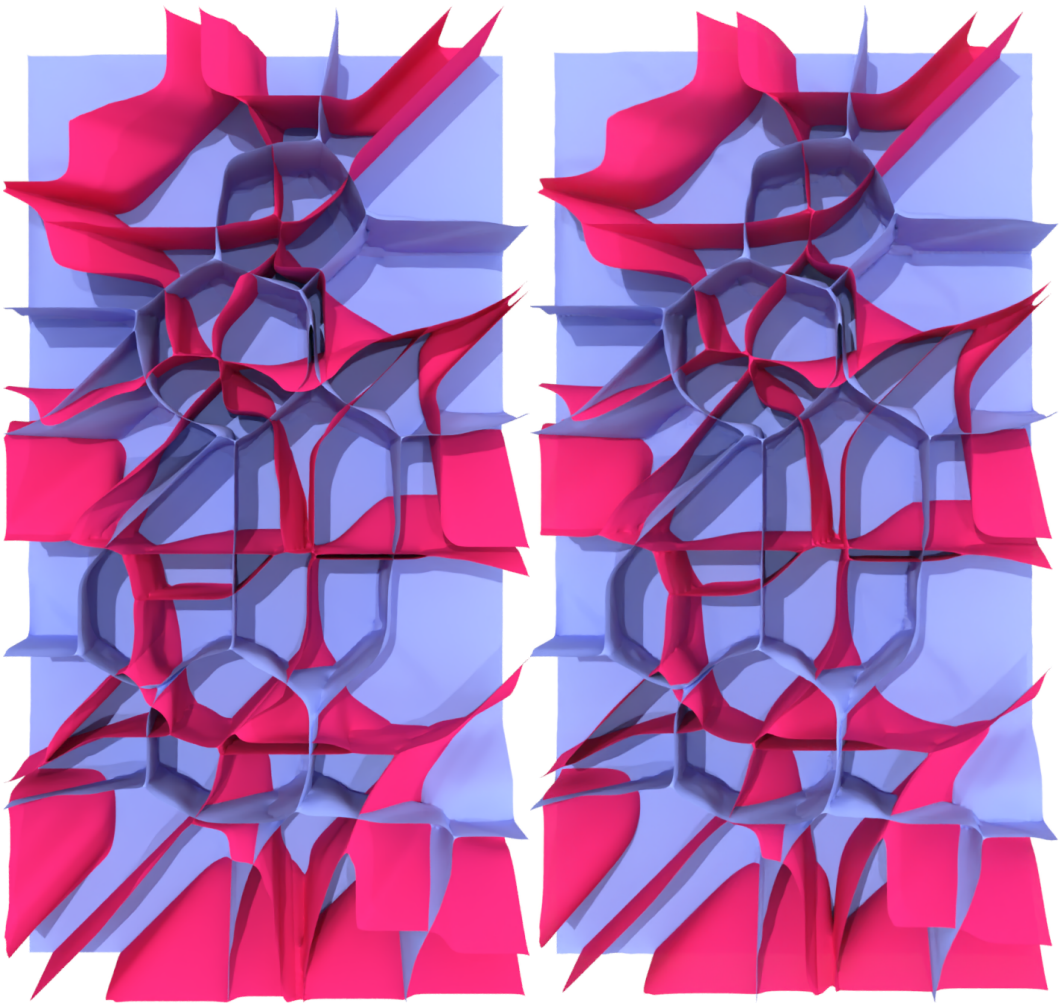
to the labels at the triangle vertices. Another difference can be spotted at the borders of the MS complex, where great sections of the border are not labeled, as a vanilla implementation of the expansion-based discrete gradient computation algorithm of Robins et al. [200] (implemented in TTK) may miss PL maxima on the domain boundary (local post-processing of the discrete gradient would be required to enforce the detection of discrete maxima in the star of boundary PL maxima). As the PLMSS assigns a maximum-minimum pair to each vertex, every vertex is properly labeled without skipping the boundary region.

6.4.2 AT Molecule

The AT dataset from the TTK Tutorial Data [233] shows the simulation of the electron density of a molecule restricted to a plane but embedded in 3D space. This example, provided in Figure 6.7, shows that the PLMSS and MS complex extract the same underlying geometry at heart if no saddle-saddle 2-cells are present in the dataset. To allow an easier comparison of the separating geometries, all geometries were colored by the underlying segmentation (ascending in red and descending in blue) and smoothed. Most of the separating geometry coincides with at most one tetrahedron space in between both representations. The largest difference can be seen in the two narrow red geometries on the middle right of the images, as their distance is smaller when using the PLMSS. The difference between both representations results from the dual graph definition of MS complex, compared to the per-vertex definition of the PLMSS.

6.4.3 Viscous Fingering

The Viscous Fingering dataset [232] represents the result of finite pointset method simulations that simulate the mixing of salt solutions inside water. Regions of high salt concentration form structures called viscous fingers. The analysis of such structures usually involves Reeb graphs or iso-surfaces to identify single fingers in the dataset [157]. The MS complex suffers from additional saddle-saddle 2-cells in such scenarios that stem from discrete Morse theory, complicating effective analysis. Filtering these saddle-saddle 2-cells out of the set of 2-cells is usually expensive as further post-processing is required to identify and simplify the saddles responsible for such 2-cells. The PLMSS, on the other hand, segments the data into regions of influence of maxima-minima pairs, providing a region for each partial viscous finger. The granularity of these separating geometries can be controlled by

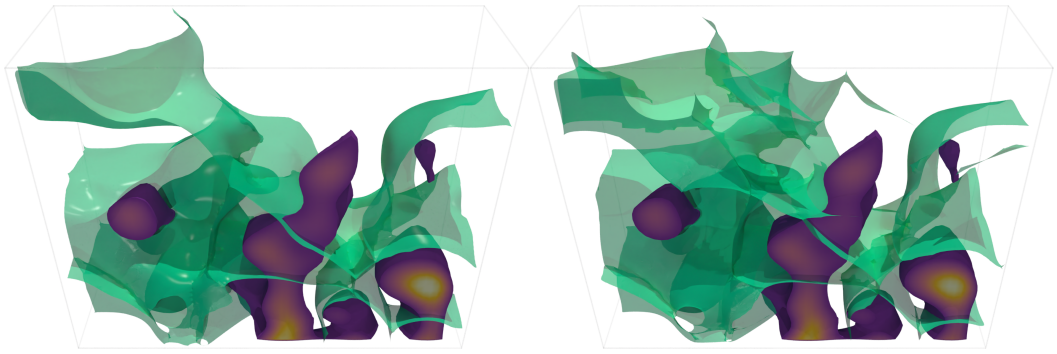


(a) PLMSS **(b)** MS complex

Fig. 6.7: Comparison between the region separators computed by PLMSS (a), and the MS complex 2-cells computed by TTK (b) using the AT dataset. The resulting surfaces are colored by the segmentation type, i.e., red for the descending and blue for the ascending segmentation. To ease visual comparison, both surfaces were smoothed 20 times using TTK's geometry smoother. Both surfaces are almost visually identical.

topological simplification, allowing users to achieve the desired level of detail of the segmentation.

Figure 6.8 shows the Viscous Fingering dataset from the TTK data repository [232] with an applied persistence threshold of 0.1. The colored isosurfaces provide the finger surfaces that correspond to regions of high salt concentration. It was clipped in the middle after all geometries were created to allow a deeper look into the dataset. Figure 6.8a shows that the PLMSS effectively separates fingers from each other, providing the area of influence of the maxima and hence, for the viscous



(a) PLMSS **(b)** MS complex

Fig. 6.8: Comparison of the PLMSS region boundaries and the MS complex using the Viscous Fingering dataset [232] simplified with an absolute persistence threshold of 0.1. Both images show the boundary interface of viscous fingers as contours of the salt concentration density scalar field, colored by the density from yellow (high concentration) to purple (low concentration). (a) shows that the PLMSS region boundaries can effectively extract the region-separating geometries that separate single viscous fingers without cluttering the visualization. (b) provides the original MS complex, where more geometry is extracted due to the saddle-saddle 2-cells cluttering the image. Those saddle-saddle walls would have to be removed by additional postprocessing.

fingers. The region separators are used to extract the area of influence of single fingers to analyze the area they are growing into with increasing salt concentration. Figure 6.8b provides the MS complex. As for additional saddle-saddle 2-cells, the image is less clear and more cluttered, which can be problematic with noisy or large datasets. Those saddle-saddle walls would have to be removed by additional postprocessing.

6.4.4 Rayleigh-Taylor Instability (Miranda)

The concept of controlled fusion using hydrogen isotopes in a laser-lighted fuel capsule was leading to the discovery of Rayleigh-Taylor instability [43] at the boundary of the capsule. The simulation models the heating process of two hydrogen isotopes for fusion burn. The energy from the laser is non-uniformly distributed and causes small perturbations that quickly grow. One time step of a simulation is analyzed using the MS complex of TTK and the PLMSS.

An absolute persistence threshold of 0.1 was applied to filter noise from the data by utilizing localized topological simplification [158]. By solely extracting the border surfaces of the area of influence created by maximum-minimum pairs, the PLMSS

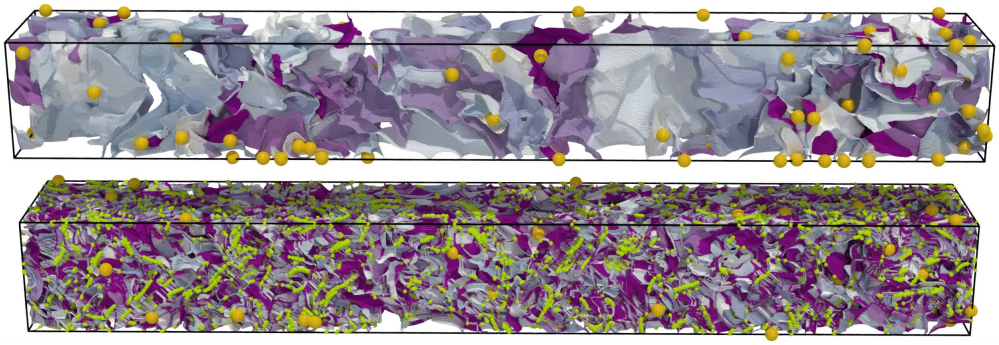


Fig. 6.9: Illustration showing the results of the PLMSS (top) and the MS complex (bottom) utilizing a Rayleigh-Taylor instability simulation [43] dataset. The image shows extrema as large orange spheres, additionally providing saddles as small green spheres in the MS complex, as saddle-saddle separatrices are the actual reason for the cluttered MS complex visualization. Even though filtering out saddle-saddle separatrices is possible, the computational overhead will favor the PLMSS in situations where saddle-saddle separatrices hide important features.

removes clutter from the separating geometries of the MS complex due to the missing saddle-saddle 2-cells.

Figure 6.9 compares the visual results of the PLMSS with the MS complex, using one timestep of a Rayleigh-Taylor instability simulation. Here, the PLMSS extracts the area of influence of minima and maxima in the dataset. The region-separating geometry is very structured, and the boundary between regions of interest can be identified. In contrast, the MS complex introduces a lot of noise due to the remaining saddle-saddle 2-cells that clutter the resulting image. Additionally, the maxima on the boundary are missing, as a vanilla implementation of the expansion-based discrete gradient computation algorithm of Robins et al. [200] (implemented in TTK) may miss PL maxima on the domain boundary.

6.5 Performance

In this section, the computational performance of our implementation is analyzed and compared to the MS complex implementation of TTK [230] and MSCEER [101]. As hinted by the authors of MSCEER, we use the “steepest_lstar” and “extractms” packages, as they supply the fastest implementation without accurate geometry. Both strong scaling studies show that the PLMSS is scaling well with core count due to the mentioned improvements. We utilize the Rayleigh-Taylor instability (Miranda) dataset [43] at a resolution of 512^3 and the Foot dataset [232] at a resolution of 256^3 for the computation speed comparisons of all three algorithms.

