TU WIEN Informatics

Visual Analysis of Defects

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

Aleksandr Amirkhanov

Matrikelnummer 01428785

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Zweitbetreuung: Dipl.-Ing. Dr.techn. Gabriel Mistelbauer, Dipl.-Ing. Dr.techn. Christoph Heinzl

Diese Dissertation haben begutachtet:

_____   _____
            Ingrid Hotz   Bernhard Preim

Wien, 13. Juli 2021   _____
                                              Aleksandr Amirkhanov

TU WIEN Informatics

Visual Analysis of Defects

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Aleksandr Amirkhanov

Registration Number 01428785

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Second advisor: Dipl.-Ing. Dr.techn. Gabriel Mistelbauer, Dipl.-Ing. Dr.techn. Christoph Heinzl

The dissertation has been reviewed by:

Ingrid Hotz	Bernhard Preim

Vienna, 13th July, 2021

Aleksandr Amirkhanov

Erklärung zur Verfassung der Arbeit

Aleksandr Amirkhanov

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 13. Juli 2021

Aleksandr Amirkhanov

Kurzfassung

Im Alltag nutzen wir zahlreiche Gegenstände, auf die wir uns verlassen und von denen wir erwarten, dass sie jederzeit korrekt funktionieren. Wir verwenden Telefone, um mit Freunden zu kommunizieren, Fahrräder, um zu pendeln, Bankomatkarten, um Lebensmittel einzukaufen. Defekte können jedoch dazu führen, dass diese Gegenstände nicht mehr funktionieren, mit teils schwerwiegenden Folgen für uns. Daher arbeiten Unternehmen kontinuierlich an der Verbesserung ihrer Produkte und Dienstleistungen. Hierbei stellen sie sicher, dass alles unter der Einhaltung von Qualitätsrichtlinien geschieht. Eine gängige Strategie des Qualitätsmanagements ist die Defektanalyse, die dazu dient, Erzeugnisse zu kontrollieren und Defekte zu identifizieren, die nicht den Richtlinien entsprechen. Traditionelle Methoden der Fehleranalyse sind oft manuelle und damit zeitaufwendige Verfahren. Um effizientere Lösungen zu schaffen, werden bei der Defektanalyse zunehmend visuelle Analysetechniken eingesetzt. Diese Techniken automatisieren und erweitern die bisherigen manuellen Analyseschritte und unterstützen neue visuelle Ansätze zur Defektdarstellung, die bestehende Defekte aufdecken, ohne neue zu verursachen. In dieser Dissertation werden visuelle Analysetechniken, die im Bereich der Defektanalyse angewendet werden, als visuelle Defektanalyse bezeichnet. Da es sich hierbei um ein neues und sich schnell entwickelndes Gebiet handelt, fehlt noch eine formalisierte Grundlage.

In dieser Dissertation wird ein Ablaufplan für die visuelle Defektanalyse vorgestellt und diskutiert, der auf dem Plan-Do-Check-Act-Zyklus der kontinuierlichen Verbesserung basiert. Der Arbeitsablauf besteht aus vier Schritten: Fehlervermeidung, Kontrolle der fehlerhaften Erzeugnisse, Bewertung und Verbesserung. Während des Schrittes der Fehlervermeidung planen Fachexperten Design- und Entwicklungsprozesse, um sicherzustellen, dass die beabsichtigten Resultate erreicht werden können, und prognostizieren gleichzeitig Risiken und Chancen. Während der Kontrolle der fehlerhaften Erzeugnisse implementieren Fachexperten die Prozesse und kontrollieren die dabei auftretenden Fehler. Während des Bewertungsschritts stellen die Fachexperten sicher, dass fehlerhafte Erzeugnisse durch Messung von Objekteigenschaften identifiziert werden. Während des Verbesserungsschritts untersuchen Fachexperten mögliche Aktionen, die die Erzeugnis-Qualität verbessern.