Tab. 6.2: Raw timing data of the MSCEER, TTK, and PLMSS algorithms in seconds. For each timing, the test was executed ten times, removing the best and worst time regarding the time listed in the last row and averaging the remaining runs. It should be noted that the top-level cell count is six times higher with TTK and PLMSS than with MSCEER. Similarly, the total number of cells in the input simplicial complex differs by a factor of roughly 3.24 (Miranda MSCEER: 1,070,599,167 / TTK, PLMSS: 3,473,956,851) (Foot MSCEER: 133,432,831 / TTK, PLMSS: 432,287,731).

		Miranda 512^3						Foot 256^3					
Task	Algo.	1T	2T	4T	8T	16T	24T	1T	2T	4T	8T	16T	24T
DGF	TTK	1,326.76	685.52	365.00	200.15	112.10	84.55	152.65	78.00	42.45	23.32	13.32	10.08
	MSCEER	116.51	67.33	34.43	20.25	15.11	12.80	20.54	12.25	7.78	5.67	4.94	4.86
	PLMSS	-	-	-	-	-	-	-	-	-	-	-	-
Asc/Desc	TTK	442.15	269.68	197.04	167.68	158.19	152.42	35.12	28.23	24.54	22.92	22.38	22.63
	MSCEER	167.60	99.48	53.12	28.71	15.89	11.63	34.77	24.07	21.35	19.88	18.97	19.25
	PLMSS	39.61	20.08	10.64	5.73	3.07	2.31	4.40	2.23	1.16	0.68	0.43	0.36
MS	TTK	4.78	2.44	1.40	1.02	0.73	0.66	0.46	0.23	0.18	0.16	0.11	0.10
	MSCEER	-	-	-	-	-	-	-	-	-	-	-	-
	PLMSS	0.32	0.16	0.09	0.05	0.03	0.02	0.04	0.02	0.01	0.01	0.00	0.00
Index	TTK	152.24	90.30	46.73	24.59	12.80	9.43	26.73	16.49	8.83	4.58	2.56	2.07
	MSCEER	117.00	58.92	30.87	16.20	8.68	6.24	15.70	8.37	4.90	3.00	2.07	1.77
	PLMSS	39.00	19.62	10.24	5.28	2.72	1.89	4.81	2.42	1.27	0.67	0.41	0.29
Geometry	TTK	491.21	263.75	143.22	83.25	50.67	41.04	94.11	49.84	27.69	16.05	10.02	8.25
	MSCEER	-	-	-	-	-	-	-	-	-	-	-	-
	PLMSS	8.99	6.68	4.12	2.81	2.03	1.79	0.59	0.42	0.27	0.18	0.13	0.14
DGF +	TTK	1,921.16	1,045.49	608.77	392.42	283.09	**246.39**	214.50	122.71	75.82	50.81	38.26	**34.78**
Index +	MSCEER	401.12	225.73	118.43	65.16	39.68	**30.67**	71.02	44.70	34.03	28.56	25.98	**25.87**
Asc/Des	PLMSS	78.61	39.70	20.89	11.01	5.79	**4.20**	9.21	4.65	2.43	1.35	0.84	**0.65**

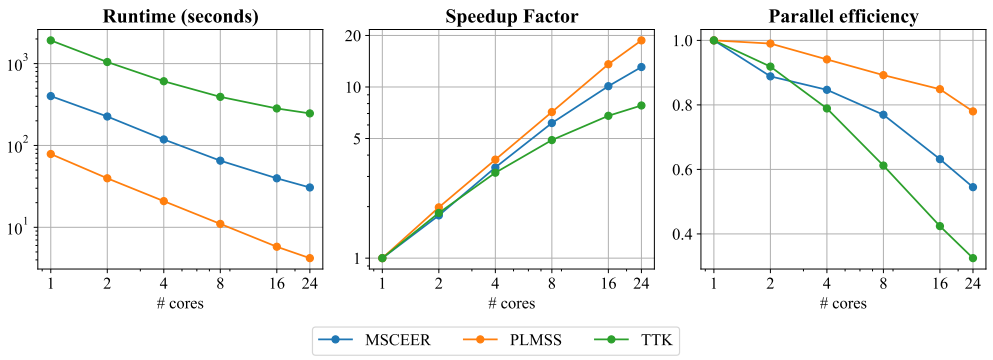


Fig. 6.10: Rayleigh-Taylor instability dataset [43] performance of all three algorithms at a resolution of 512^3. The left graph shows the runtime sum of the DGF, ascending and descending segmentation, and indexing as a log-log plot regarding the number of cores used for the experiment. In the middle, the speedup factor, i.e., the runtime of 1 thread divided by the runtime of x threads, is also shown as a log-log plot regarding core counts. On the right, the parallel efficiency, i.e., the speedup factor divided by the number of cores, is represented in a semi-log plot.

6.5.1 Algorithmic Improvements

In general, three main aspects of the PLMSS result in a strongly reduced computation time as compared to the MS complex. First, the segmentation of the domain is improved by path compression, which is much faster than computing a discrete gradient field. Second, the multi-label marching tetrahedra algorithm supports

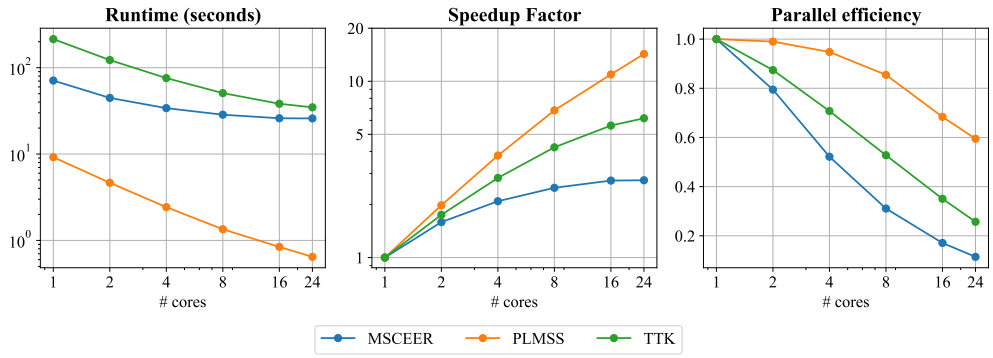


Fig. 6.11: Foot dataset [232] performance of all three algorithms at a resolution of 256^3. The left graph shows the runtime sum of the DGF, ascending and descending segmentation, and indexing as a log-log plot regarding the number of cores used for the experiment. In the middle, the speedup factor, i.e., the runtime of 1 thread divided by the runtime of x threads, is also shown as a log-log plot of core counts. On the right, the parallel efficiency, i.e., the speedup factor divided by the number of cores, is represented in a semi-log plot.

computing the separating geometries in a well-scaling way. Third, splitting the marching tetrahedra algorithms into indexing and geometry creation steps allows the allocation of the resources needed in the geometry creation step without additional computations.

Segmentation

The segmentation of the domain into the ascending and descending manifold and the intersection of both manifolds, called MS manifold, are computed differently for the MS complex and the PLMSS. Both MS complex implementations require a discrete gradient field to be computed first, then following the v-paths along the gradient field to assign labels to each vertex. However, the PLMSS segmentation utilizes path compression to assign each vertex to its designated minimum or maximum without the need for any additional structure other than the order field.

For path compression to work, all neighbors of each vertex must be visited once to get the largest and smallest neighbor of that vertex. With this information, the maximum can be found iteratively by assigning its largest neighbor's largest neighbor to the vertex. This process is executed in multiple iterations, where each iteration finds the designated maximum of a vertex or a vertex closer to the designated maximum. Here, each time the step length is doubled, yielding extremum assignment in $log(s)$ steps for each vertex, where s is the number of vertices on the integral line of the vertex to the extremum. Also, the iterations get smaller every time as more and more vertices are assigned to their extremum.

Table 6.2 shows the timings of those steps in the first three rows, where "DGF" refers to the discrete gradient field computation, "Asc/Desc" refers to the ascending and descending segmentation, and "MS" refers to the MS segmentation. Please note that the MSCEER algorithm does not compute an MS segmentation in the provided implementation.

Even in a single-threaded environment, the performance gains of retrieving the ascending and descending segmentations already show strong improvements in the computation time of the PLMSS compared to the MS complex implementations. For the Miranda dataset, PLMSS only needs a total of 39.61s for the computation of the ascending and descending segmentation. Comparing this to the MS complex implementations (MSCEER: 284.11s / TTK: 1768.91s) leaves us at a speedup of roughly 7x and 44x, respectively.

Multi-Label Marching Tetrahedra

After segmenting the domain into various regions, region-separating geometries that help visualize the segmentation can be created. Again, the PLMSS and the MS complex implementations differ in the realization of this step. Regarding the MS complex implementations, paths between critical point pairs must be traced on the DGF, whereas the PLMSS uses a marching tetrahedra algorithm.

Marching tetrahedra algorithms scale very well as they are executed per vertex. In our implementation, the binary code, as described in Section 6.3.3, and the number of triangles created per thread are computed in a preliminary step. This allows the allocation of the exact amount of memory needed for the triangles. In a follow-up step, this enables direct writing of triangles to memory.

Table 6.2 shows the timings of those steps in the 4th and 5th row, where "Index" refers to the computation of simplex indices that will spawn triangles, and "Geometry" represents writing the triangles to memory. Please note that the MSCEER algorithm does not compute the geometry itself in the provided implementation but only gathers the indices of relevant simplices.

A comparison of the single thread timings with the Miranda dataset shows that the indexing is almost four times faster compared to TTK and three times faster than MSCEER. These speedups come from the excessive use of lookup tables and a per tetrahedra execution that does not require tracing paths. With the additional memory counting in the indexing step and the lower number of triangles created, the geometry-creating part strongly improved regarding computation times.

6.5.2 TTK vs. MSCEER

Both MS complex solutions are slightly different in the specifics of their implementation. First of all, TTK uses a simplicial complex that uses tetrahedra and triangles as top-level simplices, whereas MSCEER uses cubes and squares. This already leads to roughly 324% more total cells and six times more top-level cells in the case of the TTK-based implementation, such as the MS complex or PLMSS. This will have an effect on the runtime of the algorithms, as well as the resolution of the extracted region-separating geometries. Still, this is only a limitation of the current TTK version and might change in the future to trade accuracy for runtime efficiency.

Another issue when comparing both algorithms is the output generated by each implementation. The version provided by Gyulassy et al.[101] only provides an ascending and descending segmentation of the domain and the indices of top-level cells that would spawn region-separating geometries. The MS complex representation and the surface geometries would be created by a different software at runtime that was not supplied with the library.

6.5.3 Strong Scaling Setup

Both strong scaling studies were carried out on a dual Intel XEON SP 6126 node with 24 CPU cores and 384GB of RAM. For each algorithm, various timings were created, starting with the computation of the discrete gradient field, ascending and descending segmentation, MS segmentation, indexing of tetrahedra or cubes that generate region-separating geometries, and writing the region-separating geometries to memory. The results for both studies are shown in Table 6.2, where the last row shows the total time to get an ascending and descending segmentation and mark all top-level cells that generate region-separating geometries. To mimic the layout of the 2-cells of the MS complex, the region separators of the PLMSS were computed for those results.

The experiment was executed ten times for each combination of the dataset, the number of threads, and the algorithm. The best and the worst of these ten runs, regarding total computation time, were discarded. The remaining eight runs were averaged to achieve more stable results.

6.5.4 Strong Scaling: Rayleigh-Taylor Instability

Utilizing the Rayleigh-Taylor instability dataset [43], a strong scaling analysis was carried out for a 512^3 subset of the data, simplified with a persistence threshold of 0.1. Computation times, speedup factor (i.e., the time a single thread takes divided by the time the current number of threads take), and parallel efficiency (i.e., speedup factor divided by the number of threads) are plotted against the number of cores, shown in Figure 6.10.

The total execution time in the last row of Table 6.2 shows that the PLMSS is more than an order of magnitude faster than TTK and 5 to 7 times faster than MSCEER. As the PLMSS still has to be executed on roughly three times the cells, this is still an improvement of an order of magnitude regarding the time per cell. The speedup factor and parallel efficiency also show a clear trend that PLMSS is scaling very well with more cores. In this regard, MSCEER beats TTK in terms of scalability but clearly performs worse against PLMSS. The speedup factor graph also hints that more cores would be beneficial in future experiments as PLMSS still scales well at 24 cores. Due to insufficient computational resources, we were unable to offer a larger strong scaling analysis.

6.5.5 Strong Scaling: CT Scan of a Human Foot

The foot dataset [232] was chosen to represent smaller data sizes with a resolution of 256^3, simplified with a persistence threshold of 110. It consists of a CT scan of the tip of a foot, where the threshold of 110 was chosen to represent each bone with its own region.

Regarding the total execution time at 24 Threads in Table 6.2, a speedup of over 50x and almost 40x (TTK and MSCEER) can be achieved using PLMSS. The resulting graphs in Figure 6.11 also show clear improvements regarding parallel efficiency, as PLMSS $(0, 59)$ still achieves good results that hint towards using even more cores, where TTK $(0, 26)$ and MSCEER $(0, 11)$ are already in a range where more cores do not strongly improve runtime performance and communication overhead takes over.

6.5.6 Discussion

The PLMSS showed good improvements over the MSCEER and TTK implementation for both datasets, yielding a great parallel efficiency. Great runtime performance improvements of roughly 40x were achieved for the foot dataset. Even with the 5-7x runtime improvement regarding the Rayleigh-Taylor instability dataset, 6x more top-level cells had to be traversed compared to MSCEER.

6.6 Limitations

Our entire approach aims to efficiently compute the input scalar field's ascending and descending segmentations. Its output is not a complete MS complex. First, it does not capture saddle-saddle connectors, which may be useful in certain applications. Second, it does not output an explicit CW complex modeling the MS complex (i.e., where vertices encode critical points, 1-dimensional cells encode separatrices, 2-dimensional cells encode separating geometries, and 3-dimensional cells encode regions with identical integration extremities), only the domain segmentation is provided. This can be detrimental in applications involving post-processing of the MS complex, such as regular remeshing [58] or hierarchical simplification [107]. For these applications, a standard algorithm based on discrete Morse theory (such as the one available in TTK [230] or MSCEER [101]) should be preferred.

6.7 Perspective on the Uncertainty-aware Visual Analytics Process

The presented method itself is a versatile tool to use in UAVA applications but does not present a UAVA application on its own. Still, it can automatically segment arbitrary scalar fields, for example, a sampled probability density function (PDF). High and low probability areas can be analyzed, providing the influence of maxima and minima in those functions. Also, the path compression only needs vertices and edges to function, allowing it to operate on graphs with attached scalars.

Generally, the concept of the PLMSS applies to various areas using scalar fields as their input. This enables the analysis of ensembles of scalar data, where the PLMSS is computed for each scalar field. Now, the variability of the PLMSS cells throughout the scalar fields can help to find areas of uncertain topology, redeeming uncertainty

quantification. Here, the performance advantages over the MSC play a big role, as multiple datasets have to be segmented.

6.8 Conclusion and Future Work

The presented algorithm describes a well-scaling approach to computing MS segmentations, allowing for speedups of more than an order of magnitude compared to two MS complex implementations. Utilizing path compression to create the segmentations allows us to quickly extract the labels for the multi-label marching tetrahedra algorithm that powers the generation of region-separating geometries. The algorithm is faster and has a lower memory footprint, as no discrete gradient vector field and little preprocessing of the triangulation is needed. Only a scalar field saving the 5-bit index representation for each vertex has to be created. The utilization of only top-level cells and vertices simplifies triangulation generation. Additionally, DMT design does not allow maxima at the border of the MS complex, often leading to missing border regions in each segmentation. This issue is fixed by segmenting the whole domain and retrieving all border maxima in the process. Regarding the generated separating geometries, several use cases have been presented that show the applicability of our approach where the MS complex failed to deliver without expensive post-processing using saddle-saddle 2-cell cancellation. Simply speaking, our algorithm allows us to extract areas of influence of minima, maxima, and minima-maxima pairs by separating their boundaries with two available separating geometries that can be triggered by three segmentation options. Still, this does not invalidate the MS complex as we only compute a segmentation and not the complex itself. Features like the saddle-saddle connectors are not computed. Therefore, we conclude that the PLMSS is a versatile tool to generate MS segmentations in a well-scaling parallel way, allowing users to explore their data much faster while returning to the MS complex on demand.

For future work, we are planning to improve the PLMSS in various ways. The load on each of the threads can be imbalanced when a particular thread receives a lot of triangles to generate. Here, a workload balance system could be introduced. To scale to even larger datasets, an MPI implementation will be provided to enable distributed memory execution. The knowledge gained from creating the segmentations will be implemented into the TTK MS complex implementation to improve its performance and useability. As shown in Section 6.5, a thorough comparison between MS complex implementations is out of the scope of this paper. Comparing all publicly available implementations and characterizing them in terms of input simplicial complex,

handling of functions that are not Morse, available simplification models (pre- vs. post-simplification), output options, and memory footprint would be beneficial. Additionally, a version of the PLMSS using voxels as top-level cells in 3D could potentially speed up the computation considerably and would downsize the memory footprint even more. Also, the effect of path compression might be applicable to the computation of the MS complex, so additional research in integrating it might be of interest to achieve better runtime efficiency of MS complex implementations.

7 Discussion and Conclusion

In this thesis, the visual analytics (VA) process by Keim et al. [136] has been extended to support uncertainty in every process component. It defines a uncertainty-aware visual analytics (UAVA) process that describes the quantification and handling of uncertainty in every component and transition. The guideline enables the creation of UAVA applications by giving explicit steps in the design process. Multiple example applications provide easy-to-follow instances of guideline realizations, supplying detailed descriptions of every step. Additionally, multiple references to additional information for specific application areas provide further information for application creators.

7.1 Research Questions and Goals

Two research questions that resulted in three goals (Section 1.1) showed the challenges that drove the development of this thesis. First, the need for a process that enables the inclusion of uncertainty information in the VA process has been extracted **(RQ1)**. It has been solved by defining an adapted VA process that provides a model for UAVA applications to check their compatibility **(RG1)**. To support application development with this process, necessary steps towards a UAVA application must be extracted **(RQ2)**. This has been tackled in two major steps. First, the guidelines for UAVA application creation deliver instructions to follow during application development **(RG2)**. Second, the implementation of those steps is simplified by providing real-world application examples **(RG3)**.

The VA process C of Keim et al. [136] sets the basis for the extension to systematically allow the inclusion of uncertainty, creating the UAVA process C' **(RG1)**. Therefore, existing components are adapted to hold uncertainty connected with the existing information. All connections are updated to link existing and new components together, allowing them to propagate and aggregate uncertainty throughout the process. This allows checking if a given application c is compatible with the UAVA process definition C'.

The realization can be tricky even with the available definition of a UAVA process. The guideline **(RG2)** proposes instructions to create a UAVA application, enabling readers to conceptualize applications on their own. Following the guideline maps a VA application $c \in C$ to a UAVA application $c' \in C'$. Alternatively, the guideline provides instructions to generate a new VA application first.

Additionally, example applications, like the XAI brain lesion prediction, showcase realizations of the guideline on real examples **(RG3)**, providing concrete mappings $c \mapsto c'$. Subsequently, Chapter 4 presents one of our applications in molecular stability testing, showcasing the proposed guideline by reconstructing the design process. Chapter 5 highlights utilizing various image measures to analyze brain tumors and large image databases employing uncertainty and similarity assessment. Finally, Chapter 6 proposes a general method for topological scalar field analysis, also discussing its use for UAVA applications.

7.2 Open Challenges and Future Work

This thesis presented a UAVA process and showcased application development using the provided guideline. This leads to further challenges in the field of UAVA.

Insight Uncertainty Quantification Insight is generated mainly by the conclusions drawn by the users of VA applications, complicating the quantification of uncertainty in this component of the UAVA process. So far, users can annotate the data to express their trust in the result, or the system can aggregate and propagate the uncertainty information present in the visualization and hypothesis components. Still, interdisciplinary research of VA and cognitive science could try to measure the uncertainty in the users' cognitive processes during an analysis, redeeming a more accurate quantification of insight uncertainty.

Realizing Uncertainty-aware Visual Analytics Applications The UAVA process and guidelines allow the creation of applications mapped to existing solutions in multiple post hoc analyses and application design simulations. Even though this provides a great starting point for implementing new applications, a lot can be learned from using the guidelines for creating new projects. Future applications created from these guidelines should be used to generate guidelines and frameworks tailored to the specific application area.

Uncertainty-focused Guideline The presented guideline focuses on generating a VA application first that is transformed to conform to the UAVA process. Even though this makes it applicable to existing and new applications, it might not be optimal for newly created UAVA applications. Focusing on the available uncertainty sources and their quantification as a first step might be more suitable in certain situations. This could be tested by designing new applications and learning from their challenges during the design phase.

Extended Taxonomy of Uncertainties The taxonomy of uncertainties in VA by Gillmann et al. [81] is a great starting point for identifying uncertainty sources in an application design. Still, the taxonomy is given without any claim to comprehensiveness and can be extended by including insights taken from existing and future applications. Its depth is currently restricted to a very general description of uncertainties in VA and could be extended for various application areas and data types, making it more versatile in specific application areas.

Frameworks with Ready-to-use Uncertainty-aware Visual Analytics functions Many transformations in the UAVA process can be accomplished by existing methodologies. Still, they must be reimplemented or compiled from multiple sources. Examples are the determination and description of uncertainty-aware (UA) datasets, adaptation of preprocessing and hypothesis generation approaches, and provenance generation. In this context, frameworks and libraries that provide explicit UAVA process steps that can be standardized would significantly contribute to the VA community. Gillmann et al. [85] provided a survey on uncertainty-awareness in open-source visualization solutions, which can be a great starting point for creating a UAVA framework. Also, Correa et al. [46] provides a framework that employs Gaussian mixture models to represent uncertainty in VA.

State of the Art in Uncertainty-aware Visual Analytics Even though many UAVA applications have already been published, a state-of-the-art report summarizing all related approaches, frameworks, and guidelines has not been created so far. Such a report would pose a great starting point for researchers when creating their own UAVA application, process definition, or guideline. Also, further open challenges in the UAVA field could be identified. The presented research and related works pose a great starting point for creating such a report.

Uncertainty-aware Visual Analytics Planning Tool Providing guidelines in the form of text can already be a great way to inform the reader about possibilities in the design of a UAVA application. Still, reading and mapping the guidelines to the present project can be a tedious process. Here, an online and open-source UAVA planning tool would be a great extension to adapt a design interactively. This would combine the process definition, the taxonomy of uncertainties [81], and the guidelines into a tool that could be extended by implementing lessons learned from existing and newly created UAVA applications.

Bibliography

[1] C. C. Aggarwal and C. K. Reddy. *Data clustering: algorithms and applications*. 1st. Chapman & Hall, Aug. 29, 2013.

[2] J. Ahrens, S. Jourdain, P. O'Leary, J. Patchett, D. H. Rogers, and M. Petersen. "An Image-Based Approach to Extreme Scale in Situ Visualization and Analysis". In: *SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. International Conference for High Performance Computing, Networking, Storage and Analysis. Nov. 2014, pp. 424–434.

[3] H. C. Andrews. "Digital image restoration: A survey". In: *Computer* 7.5 (May 1974), pp. 36–45.