In dieser Dissertation werden vier Ansätze vorgestellt, die die visuelle Defektanalyse in den vier verschiedenen Schritten des Ablaufplans vorantreiben. Der erste Ansatz entspricht dem Schritt der Fehlervermeidung und bietet eine Vorschau für Zahnbehandlungen. Sie hilft ZahntechnikerInnen, die am besten geeignete Behandlungsoption zu identifizieren

und Fälle zu vermeiden, in denen PatientInnen aufgrund einer schlechten Ästhetik des Zahnersatzes mit dem Resultat unzufrieden sind. Der zweite Ansatz entspricht dem Schritt der Kontrolle von fehlerhaften Erzeugnissen und unterstützt Zahntechniker bei der Gestaltung von ästhetischem und funktionellem Zahnersatz. Der Ansatz bietet eine sofortige visuelle Rückmeldung über eine Änderung des Prothesendesigns. So kann man bewerten, wie sich Änderungen auf die Ästhetik auswirkt. Der dritte Ansatz entspricht dem Bewertungsschritt und unterstützt Materialingenieure bei der Untersuchung von Schäden in Verbundwerkstoffen. Zunächst erfasst und misst das System verschiedene Defekte wie Matrixbruch, Faser/Matrix-Ablösung, Faserauszug und Faserbruch. Später analysieren Experten diese Defekte mit verschiedenen interaktiven Visualisierungstechniken. Der vierte Ansatz entspricht dem Verbesserungsschritt und visualisiert dynamische 4D-Systeme, die verschiedene Phänomene beschreiben. Unsere Lösung ermöglicht die 4D-Darstellung dynamischer Systeme und erlaubt den nahtlosen Übergang der 4D-Darstellung in, dem Anwender vertraute, niederdimensionale Graphiken.

Abstract

In everyday life, we use many *objects* on which we rely and expect them to work correctly. We use phones to communicate with friends, bicycles to commute, payment cards to buy groceries. However, due to *defects*, these objects may fail at some time, leading to adverse outcomes. Modern industry continually improves the quality of outputs (e.g., products and services) and ensures that they meet their specifications. A common quality management strategy is the defect analysis used to identify and control outputs that do not conform to their specifications. Traditional defect analysis methods are often manual and, therefore, time-consuming procedures. To build more efficient solutions, defect analysis increasingly employs visual analytics techniques. These techniques automatize and enhance the up-to-now manual analysis steps and support new visual approaches for defect representations that resolve existing defects without introducing new ones. In this dissertation, visual analytics techniques applied to defect analysis are referred to as *visual analysis of defects*. Being a rapidly developing area, the domain of visual analysis of defects is still missing a formalized basis.

This dissertation presents and discusses a workflow for the visual analysis of defects based on the plan-do-check-act cycle of continual improvement. The workflow consists of four steps: defect prevention, control of defective outputs, performance evaluation, and improvement. During the *defect prevention* step, domain experts plan the design and development processes to ensure that intended results can be achieved while forecasting risks and opportunities. During the *control of defective outputs* step, domain experts implement the processes and control defects arising throughout these processes. During the *performance evaluation* step, domain experts ensure that defective outputs are identified by measuring the object's characteristics. During the *improvement* step, domain experts explore possible actions that improve the object quality.

This dissertation presents four solutions that advance the visual analysis of defects at the four distinct steps of the workflow. The first solution corresponds to the *defect prevention* step and provides a preview of dental treatment. It helps dental technicians to identify the most suitable treatment option and avoid cases when patients are unsatisfied with the results due to poor denture aesthetics. The second solution corresponds to the *control of defective outputs* step and supports dental technicians in designing aesthetic and functional dentures. The approach provides immediate visual feedback on a change in the denture design, which helps to evaluate how the change affects aesthetics. The

third solution corresponds to the *performance evaluation* step and supports material engineers in investigating the damage mechanism in composite materials. First, the system captures and measures various defects such as matrix fracture, fiber/matrix debonding, fiber pull-out, and fiber fracture. Later, users analyze these defects using several interactive visualization techniques. The fourth solution corresponds to the *improvement* step and visualizes 4D dynamical systems describing various phenomena. The solution enables the 4D representation of dynamical systems and allows the 4D representation to seamlessly transition into, familiar to the user, lower-dimensional plots.

Contents

Introduction

We are surrounded by an enormous number of *objects*, which have become an indispensable part of our everyday lives, as we use them to perform various tasks and activities. For example, an alarm clock wakes us up in the morning, the key locks the door, the bus transports us within the city. We usually expect these objects to function flawlessly. However, this is not always the case, and everything may fail at some time.