[4] K. Andrews, W. Putz, and A. Nussbaumer. "The Hierarchical Visualisation System (HVS)". In: *2007 11th International Conference Information Visualization (IV '07)*. 2007 11th International Conference Information Visualization (IV '07). July 2007, pp. 257–262.

[5] N. Andrienko, T. Lammarsch, G. Andrienko, G. Fuchs, D. Keim, S. Miksch, and A. Rind. "Viewing Visual Analytics as Model Building". In: *Computer Graphics Forum* 37.6 (2018), pp. 275–299.

[6] U. Artmann. "Image quality assessment using the dead leaves target: experience with the latest approach and further investigations". In: *Digital Photography XI*. Digital Photography XI. Vol. 9404. International Society for Optics and Photonics, Feb. 27, 2015.

[7] I. Avcibas, B. Sankur, and K. Sayood. "Statistical evaluation of image quality measures". In: *Journal of Electronic Imaging* 11.2 (Apr. 2002), pp. 206–223.

[8] D. Balasubramaniam and E. A. Komives. "Hydrogen-exchange mass spectrometry for the study of intrinsic disorder in proteins". In: *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics* 1834.6 (June 1, 2013), pp. 1202–1209.

[9] T. Banchoff. "Critical points and curvature for embedded polyhedra". In: *Journal of Differential Geometry* 1.3-4 (1967), pp. 245–256.

[10] T. F. Banchoff. "Critical points and curvature for embedded polyhedral surfaces". In: *The American Mathematical Monthly* 77.5 (1970), pp. 475–485.

[11] R. Barber, W. Equitz, W. Flickner, W. Niblack, D. Petkovic, and P. Yanker. "Efficient query by image content for very large image databases". In: *Digest of Papers. Compcon Spring*. Digest of Papers. Compcon Spring. Feb. 1993, pp. 17–19.

[12]B. B. Bederson, B. Shneiderman, and M. Wattenberg. “Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies”. In: *ACM Transactions on Graphics* 21.4 (Oct. 1, 2002), pp. 833–854.

[13]H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. “The Protein Data Bank”. In: *Nucleic Acids Research* 28.1 (Jan. 1, 2000), pp. 235–242.

[14]S. Beucher and C. Lantuejoul. “Use of watersheds in contour detection. Int”. In: *Workshop on Image Processing, CCETT/IRISA, Rennes, France*. 1979.

[15]S. Beucher. “Watersheds of functions and picture segmentation”. In: *ICASSP’82. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 7. IEEE. 1982, pp. 1928–1931.

[16]H. Bhatia, A. G. Gyulassy, V. Lordi, J. E. Pask, V. Pascucci, and P.-T. Bremer. “TopoMS: Comprehensive topological exploration for molecular and condensed-matter systems”. In: *Journal of computational chemistry* 39.16 (2018), pp. 936–952.

[17]U. Bhatt, J. Antorán, Y. Zhang, Q. V. Liao, et al. “Uncertainty as a Form of Transparency: Measuring, Communicating, and Using Uncertainty”. In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. AIES ’21. New York, NY, USA: Association for Computing Machinery, July 30, 2021, pp. 401–413.

[18]T. Bin Masood, J. Budin, M. Falk, G. Favelier, et al. “An Overview of the Topology ToolKit”. In: *TopoInVis*. 2019.

[19]G.-P. Bonneau, H.-C. Hege, C. R. Johnson, M. M. Oliveira, K. Potter, P. Rheingans, and T. Schultz. “Overview and State-of-the-Art of Uncertainty Visualization”. In: *Scientific Visualization: Uncertainty, Multifield, Biomedical, and Scalable Visualization*. Ed. by C. D. Hansen, M. Chen, C. R. Johnson, A. E. Kaufman, and H. Hagen. Mathematics and Visualization. London: Springer, 2014, pp. 3–27.

[20]P. Booth, N. Gibbins, and S. Galanis. “Design Spaces in Visual Analytics Based on Goals: Analytical Behaviour, Exploratory Investigation, Information Design & Perceptual Tasks”. In: *Proceedings of the 52nd Hawaii International Conference on System Sciences*. University of Hawai’i at Manoa, Jan. 8, 2019.

[21]C. Bors, J. Bernard, M. Bögl, T. Gschwandtner, J. Kohlhammer, and S. Miksch. “Quantifying Uncertainty in Multivariate Time Series Pre-Processing”. In: *EuroVA 2019, EuroVis Workshop on Visual Analytics*. EuroVA 2019, EuroVis Workshop on Visual Analytics. 2019.

[22]M. Boumans, G. Hon, and A. C. Petersen. *Error and Uncertainty in Scientific Practice*. Routledge, Oct. 6, 2015. 264 pp.

[23]A. K. Boyat and B. K. Joshi. “A Review Paper: Noise Models in Digital Image Processing”. In: *Signal & Image Processing* 6.2 (May 13, 2015), p. 63.

[24]P.-T. Bremer, B. Hamann, H. Edelsbrunner, and V. Pascucci. “A topological hierarchy for functions on triangulated surfaces”. In: *IEEE Transactions on Visualization and Computer Graphics* 10.4 (2004), pp. 385–396.

[25]P. Brivio, M. Tarini, and P. Cignoni. “Browsing Large Image Datasets through Voronoi Diagrams”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (Nov. 2010), pp. 1261–1270.

[26]D. Brodbeck, R. Mazza, and D. Lalanne. “Interactive Visualization - A Survey”. In: *Human Machine Interaction: Research Results of the MMI Program*. Ed. by D. Lalanne and J. Kohlas. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2009, pp. 27–46.

[27]K. Brodlie, R. Allendes Osorio, and A. Lopes. “A Review of Uncertainty in Data Visualization”. In: *Expanding the Frontiers of Visual Analytics and Visualization*. Ed. by J. Dill, R. Earnshaw, D. Kasik, J. Vince, and P. C. Wong. London: Springer, 2012, pp. 81–109.

[28]E. Buxbaum. *Fundamentals of Protein Structure and Function*. 2nd ed. Springer International Publishing Switzerland, 2015.

[29]*Bynder*. 2021. URL: `https://www.bynder.com/`.

[30]G. Cai and S. Mahadevan. “Big Data Analytics in Uncertainty Quantification: Application to Structural Diagnosis and Prognosis”. In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 4.1 (Mar. 1, 2018), p. 04018003.

[31]S. Cai, B. Gallina, D. Nyström, and C. Seceleanu. “Data aggregation processes: a survey, a taxonomy, and design guidelines”. In: *Computing* 101.10 (Oct. 1, 2019), pp. 1397–1429.

[32]J. Camargo and F. González. “Visualization, Summarization and Exploration of Large Collections of Images: State Of The Art”. In: *Latin-American Conference On Networked and Electronic Media*. Latin-American Conference On Networked and Electronic Media (LACNEM). 2009.

[33]H. Carr, T. Moller, and J. Snoeyink. “Artifacts caused by simplicial subdivision”. In: *IEEE Transactions on Visualization and Computer Graphics* 12.2 (2006), pp. 231–242.

[34]O. Carugo. “How large B-factors can be in protein crystal structures”. In: *BMC Bioinformatics* 19.1 (Feb. 23, 2018), p. 9.

[35]N. Chamoli, S. Kukreja, and M. Semwal. “Survey and Comparative Analysis on Entropy Usage for Several Applications in Computer Vision”. In: *International Journal of Computer Applications* 97.16 (2014).

[36]Y.-H. Chan. “Uncertainty-Aware Visual Analytics in Sensitivity Scatterplots”. PhD thesis. University of California, 2014.

[37]H. Choi and S. Choi. “Robust kernel Isomap”. In: *Pattern Recognition* 40.3 (Mar. 1, 2007), pp. 853–862.

[38]M. Chouchane, A. Rucci, and A. A. Franco. “A Versatile and Efficient Voxelization-Based Meshing Algorithm of Multiple Phases”. In: *ACS Omega* 4.6 (2019), pp. 11141–11144.

[39]H.-K. Chung, B. J. Braams, K. Bartschat, A. G. Császár, G. W. F. Drake, T. Kirchner, V. Kokoouline, and J. Tennyson. “Uncertainty estimates for theoretical atomic and molecular data”. In: *Journal of Physics D: Applied Physics* 49.36 (Aug. 2016).

[40]K. Clark, B. Vendt, K. Smith, J. Freymann, et al. “The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository”. In: *Journal of Digital Imaging* 26.6 (Dec. 2013), pp. 1045–1057.

[41]A. A. Clifford. *Multivariate Error Analysis: A Handbook of Error Propagation and Calculation in Many-Parameter Systems*. Wiley, 1973.

[42]M. Conlen, S. Stalla, C. Jin, M. Hendrie, H. Mushkin, S. Lombeyda, and S. Davidoff. “Towards Design Principles for Visual Analytics in Operations Contexts”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI ’18. New York, NY, USA: Association for Computing Machinery, Apr. 19, 2018, pp. 1–7.

[43]A. W. Cook, W. Cabot, and P. L. Miller. “The mixing transition in Rayleigh-Taylor instability”. In: *Journal of Fluid Mechanics* 511 (2004), pp. 333–362.

[44]K. A. Cook and J. J. Thomas. *Illuminating the path: The research and development agenda for visual analytics*. IEEE Computer Society, 2005.

[45]C. C. Cornilescu, F. Bouamr, X. Yao, C. Carter, and N. Tjandra. “Structural analysis of the N-terminal domain of the human T-cell leukemia virus capsid protein”. In: *Journal of Molecular Biology* 306.4 (Mar. 2, 2001), pp. 783–797.

[46]C. D. Correa, Y.-H. Chan, and K.-L. Ma. “A framework for uncertainty-aware visual analytics”. In: *2009 IEEE Symposium on Visual Analytics Science and Technology*. 2009 IEEE Symposium on Visual Analytics Science and Technology. Oct. 2009, pp. 51–58.

[47]M. V. C. Coutinho, J. S. Redford, B. A. Church, A. C. Zakrzewski, J. J. Couchman, and J. D. Smith. “The interplay between uncertainty monitoring and working memory: Can metacognition become automatic?” In: *Memory & Cognition* 43.7 (Oct. 2015), pp. 990–1006.

[48]E. Danovaro, L. De Floriani, P. Magillo, M. M. Mesmoudi, and E. Puppo. “Morphology-driven simplification and multiresolution modeling of terrains”. In: *Proceedings of the 11th ACM international symposium on Advances in geographic information systems*. 2003, pp. 63–70.

[49]E. Danovaro, L. D. Floriani, and M. M. Mesmoudi. “Topological analysis and characterization of discrete scalar fields”. In: *Geometry, Morphology, and Computational Imaging*. Springer, 2003, pp. 386–402.

[50]A. Dasgupta and R. Kosara. “The Need for Information Loss Metrics in Visualization”. In: *Workshop on The Role of Theory in Information Visualization, VisWeek*. Workshop on The Role of Theory in Information Visualization, VisWeek. Oct. 2010.

[51]Data Science at Scale team at Los Alamos National Laboratory. *Cinema Project*. 2019. URL: `https://cinemascience.github.io/`.

[52]L. De Floriani, U. Fugacci, F. Iuricich, and P. Magillo. “Morse complexes for shape segmentation and homological analysis: discrete models and algorithms”. In: *Computer Graphics Forum*. Vol. 34. Wiley Online Library. 2015, pp. 761–785.

[53]J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009 IEEE Conference on Computer Vision and Pattern Recognition. June 2009, pp. 248–255.

[54]E. Deryusheva, E. Nemashkalova, M. Galloux, C.-A. Richard, J.-F. Eléouët, D. Kovacs, K. V. Belle, P. Tompa, V. Uversky, and S. Permyakov. “Does Intrinsic Disorder in Proteins Favor Their Interaction with Lipids?” In: *Proteomics* 19.6 (2019).

[55]J. L. Devore. *Probability and Statistics for Engineering and the Sciences*. Cengage Learning, Jan. 1, 2015. 797 pp.

[56]M. E. Diamond. “Perceptual uncertainty”. In: *PLOS Biology* 17.8 (Aug. 27, 2019).

[57]A. Doi and A. Koide. “An efficient method of triangulating equi-valued surfaces by using tetrahedral cells”. In: *IEICE TRANSACTIONS on Information and Systems* 74.1 (1991), pp. 214–224.

[58]S. Dong, P. Bremer, M. Garland, V. Pascucci, and J. C. Hart. “Spectral surface quadrangulation”. In: *ACM Trans. Graph.* 25.3 (2006).

[59]H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2009.

[60]H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. “Morse-Smale complexes for piecewise linear 3-manifolds”. In: *Proceedings of the nineteenth annual symposium on Computational geometry*. 2003, pp. 361–370.

[61]H. Edelsbrunner, J. Harer, and A. Zomorodian. “Hierarchical Morse complexes for piecewise linear 2-manifolds”. In: *Proceedings of the seventeenth annual symposium on Computational geometry*. 2001, pp. 70–79.

[62]H. Edelsbrunner and E. P. Mücke. “Simulation of Simplicity: A Technique to Cope with Degenerate Cases in Geometric Algorithms”. In: *ACM Transactions on Graphics (tog)* 9.1 (1990), pp. 66–104.

[63]B. Enke and T. Graeber. *Cognitive Uncertainty*. Nov. 2019.

[64]P. Federico, A. Amor-Amorós, and S. Miksch. “A Nested Workflow Model for Visual Analytics Design and Validation”. In: *Proceedings of the Sixth Workshop on Beyond Time and Errors on Novel Evaluation Methods for Visualization*. BELIV ’16. New York, NY, USA: Association for Computing Machinery, Oct. 24, 2016, pp. 104–111.

[65]R. Fellegara, F. Luricich, L. De Floriani, and K. Weiss. “Efficient computation and simplification of discrete Morse decompositions on triangulated terrains”. In: *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 2014, pp. 223–232.

[66]J. M. Fernández Güell. “How to approach urban complexity, diversity and uncertainty when involving stakeholders into the planning process”. In: *AESOP 2017 Congress. Book of proceedings*. Lisbon AESOP Annual Congress 2017. Lisboa, Portugal: E.T.S. Arquitectura (UPM), 2017, pp. 2780–2791.

[67]J. R. Fienup. “Invariant error metrics for image reconstruction”. In: *Applied optics* 36.32 (1997), pp. 8352–8357.

[68]R. Forman. “A user’s guide to discrete Morse theory.” In: *Séminaire Lotharingien de Combinatoire [electronic only]* 48 (2002), B48c–35.

[69]R. Forman. “Morse theory for cell complexes”. In: *Advances in mathematics* 134.1 (1998), pp. 90–145.

[70]A. U. Frank. “Incompleteness, Error, Approximation, and Uncertainty: an Ontological Approach to Data Quality”. In: *Geographic Uncertainty in Environmental Security (NATO Science for Peace and Security Series / NATO Science for Peace and Security Series C: Environmental Security)*. NATO Advance Research Workshop 2007. Springer, 2007.

[71]P. H. Furfey. “A note on Lefever’s" standard deviational ellipse"”. In: *American Journal of Sociology* 33.1 (1927), pp. 94–98.

[72]Y. Gabrielyan, V. Yeghiazaryan, and I. Voiculescu. “Parallel Partitioning: Path Reducing and Union–Find Based Watershed for the GPU”. In: *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2022, pp. 1501–1505.

[73]Y. Gal and Z. Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In: *Proceedings of The 33rd International Conference on Machine Learning*. International Conference on Machine Learning. Vol. 48. PMLR, June 11, 2016, pp. 1050–1059.

[74]T. Gerrits, C. Rössl, and H. Theisel. “Towards Glyphs for Uncertain Symmetric Second-Order Tensors”. In: *Computer Graphics Forum* 38.3 (2019), pp. 325–336.

[75]C. Ghilani. “Statistics and adjustments explained part 3: Error propagation”. In: *Surveying and Land Information Science* 64.1 (Mar. 1, 2004), pp. 29–33.

[76]G. Gigerenzer. “The Psychology of Good Judgment: Frequency Formats and Simple Algorithms”. In: *Medical Decision Making* 16.3 (Aug. 1, 1996), pp. 273–280.

[77]C. Gillmann. “Image Processing under Uncertainty”. PhD thesis. Kaiserslautern, Germany: Technische Universität Kaiserslautern, 2019. 237 pp.

[78]C. Gillmann, P. Arbeláez, J. Peñaloza, H. Hagen, and T. Wischgoll. “Intuitive Error Space Exploration of Medical Image Data in Clinical Daily Routine”. In: *Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Short Papers* (June 12, 2017).

[79]C. Gillmann, H. Leitte, T. Wischgoll, and H. Hagen. “From theory to usage: Requirements for successful visualizations in applications”. In: *IEEE VIS, Creation, Curation, Critique and Conditioning of Principles and Guidelines in Visualization (C4PGV)*. IEEE Visualization Conference (VIS)-C4PGV Workshop. Vol. 5. 2016, p. 4.

[80]C. Gillmann, R. G. C. Maack, T. Post, T. Wischgoll, and H. Hagen. “An Uncertainty-aware Workflow for Keyhole Surgery Planning using Hierarchical Image Semantics”. In: *Visual Informatics*. Proceedings of PacificVAST 2018 2.1 (2018), pp. 26–36.

[81] C. Gillmann, R. G. C. Maack, F. Raith, J. F. Pérez, and G. Scheuermann. “A Taxonomy of Uncertainty Events in Visual Analytics”. In: *IEEE Computer Graphics and Applications* 43.5 (Sept. 2023), pp. 62–71.

[82] C. Gillmann, L. Peter, C. Schmidt, D. Saur, and G. Scheuermann. “Visualizing Multimodal Deep Learning for Lesion Prediction”. In: *IEEE Computer Graphics and Applications* 41.5 (Sept. 2021), pp. 90–98.

[83] C. Gillmann, D. Saur, T. Wischgoll, and G. Scheuermann. “Uncertainty-aware Visualization in Medical Imaging - A Survey”. In: *Computer Graphics Forum* 40.3 (2021), pp. 665–689.

[84] C. Gillmann, N. N. Smit, E. Gröller, B. Preim, A. Vilanova, and T. Wischgoll. “Ten Open Challenges in Medical Visualization”. In: *IEEE Computer Graphics and Applications* 41.5 (Sept. 2021), pp. 7–15.

[85] C. Gillmann, T. Wischgoll, and H. Hagen. “Uncertainty-Awareness in Open Source Visualization Solutions”. In: *IEEE VIS, Vis in Practice, 2016*. 2016, p. 5.

[86] C. Gillmann, T. Wischgoll, B. Hamann, and J. Ahrens. “Modeling and Visualization of Uncertainty-Aware Geometry Using Multi-variate Normal Distributions”. In: *2018 IEEE Pacific Visualization Symposium (PacificVis)*. 2018 IEEE Pacific Visualization Symposium (PacificVis). IEEE, Apr. 2018, pp. 106–110.

[87] C. Gillmann, T. Wischgoll, J. Hernandez, and H. Hagen. “Teaching Image Processing and Visualization Principles to Medicine Students”. In: *Computer Science and Engineering Faculty Publications* (Oct. 1, 2017).

[88] M. Gleicher, M. Riveiro, T. Landesberger, O. Deussen, R. Chang, C. Gillmann, and T.-M. Rhyne. “A Problem Space for Designing Visualizations”. In: *IEEE computer graphics and applications* 43 (July 1, 2023), pp. 111–120.

[89] J. A. G. Gómez. *John Alexis Guerra Gómez’s Homepage*. 2019. URL: `https://johnguerra.co/`.

[90] A. Gomi and T. Itoh. “A personal photograph browser for life log analysis based on location, time, and person”. In: *Proceedings of the 2011 ACM Symposium on Applied Computing*. SAC ’11. New York, NY, USA: Association for Computing Machinery, Mar. 21, 2011, pp. 1245–1251.

[91] J. Gong. “Clarifying the Standard Deviational Ellipse”. In: *Geographical Analysis* 34.2 (2002), pp. 155–167.

[92] J. Görtler. “Quantitative Methods for Uncertainty Visualization”. PhD thesis. Konstanz: Universität Konstanz, 2021.

[93] J. Görtler, T. Spinner, D. Streeb, D. Weiskopf, and O. Deussen. “Uncertainty-Aware Principal Component Analysis”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (Jan. 2020), pp. 822–831.

[94] R. Green and C. Zardecki. *Guide to Understanding PDB Data*. PDB-101 Molecular explorations through biology and medicine. 2019. URL: `https://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/dealing-with-coordinates` (visited on 04/12/2021).

[95]T. M. Green, W. Ribarsky, and B. Fisher. “Building and Applying a Human Cognition Model for Visual Analytics”. In: *Information Visualization* 8.1 (Jan. 1, 2009), pp. 1–13.

[96]H. Griethe and H. Schumann. “The Visualization of Uncertain Data: Methods and Problems”. In: *Proceedings of SimVis ’06*. SCS Publishing House, 2006, pp. 143–156.

[97]N. Guex and M. C. Peitsch. “SWISS-MODEL and the Swiss-Pdb Viewer: An environment for comparative protein modeling”. In: *ELECTROPHORESIS* 18.15 (1997), pp. 2714–2723.

[98]P. Guillou, J. Vidal, and J. Tierny. “Discrete Morse Sandwich: Fast Computation of Persistence Diagrams for Scalar Data – An Algorithm and A Benchmark”. In: *ArXiv e-prints* (2022).

[99]D. Günther, R. A. Boto, J. Contreras-Garcia, J.-P. Piquemal, and J. Tierny. “Characterizing molecular interactions in chemical systems”. In: *IEEE transactions on visualization and computer graphics* 20.12 (2014), pp. 2476–2485.

[100]D. Günther, J. Reininghaus, H. Wagner, and I. Hotz. “Efficient computation of 3D Morse–Smale complexes and persistent homology using discrete Morse theory”. In: *The Visual Computer* 28.10 (2012), pp. 959–969.

[101]A. Gyulassy, P. Bremer, and V. Pascucci. “Shared-Memory Parallel Computation of Morse-Smale Complexes with Improved Accuracy”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2018), pp. 1183–1192.

[102]A. Gyulassy, P.-T. Bremer, B. Hamann, and V. Pascucci. “A practical approach to Morse-Smale complex computation: Scalability and generality”. In: *IEEE Transactions on Visualization and Computer Graphics* 14.6 (2008), pp. 1619–1626.

[103]A. Gyulassy, P.-T. Bremer, and V. Pascucci. “Computing Morse-Smale complexes with accurate geometry”. In: *IEEE transactions on visualization and computer graphics* 18.12 (2012), pp. 2014–2022.

[104]A. Gyulassy, M. Duchaineau, V. Natarajan, V. Pascucci, E. Bringa, A. Higginbotham, and B. Hamann. “Topologically clean distance fields”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1432–1439.

[105]A. Gyulassy, D. Günther, J. A. Levine, J. Tierny, and V. Pascucci. “Conforming morse-smale complexes”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2595–2603.

[106]A. Gyulassy, A. Knoll, K. C. Lau, B. Wang, P.-T. Bremer, M. E. Papka, L. A. Curtiss, and V. Pascucci. “Interstitial and interlayer ion diffusion geometry extraction in graphitic nanosphere battery materials”. In: *IEEE transactions on visualization and computer graphics* 22.1 (2015), pp. 916–925.

[107]A. Gyulassy, V. Natarajan, V. Pascucci, P. Bremer, and B. Hamann. “A Topological Approach to Simplification of Three-Dimensional Scalar Functions”. In: *IEEE Trans. Vis. Comput. Graph.* 12.4 (2006).