Failures have unintended and often harmful consequences, the effect of which depends on many factors such as an object's intended use, materials employed, or economic value. For example, a car failure and a toy radio-controlled car failure have entirely different consequences. The former's potential effect is much higher as it puts human life in danger and has a higher economic impact. We further provide several examples of notable failures in recent history to illustrate how they can affect our lives and cause significant disruption.

In 1983, a Soviet missile attack warning system reported a lunch of several intercontinental ballistic missiles from United States bases [215]. Lieutenant colonel Stanislav Petrov was the officer on duty at the military base where the warning system was located. He was responsible for warnings validation and reporting them further up in the chain of command. He could not see the missile lunch in another radar system. Also, he assumed that if the United States decided to attack the Soviet Union, they would lunch many missiles, not a few. Therefore, he decided that it was a false alarm and reported it correspondingly. If he decided differently, it might have resulted in an immediate massive nuclear counterattack. The later investigation showed that the missile warning system's error was caused by a sensor overexposure of the satellite carrying the system. Later the warning system was modified to filter out such rare events.

Another example is the fatal accident of the spaceship Challenger, where the whole crew was killed during the spaceship launching [268]. According to a later examination, the catastrophic event might have been produced by a failure in O-rings that were sealing joins of rocket booster sections. After the disaster, the number of O-rings was increased from two to three in all NASA's boosters to increase the system's robustness.

The third and last example is the explosion of the Chernobyl nuclear power station that caused major destruction and large radioactive contamination [220]. There are several versions of what has caused the explosion. However, the exact reason is still not determined. The official version says that the reactor explosion was caused by improper reactor maintenance by the power plant crew. It was discovered that the nuclear reactor had several design flaws. Even though the reactor constructors knew these flaws, they were not properly reflected in the operational documents. Insufficient operational documents might have also contributed to the crew's wrong actions.

Modern industry analyzes failures to develop strategies for improving the quality of outputs such as products and services. During these analyses, users collect and carefully analyze data concerning the object condition before and after the failure if possible. Subsequently, they generate hypotheses of possible failure causes and build models that describe the failure process. A common cause of failures is *defects*, which are introduced during the design and development processes.

1.1 What Is a Defect?

Defect is an umbrella term related to different objects such as products, processes, or systems. All defects are undesirable and may lead to failures. There are many formal definitions of a defect that refer to different concepts. They vary by scientific discipline and the level of technical details. For example, in chemistry and medicine different phenomena are characterized or assumed to be defects.

Within the scope of this dissertation, we use the definition of a defect according to ISO 9000 [154]. This choice was made due to its clarity and ability to relate to the object quality concept:

> **defect**
> *nonconformity* related to an intended or specified use.
> ↕
> **nonconformity**
> non-fulfilment of a *requirement*.
> ↕
> **requirement**
> need or expectation that is stated, generally implied, or obligatory.
> ↕
> **quality**
> degree to which a set of inherent characteristics of an object fulfils requirements.
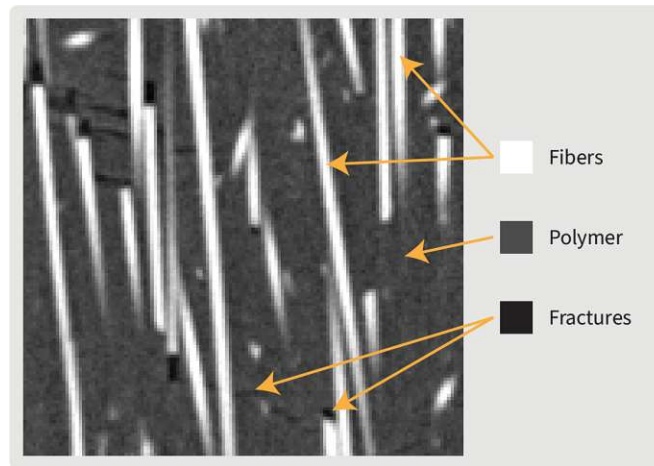
Figure 1.1: Computed tomography imaging of a composite material enables capturing internal defects. The material consists of glass fibers (white) bound with a polymer (gray). Fractures (black) are defects, and they reduce material strength characteristics.