[108]A. Gyulassy, V. Natarajan, V. Pascucci, and B. Hamann. “Efficient computation of Morse-Smale complexes for three-dimensional scalar functions”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1440–1447.

[109]A. Gyulassy, V. Pascucci, T. Peterka, and R. Ross. “The parallel computation of Morse-Smale complexes”. In: *2012 IEEE 26th International Parallel and Distributed Processing Symposium*. IEEE. 2012, pp. 484–495.

[110]D. Hägele, C. Schulz, C. Beschle, H. Booth, M. Butt, A. Barth, O. Deussen, and D. Weiskopf. “Uncertainty visualization: Fundamentals and recent developments”. In: *Information Technology* 64.4 (Aug. 1, 2022), pp. 121–132.

[111]J. Y. Halpern. *Reasoning about Uncertainty, second edition*. MIT Press, Apr. 7, 2017. 505 pp.

[112]T. M. Hegel, S. A. Cushman, J. Evans, and F. Huettmann. “Current State of the Art for Statistical Modelling of Species Distributions”. In: *Spatial Complexity, Informatics, and Wildlife Conservation*. Ed. by S. A. Cushman and F. Huettmann. Tokyo: Springer Japan, 2010, pp. 273–311.

[113]C. Heine, H. Leitte, M. Hlawitschka, F. Iuricich, L. De Floriani, G. Scheuermann, H. Hagen, and C. Garth. “A survey of topology-based methods in visualization”. In: *Computer Graphics Forum*. Vol. 35. Wiley Online Library. 2016, pp. 643–667.

[114]M. Herschel, R. Diestelkämper, and H. Ben Lahmar. “A survey on provenance: What for? What form? What from?” In: *The VLDB Journal* 26.6 (Dec. 1, 2017), pp. 881–906.

[115]G. Hinton and S. Roweis. “Stochastic Neighbor Embedding”. In: *NIPS’02: Proceedings of the 15th International Conference on Neural Information Processing Systems* 15 (2002), pp. 857–864.

[116]M. Höferlin, B. Höferlin, D. Weiskopf, and G. Heidemann. “Uncertainty-aware video visual analytics of tracked moving objects”. In: *Journal of Spatial Information Science* 2011.2 (2011), pp. 87–117.

[117]U. Homberg, D. Baum, A. Wiebel, S. Prohaska, and H.-C. Hege. “Definition, extraction, and validation of pore structures in porous materials”. In: *Topological methods in data analysis and visualization III*. Springer, 2014, pp. 235–248.

[118]Z. Huang, Y. Lu, E. A. Mack, W. Chen, and R. Maciejewski. “Exploring the Sensitivity of Choropleths under Attribute Uncertainty”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.8 (Aug. 2020), pp. 2576–2590.

[119]J. Hullman, X. Qiao, M. Correll, A. Kale, and M. Kay. “In Pursuit of Error: A Survey of Uncertainty Visualization Evaluation”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019), pp. 903–913.

[120]L. M. Iakoucheva, C. J. Brown, J. D. Lawson, Z. Obradović, and A. K. Dunker. “Intrinsic Disorder in Cell-signaling and Cancer-associated Proteins”. In: *Journal of Molecular Biology* 323.3 (Oct. 25, 2002), pp. 573–584.

[121]L. M. Iakoucheva, P. Radivojac, C. J. Brown, T. R. O’Connor, J. G. Sikes, Z. Obradovic, and A. K. Dunker. “The importance of intrinsic disorder for protein phosphorylation”. In: *Nucleic Acids Research* 32.3 (Feb. 1, 2004), pp. 1037–1049.

[122]I. Irum, M. Sharif, M. Raza, and S. Mohsin. “A Nonlinear Hybrid Filter for Salt & Pepper Noise Removal from Color Images”. In: *Journal of Applied Research and Technology* 13.1 (Feb. 1, 2015), pp. 79–85.

[123]S. Jänicke, A. Geßner, M. Büchler, and G. Scheuermann. “Visualizations for Text Re-use”. In: *2014 International Conference on Information Visualization Theory and Applications (IVAPP)*. 2014 International Conference on Information Visualization Theory and Applications (IVAPP). IEEE, Jan. 2014, pp. 59–70.

[124]L. JayaV and R. Gopikakumari. “IEM: A New Image Enhancement Metric for Contrast and Sharpness Measurements”. In: *International Journal of Computer Applications* 79.9 (2013).

[125]A. Jena, U. Engelke, T. Dwyer, V. Raiamanickam, and C. Paris. “Uncertainty Visualisation: An Interactive Visual Survey”. In: *2020 IEEE Pacific Visualization Symposium (PacificVis)*. 2020 IEEE Pacific Visualization Symposium (PacificVis). June 2020, pp. 201–205.

[126]C. Johnson. “Top scientific visualization research problems”. In: *IEEE Computer Graphics and Applications* 24.4 (July 2004), pp. 13–17.

[127]D. E. Johnson, B. Xue, M. D. Sickmeier, J. Meng, M. S. Cortese, C. J. Oldfield, T. Le Gall, A. K. Dunker, and V. N. Uversky. “High-throughput characterization of intrinsic disorder in proteins from the Protein Structure Initiative”. In: *Journal of Structural Biology* 180.1 (Oct. 1, 2012), pp. 201–215.

[128]M. W. Jones. “3D Distance from a Point to a Triangle”. In: *Department of Computer Science, University of Wales Swansea Technical Report CSR-5* (1995), p. 5.

[129]D. Kahneman and S. Frederick. “Representativeness revisited: Attribute substitution in intuitive judgment”. In: *Heuristics of Intuitive Judgment: Extensions and Applications*. New York: Cambridge University Press, 2002, pp. 49–81.

[130]A. Kamal, P. Dhakal, A. Y. Javaid, V. K. Devabhaktuni, D. Kaur, J. Zaientz, and R. Marinier. “Recent advances and challenges in uncertainty visualization: a survey”. In: *Journal of Visualization* 24.5 (Oct. 1, 2021), pp. 861–890.

[131]S. Al-Karadaghi. *PDB File Format and Content*. Protein Structures. 2010. URL: `https://proteinstructures.com/structure/protein-databank/` (visited on 04/12/2021).

[132]A. Karami. “A Framework for Uncertainty-Aware Visual Analytics in Big Data”. In: *CEUR Workshop Proceedings* 1510 (Nov. 12, 2015). Ed. by A. Lieto, C. Battaglino, D. P. Radicioni, and M. Sanguinetti, pp. 146–155.

[133]A. Karshikoff, L. Nilsson, and R. Ladenstein. “Rigidity versus flexibility: the dilemma of understanding protein thermal stability”. In: *The FEBS Journal* 282.20 (2015), pp. 3899–3917.

[134]D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. “Visual Analytics: Definition, Process, and Challenges”. In: *Information Visualization: Human-Centered Issues and Perspectives*. Ed. by A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North. Vol. 4090. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008, pp. 154–175.

[135]D. Keim and L. Zhang. “Solving problems with visual analytics: challenges and applications”. In: *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*. i-KNOW ’11. New York, NY, USA: Association for Computing Machinery, Sept. 7, 2011, pp. 1–4.

[136]D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. “Visual Analytics: Scope and Challenges”. In: *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*. Ed. by S. J. Simoff, M. H. Böhlen, and A. Mazeika. Vol. 4404. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008, pp. 76–90.

[137]R. Khulusi, J. Kusnick, C. Meinecke, C. Gillmann, J. Focht, and S. Jänicke. “A Survey on Visualizations for Musical Data”. In: *Computer Graphics Forum* 39.6 (2020), pp. 82–110.

[138]H. King, K. Knudson, and N. Mramor. “Generating discrete Morse functions from point data”. In: *Experimental Mathematics* 14.4 (2005), pp. 435–444.

[139]C. Klein, J. Hollender, H. Bender, and G. E. Schulz. “Catalytic center of cyclodextrin glycosyltransferase derived from x-ray structure analysis combined with site-directed mutagenesis”. In: *Biochemistry* 31 (1992), p. 7.

[140]C. Klein and G. E. Schulz. “Structure of cyclodextrin glycosyltransferase refined at 2.0 Å resolution”. In: *Journal of Molecular Biology* 217.4 (Feb. 20, 1991), pp. 737–750.

[141]A. Knoll, M. K. Y. Chan, K. C. Lau, B. Liu, J. Greeley, L. Curtiss, M. Hereld, and M. E. Papka. “Uncertainty classification and visualization of molecular interfaces”. In: *International Journal for Uncertainty Quantification* 3.2 (2013).

[142]J. Kohlhammer, T. May, and M. Hoffmann. “Visual Analytics for the Strategic Decision Making Process”. In: *GeoSpatial Visual Analytics*. Ed. by R. D. Amicis, R. Stojanovic, and G. Conti. NATO Science for Peace and Security Series C: Environmental Security. Dordrecht: Springer Netherlands, 2009, pp. 299–310.

[143]B. Kozlíková, M. Krone, M. Falk, N. Lindow, M. Baaden, D. Baum, I. Viola, J. Parulek, and H.-C. Hege. “Visualization of Biomolecular Structures: State of the Art Revisited”. In: *Computer Graphics Forum* 36.8 (2017), pp. 178–204.

[144]L. P. Kozlowski and J. M. Bujnicki. “MetaDisorder: a meta-server for the prediction of intrinsic disorder in proteins”. In: *BMC Bioinformatics* 13.1 (May 24, 2012), p. 111.

[145]V. Kretzschmar, C. Gillmann, F. Günther, M. Stommel, and G. Scheuermann. “Visualization Framework for Assisting Interface Optimization of Hybrid Component Design”. In: *Vision, Modeling, and Visualization*. The Eurographics Association, Sept. 5, 2020.

[146]H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. “Empirical Studies in Information Visualization: Seven Scenarios”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.9 (Sept. 2012), pp. 1520–1536.

[147]D. Laney, A. Mascarenhas, P. Miller, and V. Pascucci. “Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities”. In: *IEEE Transactions on Visualization and Computer Graphics* 12.5 (2006), pp. 1053–1060.

[148]C. H. Lee and A. Varshney. “Representing thermal vibrations and uncertainty in molecular surfaces”. In: *Proc. SPIE 4665*. Visualization and Data Analysis 2002. International Society for Optics and Photonics, Mar. 12, 2002, pp. 80–90.

[149]J. A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Springer Science & Business Media, Oct. 31, 2007. 309 pp.

[150]D. Leffrang and O. Müller. “Should I Follow this Model? The Effect of Uncertainty Visualization on the Acceptance of Time Series Forecasts”. In: *2021 IEEE Workshop on TRust and EXpertise in Visual Analytics (TREX)*. 2021 IEEE Workshop on TRust and EXpertise in Visual Analytics (TREX). Oct. 2021, pp. 20–26.

[151]C. Leung, W. Chen, P. Kwok, and F. Chan. “Brain tumor boundary detection in MR image with generalized fuzzy operator”. In: *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*. International Conference on Image Processing. Vol. 2. Sept. 2003, pp. II–1057.

[152]S. Lewandowsky, T. Ballard, and R. D. Pancost. “Uncertainty as knowledge”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 373.2055 (Nov. 28, 2015), p. 20140462.

[153]T. Lewiner. “Critical sets in discrete Morse theories: Relating Forman and piecewise-linear approaches”. In: *Computer Aided Geometric Design* 30.6 (2013), pp. 609–621.

[154]H. Liu, X. Xie, X. Tang, Z.-W. Li, and W.-Y. Ma. “Effective browsing of web image search results”. In: *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*. MIR ’04. Association for Computing Machinery, Oct. 15, 2004, pp. 84–90.

[155]S. Liu, G. Andrienko, Y. Wu, N. Cao, L. Jiang, C. Shi, Y.-S. Wang, and S. Hong. “Steering data quality with visual analytics: The complexity challenge”. In: *Visual Informatics* 2.4 (Dec. 1, 2018), pp. 191–197.

[156]W. E. Lorensen and H. E. Cline. “Marching cubes: A high resolution 3D surface construction algorithm”. In: *ACM siggraph computer graphics* 21.4 (1987), pp. 163–169.

[157]J. Lukasczyk, G. Aldrich, M. Steptoe, G. Favelier, C. Gueunet, J. Tierny, R. Maciejewski, B. Hamann, and H. Leitte. “Viscous fingering: A topological visual analytic approach”. In: *Applied Mechanics and Materials*. Vol. 869. Trans Tech Publ. 2017, pp. 9–19.

[158]J. Lukasczyk, C. Garth, R. Maciejewski, and J. Tierny. “Localized topological simplification of scalar data”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (2020), pp. 572–582.

[159]W.-Y. Ma and B. S. Manjunath. “NeTra: A toolbox for navigating large image databases”. In: *Multimedia Systems* 7.3 (1999), pp. 184–198.

[160]R. G. C. Maack, M. L. Raymer, T. Wischgoll, H. Hagen, and C. Gillmann. “A framework for uncertainty-aware visual analytics of proteins”. In: *Computers & Graphics* 98 (Aug. 1, 2021), pp. 293–305.

[161]A. M. MacEachren. “Visual Analytics and Uncertainty: Its Not About the Data”. In: *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2015.

[162]A. M. MacEachren, R. E. Roth, J. O'Brien, B. Li, D. Swingley, and M. Gahegan. "Visual Semiotics & Uncertainty Visualization: An Empirical Study". In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (Dec. 2012), pp. 2496–2505.

[163]H. Maier and B. Tolson. "Sensitivity and Uncertainty". In: *Encyclopedia of Ecology*. Dec. 31, 2008, pp. 3214–3221.

[164]A. P. Mangan and R. T. Whitaker. "Partitioning 3D surface meshes using watershed segmentation". In: *IEEE Transactions on Visualization and Computer Graphics* 5.4 (1999), pp. 308–321.

[165]M. D. Mastrandrea, K. J. Mach, G.-K. Plattner, O. Edenhofer, T. F. Stocker, C. B. Field, K. L. Ebi, and P. R. Matschoss. "The IPCC AR5 guidance note on consistent treatment of uncertainties: a common approach across the working groups". In: *Climatic Change* 108.4 (Aug. 18, 2011), p. 675.

[166]A. McCulloch. "Sensitivity Analysis in Practice: a Guide to Assessing Scientific Models". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 168.2 (2005), pp. 466–466.

[167]F. Meng, V. Uversky, and L. Kurgan. "Computational Prediction of Intrinsic Disorder in Proteins". In: *Current Protocols in Protein Science* 88.1 (2017).

[168]F. Meyer. "Integrals, gradients and watershed lines". In: *Proc. Mathematical morphology and its applications to signal processing*. 1993, pp. 70–75.

[169]F. Meyer. "Topographic distance and watershed lines". In: *Signal processing* 38.1 (1994), pp. 113–125.

[170]F. Meyer and S. Beucher. "Morphological segmentation". In: *Journal of visual communication and image representation* 1.1 (1990), pp. 21–46.

[171]J. Milnor. *Morse Theory*. Princeton university press, 1963.

[172]T. Müller and F. Raether. "3D modelling of ceramic composites and simulation of their electrical, thermal and elastic properties". In: *Computational Materials Science* 81 (2014), pp. 205–211.

[173]L. Najman and M. Schmitt. "Definition and some properties of the watershed of a continuous function". In: *Mathematical Morphology and its applications to Signal Processing*. 1993, pp. 76–81.

[174]I. Neutelings. *Gaussian distributions & statistical tests*. `https://tikz.net/gaussians/`. 2021.

[175]G. P. Nguyen and M. Worring. "Interactive access to large image collections using similarity-based visualization". In: *Journal of Visual Languages & Computing* 19.2 (Apr. 1, 2008), pp. 203–224.

[176]G. Nielson and R. Franke. "Computing the separating surface for segmented data". In: *Proceedings. Visualization '97 (Cat. No. 97CB36155)*. 1997, pp. 229–233.

[177]G. M. Nielson and B. Hamann. *The asymptotic decider: Rosolving the ambiguity in marching cubes*. eScholarship, University of California, 1991.

[178]M. Olejniczak, A. Severo Pereira Gomes, and J. Tierny. “A topological data analysis perspective on noncovalent interactions in relativistic calculations”. In: *International Journal of Quantum Chemistry* 120.8 (2020), e26133.

[179]A. J. Olson. “Perspectives on Structural Molecular Biology Visualization: From Past to Present”. In: *Journal of Molecular Biology* 430.21 (Oct. 19, 2018), pp. 3997–4012.

[180]C. Olston and J. Mackinlay. “Visualizing data with bounded uncertainty”. In: *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.* IEEE Symposium on Information Visualization, 2002. INFOVIS 2002. Oct. 2002, pp. 37–40.

[181]L. Padilla, M. Kay, and J. Hullman. “Uncertainty Visualization”. In: *Computational Statistics in Data Science*. Wiley, 2022, pp. 405–421.

[182]L. M. Padilla, S. H. Creem-Regehr, M. Hegarty, and J. K. Stefanucci. “Decision making with visualizations: a cognitive framework across disciplines”. In: *Cognitive Research: Principles and Implications* 3.1 (July 11, 2018), p. 29.

[183]A. Pesce, S. Dewilde, L. Kiger, M. Milani, P. Ascenzi, M. C. Marden, M.-L. Van Hauwaert, J. Vanfleteren, L. Moens, and M. Bolognesi. “Very high resolution structure of a trematode hemoglobin displaying a TyrB10-TyrE7 heme distal residue pair and high oxygen affinity”. In: *Journal of Molecular Biology* 309.5 (June 22, 2001), pp. 1153–1164.

[184]T. Peterka, R. Ross, A. Gyulassy, V. Pascucci, W. Kendall, H.-W. Shen, T.-Y. Lee, and A. Chaudhuri. “Scalable parallel building blocks for custom data analysis”. In: *2011 IEEE Symposium on Large Data Analysis and Visualization*. IEEE. 2011, pp. 105–112.

[185]E. F. Pettersen, T. D. Goddard, C. C. Huang, E. C. Meng, G. S. Couch, T. I. Croll, J. H. Morris, and T. E. Ferrin. “UCSF ChimeraX: Structure visualization for researchers, educators, and developers”. In: *Protein Science* 30.1 (2021), pp. 70–82.

[186]R. Piotr. *iMol Overview*. 2007. URL: `https://www.pirx.com/iMol/overview.shtml` (visited on 04/30/2021).

[187]pixx.io GmbH. *pixx.io*. 2021. URL: `https://www.pixxio-bildverwaltung.de`.

[188]S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld. “Adaptive histogram equalization and its variations”. In: *Computer Vision, Graphics, and Image Processing* 39.3 (Sept. 1, 1987), pp. 355–368.

[189]C. Plaisant. “The challenge of information visualization evaluation”. In: *Proceedings of the working conference on Advanced visual interfaces*. Advanced visual interfaces. AVI ’04. New York, NY, USA: Association for Computing Machinery, May 25, 2004, pp. 109–116.

[190]A. Porollo and J. Meller. “Versatile annotation and publication quality visualization of protein complexes using POLYVIEW-3D”. In: *BMC Bioinformatics* 8.1 (Aug. 29, 2007), p. 316.

[191]K. Potter, P. Rosen, and C. R. Johnson. “From Quantification to Visualization: A Taxonomy of Uncertainty Visualization Approaches”. In: *Uncertainty Quantification in Scientific Computing*. Ed. by A. M. Dienstfrey and R. F. Boisvert. IFIP Advances in Information and Communication Technology. Berlin, Heidelberg: Springer, 2012, pp. 226–249.

[192]A. Preston, M. Gomov, and K.-L. Ma. “Uncertainty-Aware Visualization for Analyzing Heterogeneous Wildfire Detections”. In: *IEEE Computer Graphics and Applications* 39.5 (Sept. 2019), pp. 72–82.

[193]E. D. Ragan, A. Endert, J. Sanyal, and J. Chen. “Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (Jan. 2016), pp. 31–40.

[194]F. Raith, G. Scheuermann, and C. Gillmann. “Uncertainty-aware Detection and Visualization of Ocean Eddies in Ensemble Flow Fields - A Case Study of the Red Sea”. In: Workshop on Visualisation in Environmental Sciences (EnvirVis). The Eurographics Association, 2021.

[195]G. N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan. “Stereochemistry of polypeptide chain configurations”. In: *Journal of Molecular Biology* 7 (July 1963), pp. 95–99.

[196]C. Ramakrishnan and G. N. Ramachandran. “Stereochemical Criteria for Polypeptide and Protein Chain Conformations: II. Allowed Conformations for a Pair of Peptide Units”. In: *Biophysical Journal* 5.6 (Nov. 1, 1965), pp. 909–933.

[197]M. Rasheed, N. Clement, A. Bhowmick, and C. L. Bajaj. “Statistical Framework for Uncertainty Quantification in Computational Molecular Modeling”. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 16.4 (July 2019), pp. 1154–1167.

[198]R. Rebba and S. Mahadevan. “Computational methods for model reliability assessment”. In: *Reliability Engineering & System Safety* 93.8 (Aug. 1, 2008), pp. 1197–1207.

[199]P. Rheingans and S. Joshi. “Visualization of Molecules with Positional Uncertainty”. In: *Data Visualization ’99. Eurographics*. Ed. by E. Gröller, H. Löffelmann, and W. Ribarsky. Springer, 1999, pp. 299–306.

[200]V. Robins, P. J. Wood, and A. P. Sheppard. “Theory and algorithms for constructing discrete Morse complexes from grayscale digital images”. In: *IEEE Transactions on pattern analysis and machine intelligence* 33.8 (2011), pp. 1646–1658.

[201]S. Rocha Renato, Dorn, Amelie, Piringer, Barbara, and Wandl-Vogt, Eveline. “Towards A Taxonomy of Uncertainties: Analysing Sources of Spatio-Temporal Uncertainty on the Example of Non-Standard German Corpora”. In: *Informatics* 6.3 (Sept. 1, 2019), p. 34.

[202]S. T. Roweis and L. K. Saul. “Nonlinear Dimensionality Reduction by Locally Linear Embedding”. In: *Science* 290.5500 (Dec. 22, 2000), pp. 2323–2326.

[203]D. Sacha, H. Senaratne, B. C. Kwon, G. Ellis, and D. A. Keim. “The Role of Uncertainty, Awareness, and Trust in Visual Analytics”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (Jan. 2016), pp. 240–249.

[204]D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim. “Knowledge Generation Model for Visual Analytics”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (Dec. 2014), pp. 1604–1613.

[205]J. Sammon. “A Nonlinear Mapping for Data Structure Analysis”. In: *IEEE Transactions on Computers* C-18.5 (May 1969), pp. 401–409.

[206]V. Sasisekharan. “Stereochemical criteria for polypeptide and protein structures”. In: *Collagen, Madras, India. Wiley* (1962), pp. 39–78.

[207]L. Scarpace, T. Mikkelsen, S. Cha, S. Rao, et al. *The Cancer Genome Atlas Glioblastoma Multiforme Collection (TCGA-GBM)*. In collab. with TCIA Team. Version 4. 2016.

[208]I. Scholl, T. Aach, T. M. Deserno, and T. Kuhlen. “Challenges of medical image processing”. In: *Computer Science - Research and Development* 26.1 (Feb. 1, 2011), pp. 5–13.

[209]W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit–An Object-Oriented Approach To 3D Graphics*. 4th ed. Kitware, Inc., 2006.

[210]C. Schulz. “Uncertainty-aware visualization techniques”. doctoralThesis. 2021.