The terms *defect* and *nonconformity* are closely related and often can be used synonymously. Nevertheless, they refer to concepts that have different legal implications [154]. For simplicity, in this dissertation, we do not differentiate nonconformities and defects and refer to nonconformities as defects.

According to the previous definitions, a defective object cannot fulfill all the requirements and therefore has a lower quality or cannot be used for its intended purpose. Figure 1.1 shows an example of a defective object, i.e., a composite material with fractures. These fractures reduce the material durability and increase the risk of breakages. Even though fractures might not necessarily lead to a failure, their monitoring and analysis are essential to ensure object conformity. To this end, continual improvement of the design and development processes is essential.

Within the scope of the dissertation, we consider four defect types:

- **Spatial defects:** they are described as geometrical objects in Euclidean space or on a Euclidean plane (e.g., material fracture).

- **Temporal defects:** they are described as object characteristics measured in time (e.g., the brake system temperature measured multiple times during stopping).

- **Spatiotemporal defects:** they are described as geometrical objects in Euclidean space or on a Euclidean plane measured in time (e.g., corrosion development in time).

- **Design space defects:** they are described as (combinations of) object characteristics in the design space that prevent the object from functioning correctly (e.g., bridge collapse during bad weather).

3
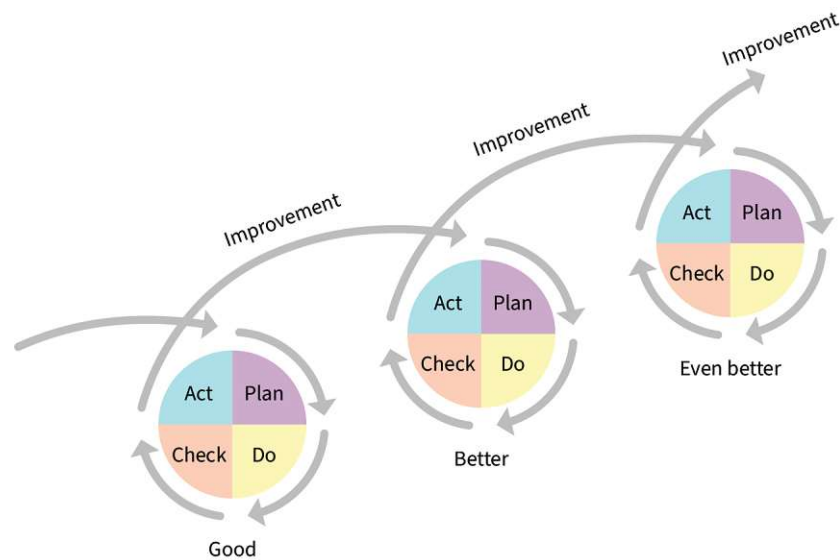
Figure 1.2: The plan-do-check-act cycle of continual improvement.

1.2 The Plan-Do-Check-Act Cycle

The plan-do-check-act (PDCA) cycle is an iterative four-step cycle of continual improvement [22, 87]. We illustrate the concept behind this process in Figure 1.2. The PDCA cycle defines the sequence of actions to ensure the proper management of processes and resources. Every new iteration of the circle brings additional knowledge or improvements to the design and development processes. The knowledge can be successful as well as failed experiments. For example, knowing that changing an object in a specific way will not eliminate the defect, is essential as it provides an additional clue for the next iteration. The PDCA cycle is described as follows:

- **Plan:** the user formulates expectations from the result, sets the system's objectives and processes, specifies needed resources to achieve the intended result, and identifies risks and opportunities. In this step, the output is not yet produced and delivered. The user needs to plan developing processes to avoid possible defects.

- **Do:** the user implements what was planned. In this step, the output is being produced and not yet delivered. The user needs to control design and manufacturing processes to avoid possible defects.

- **Check:** the user monitors and measures processes and results. In this step, the output is produced and not yet delivered. The user needs to verify the integrity and conformity of the output.
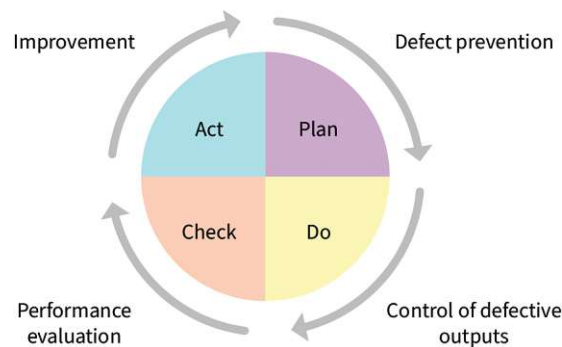
Figure 1.3: Four steps of defect analysis.

- **Act:** the user studies the results, identifies possible actions to improve the system's performance. In this step, the output is produced and delivered. If any issues occurred in the *do* or *check* step, the user has obtained additional experience and knowledge that can be used in the next iteration.

Users can apply the PDCA cycle to various processes, activities, and systems to achieve an improvement. The focus of the improvement depends on the users' needs. For example, the focus might be to make a manufacturing process more cost-efficient, faster, more environmentally friendly, more supplier independent, or to increase the quality of outcomes. Users can apply the PDCA cycle to manage and control defects to increase the quality of outcomes.

1.3 Defect Analysis

We refer to defect analysis as part of the quality improvement procedure that is focused on defects. Defect analysis follows the PDCA cycle, and according to the ISO 9001 [155] consists of the following steps, which are also illustrated in Figure 1.3:

- **Plan → defect prevention:** Domain experts implement a strategy to ensure that intended results can be achieved. This is done by evaluating the effect of any action or change. The organization aims at enhancing desirable effects and preventing or reducing undesired ones.

- **Do → control of defective outputs:** Domain experts implement a design and development process. During this process, occurring defects are corrected to ensure that outputs conform to the requirements. After a defect correction, the output conformity should be verified.

- **Check → performance evaluation:** Domain experts ensure that defective outputs are identified by measuring the object's characteristics such as size, shape, volume, or number of micro-cracks. Some characteristics can be directly related to

undesirable or defective outputs. Depending on the requirements, however, a certain level of these characteristics can be acceptable. Monitoring these characteristics is essential to ensure conforming outputs.

- **Act → improvement:** Domain experts explore possible actions that would improve the object quality. This task requires an overview of the system describing the object behavior and how a change will affect the results. Having this overview, the organization decides how to modify the system state to enhance the quality without introducing new defects.

Defect analysis increasingly employs computer graphics techniques, which are faster and more efficient than previous manual approaches. These techniques provide visual representations of objects and defects to support domain experts in decision making and hypothesis confirmation [205]. Additionally, they provide a platform for discussions among the users and for knowledge dissemination [221]. We refer to a visually assisted defect analysis process as *visual analysis of defects.* The visual analysis of defects widely employs visualization and visual analytics techniques, which we discuss in Chapter 2.

1.4   Research Aim

The research aim of this dissertation is formulated as follows:

> Investigate strategies supporting the visual analysis of defects within the four steps of defect prevention, control of defective outputs, performance evaluation, and improvement.

Despite numerous potential application areas, there is no dedicated previous work that reflects on the topic of visual analysis of defects and provides comprehensive recommendations on selecting adequate visual analytics methods. Although many applications follow similar procedures, our proposed workflow has not been previously presented and discussed. Additionally, suitable methods for the visual analysis of defects at different defect analysis steps have not been previously reviewed, and relevant questions such as how visual analytics can support defect identification, quantification, and resolution were not sufficiently addressed before our work.

Based on the research aim, we specify the following three *objectives*:

[Obj1] Identify visualization and visual analytics strategies relevant to the visual analysis of defects.

[Obj2] Design and develop novel strategies supporting the visual analysis of defects within the four steps of defect prevention, control of defective outputs, performance evaluation, and improvement.