[211]C. Schulz, K. Schatz, M. Krone, M. Braun, T. Ertl, and D. Weiskopf. “Uncertainty Visualization for Secondary Structures of Proteins”. In: *2018 IEEE Pacific Visualization Symposium (PacificVis)*. 2018 IEEE Pacific Visualization Symposium (PacificVis). IEEE, Apr. 2018, pp. 96–105.

[212]R. Seidel and M. Sharir. “Top-Down Analysis of Path Compression”. In: *SIAM Journal on Computing* 34.3 (2005), pp. 515–525.

[213]H. Senaratne. “Uncertainty-aware Visual Analytics for Spatio-temporal Data Exploration”. PhD thesis. University of Konstanz, 2017.

[214]N. Shivashankar and V. Natarajan. “Parallel Computation of 3D Morse-Smale Complexes”. In: *Computer Graphics Forum* 31 (2012), pp. 965–974.

[215]N. Shivashankar, M. Senthilnathan, and V. Natarajan. “Parallel computation of 2D Morse-Smale complexes”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.10 (2012), pp. 1757–1770.

[216]M.-L. Shyu, C. Haruechaiyasak, S.-C. Chen, and K. Premaratne. “Mining Association Rules with Uncertain Item Relationships”. In: *Computers and Industrial Engineering* 34.1 (Sept. 6, 2002), pp. 3–20.

[217]R. Skånberg, M. Falk, M. Linares, A. Ynnerman, and I. Hotz. “Tracking Internal Frames of Reference for Consistent Molecular Distribution Functions”. In: *IEEE Transactions on Visualization and Computer Graphics* (2021), pp. 1–1.

[218]P. Soille. *Morphological Image Analysis Principles and Applications*. Springer, 2004.

[219]C. O. S. Sorzano, J. Vargas, and A. P. Montano. *A survey of dimensionality reduction techniques*. Mar. 12, 2014.

[220]T. Sousbie. "The persistent cosmic web and its filamentary structure–I. Theory and implementation". In: *Monthly Notices of the Royal Astronomical Society* 414.1 (2011), pp. 350–383.

[221]L. Sperling, S. Lämmer, H. Hagen, G. Scheuermann, and C. Gillmann. "Uncertainty-aware Evaluation of Machine Learning Performance in binary Classification Tasks". In: *Journal of WSCG* 30.1 (2022), pp. 63–71.

[222]S. L. Stoev and W. Straßer. "Extracting regions of interest applying a local watershed transformation". In: *Proceedings Visualization 2000*. IEEE. 2000, pp. 21–28.

[223]V. Subhash, K. Pandey, and V. Natarajan. "GPU parallel computation of Morse-Smale complexes". In: *2020 IEEE Visualization Conference (VIS)*. IEEE. 2020, pp. 36–40.

[224]D. A. Szafir. "The good, the bad, and the biased: five ways visualizations can mislead (and how to fix them)". In: *Interactions* 25.4 (June 27, 2018), pp. 26–33.

[225]C. Tannert, H.-D. Elvers, and B. Jandrig. "The ethics of uncertainty". In: *EMBO reports* 8.10 (Oct. 2007), pp. 892–896.

[226]J. B. Tenenbaum, V. d. Silva, and J. C. Langford. "A Global Geometric Framework for Nonlinear Dimensionality Reduction". In: *Science* 290.5500 (Dec. 22, 2000), pp. 2319–2323.

[227]The Qt Company. *QT*. Cross-platform software development for embedded & desktop. 2020. URL: `qt.io` (visited on 05/03/2021).

[228]R. Therón and J. F. De Paz. "Visual Sensitivity Analysis for Artificial Neural Networks". In: *Intelligent Data Engineering and Automated Learning – IDEAL 2006*. Intelligent Data Engineering and Automated Learning – IDEAL 2006. Ed. by E. Corchado, H. Yin, V. Botti, and C. Fyfe. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, pp. 191–198.

[229]Q. Tian, Q. Xue, N. Sebe, and T. S. Huang. "Error metric analysis and its applications". In: *Internet Multimedia Management Systems V*. Internet Multimedia Management Systems V. Vol. 5601. International Society for Optics and Photonics, Oct. 25, 2004, pp. 46–57.

[230]J. Tierny, G. Favelier, J. A. Levine, C. Gueunet, and M. Michaux. "The Topology ToolKit". In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2017), pp. 832–842.

[231]A. Torralba, R. Fergus, and Y. Weiss. "Small codes and large image databases for recognition". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. 2008 IEEE Conference on Computer Vision and Pattern Recognition. June 2008, pp. 1–8.

[232]TTK Contributers. *TTK Data Repository, 2020*. `https://github.com/topology-tool-kit/ttk-data`.

[233]TTK Contributers. *TTK Tutorial Data, 2021*. https://topology-tool-kit.github.io/stuff/ttk_tutorial_data.zip.

[234]U.S. Department of Energy National Energy Technology Laboratory. *National Energy Technology Laboratory*. 2019. URL: https://mfix.netl.doe.gov/.

[235]M. Varga and C. Varga. "Visual Analytics: Data, Analytical and Reasoning Provenance". In: *Building Trust in Information*. Ed. by V. L. Lemieux. Springer Proceedings in Business and Economics. Cham: Springer International Publishing, 2016, pp. 141–150.

[236]J. .-. Vay, A. Almgren, J. Bell, L. Ge, et al. "Warp-X: A new exascale computing platform for beam–plasma simulations". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*. 3rd European Advanced Accelerator Concepts workshop (EAAC2017) 909 (Nov. 12, 2018), pp. 476–479.

[237]A. Venkat, A. Gyulassy, G. Kosiba, A. Maiti, H. Reinstein, R. Gee, P.-T. Bremer, and V. Pascucci. "Towards replacing physical testing of granular materials with a Topology-based Model". In: *IEEE Transactions on Visualization and Computer Graphics* 1 (2021), pp. 76–85.

[238]C. T. Vi, K. Takashima, H. Yokoyama, G. Liu, Y. Itoh, S. Subramanian, and Y. Kitamura. "D-FLIP: Dynamic and Flexible Interactive PhotoShow". In: *Advances in Computer Entertainment*. International Conference on Advances in Computer Entertainment Technology. Ed. by D. Reidsma, H. Katayose, and A. Nijholt. Lecture Notes in Computer Science. Springer International Publishing, 2013, pp. 415–427.

[239]L. Vincent and P. Soille. "Watersheds in digital spaces: an efficient algorithm based on immersion simulations". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 13.06 (1991), pp. 583–598.

[240]S. C. Vogel, C. M. Biwer, D. H. Rogers, J. P. Ahrens, R. E. Hackenberg, D. Onken, and J. Zhang. "Interactive visualization of multi-data-set Rietveld analyses using Cinema:Debye-Scherrer". In: *Journal of Applied Crystallography* 51.3 (June 1, 2018), pp. 943–951.

[241]I. Wald, G. Johnson, J. Amstutz, C. Brownlee, A. Knoll, J. Jeffers, J. Günther, and P. Navratil. "OSPRay - A CPU Ray Tracing Framework for Scientific Visualization". In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (Jan. 2017), pp. 931–940.

[242]E. Wall, L. M. Blaha, C. L. Paul, K. Cook, and A. Endert. "Four Perspectives on Human Bias in Visual Analytics". In: *Cognitive Biases in Visualizations*. Ed. by G. Ellis. Cham: Springer International Publishing, 2018, pp. 29–42.

[243]B. Wang, W. Shi, and Z. Miao. "Confidence Analysis of Standard Deviational Ellipse and Its Extension into Higher Dimensional Euclidean Space". In: *PLOS ONE* 10.3 (Mar. 13, 2015).

[244]H. Wang and D.-Y. Yeung. "A Survey on Bayesian Deep Learning". In: *ACM Computing Surveys* 53.5 (Sept. 28, 2020), 108:1–108:37.

[245]J. Wang, S. Hazarika, C. Li, and H.-W. Shen. “Visualization and Visual Analysis of Ensemble Data: A Survey”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.9 (Sept. 2019), pp. 2853–2872.

[246]X. Wang, T. Butkiewicz, W. Dou, E. Bier, and W. Ribarsky. “Designing Knowledge-assisted Visual Analytics Systems for Organizational Environments”. In: VINCI 2011 conference on visual , information communication and interaction. 2011.

[247]T. Weinkauf, Y. Gingold, and O. Sorkine. “Topology-based smoothing of 2D scalar fields with C1-continuity”. In: *Computer Graphics Forum*. Vol. 29. Wiley Online Library. 2010, pp. 1221–1230.

[248]D. Weinstein. “Scanline surfacing: building separating surfaces from planar contours”. In: *Proceedings Visualization 2000*. 2000, pp. 283–289.

[249]F. Welle, K. Stoll, C. Gillmann, J. Henkelmann, et al. “Tissue Outcome Prediction in Patients with Proximal Vessel Occlusion and Mechanical Thrombectomy Using Logistic Models”. In: *Translational Stroke Research* (May 30, 2023).

[250]K. Xu, A. Ottley, C. Walchshofer, M. Streit, R. Chang, and J. Wenskovitch. “Survey on the Analysis of User Interactions and Visualization Provenance”. In: *Computer Graphics Forum* 39.3 (2020), pp. 757–783.

[251]L. Yan, Y. Wang, E. Munch, E. Gasparovic, and B. Wang. “A Structural Average of Labeled Merge Trees for Uncertainty Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (Jan. 2020), pp. 832–842.

[252]V. Yeghiazaryan and I. Voiculescu. “Path reducing watershed for the GPU”. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 577–585.

[253]Z. Yuan, T. L. Bailey, and R. D. Teasdale. “Prediction of protein B-factor profiles”. In: *Proteins: Structure, Function, and Bioinformatics* 58.4 (2005), pp. 905–912.

[254]J. Zhang. *Visualization for Information Retrieval*. Springer Science & Business Media, Nov. 24, 2007. 300 pp.

[255]N. Zhang, X. Zhou, D. Sha, X. Yuan, K. K. Tamma, and B. Chen. “Integrating Mesh and Meshfree Methods for Physics-Based Fracture and Debris Cloud Simulation.” In: *PBG@ SIGGRAPH*. 2006, pp. 145–154.

[256]A. J. Zomorodian. “Topology for Computing”. In: *Algorithms and Theory of Computation Handbook (Second Edition)*. Ed. by M. J. Atallah and M. Blanton. CRC Press, 2010. Chap. 3, pp. 82–112.

Curriculum Vitae

Robin Georg Claus Maack
RPTU Kaiserslautern-Landau
Scientific Visualization Lab
Department Computer Science
maack@rptu.de

Work Experience

2020 – 2024 **Scientific Staff Member**
TU Kaiserslautern and RPTU Kaiserslautern-Landau
Department Computer Science
AG Computer Graphics & HCI
and Scientific Visualization Lab

2017 – 2020 **Student Assistant** TU Kaiserslautern
Department Computer Science
AG Computer Graphics & HCI

Education

2018 – 2020 **Master of Science in Computer Science** TU Kaiserslautern
Thesis: *Towards an Open Source Framework for Uncertainty-Aware Protein Visualization*

2013 – 2018 **Bachelor of Science in Computer Science** TU Kaiserslautern
Thesis: *Validierung der automatischen Spalt- und Bündigkeitsmessung im virtuellen Meisterbock unter Verwendung eines physischen Normals*

2006 – 2013 **Abitur** Pamina Gymnasium Herxheim
Essay: *Scriptsprachen in Computerspielen*

Overview of Utilized Tools

In the creation of this thesis, I have used Grammarly (Chrome Browser Extention, Version 14.1147.0[1]), including the generative AI assistant[2]. The tool has been used for spell-checking, grammar-checking, text formulation, and text editing.

Zotero (Version 6.0.30)[3] has been used for literature management and citation management.

Overleaf[4] has been used as a text editor, including spell-checking.

[1] www.grammarly.com/browser/chrome
[2] www.grammarly.com/ai
[3] www.zotero.org
[4] www.overleaf.com