[Obj3] Describe how the visual analysis of defects addresses different types of defects (spatial, temporal, spatiotemporal, and design space defects).
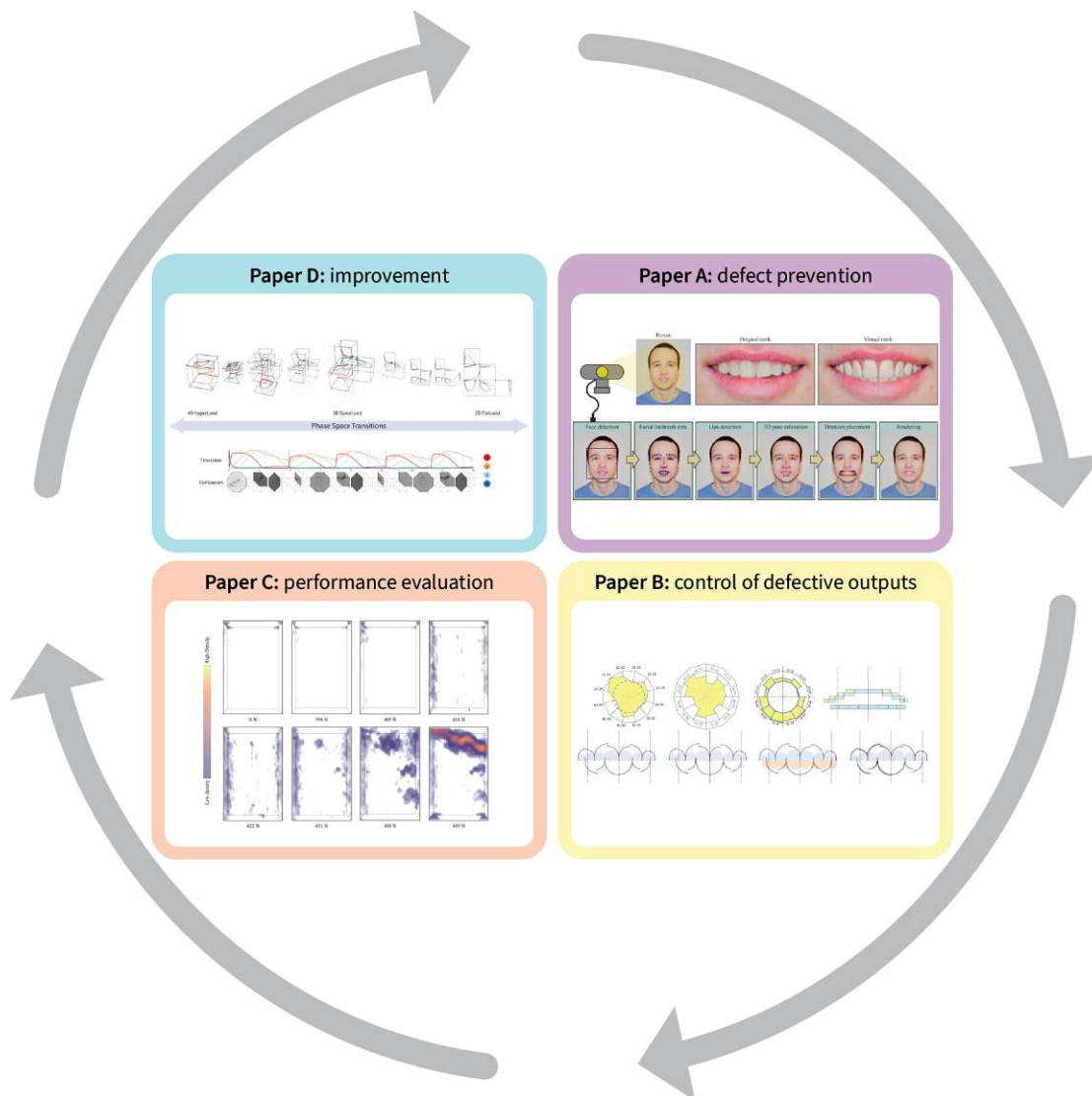
Figure 1.4: The four steps of visual analysis of defects and the solutions proposed within the scope of this dissertation to address them.

1.5 Contributions

The contributions of this dissertation consist of four solutions supporting the visual analysis of defects, which were designed, implemented, and published within the scope of this dissertation. Figure 1.4 shows how our proposed solutions cover all four steps of the visual analysis of defects process. In the next pages, we overview these solutions and provide individual contributions of authors bases on the contributor roles taxonomy (CRediT) [6].

Paper A: *Aleksandr Amirkhanov*, Artem Amirkhanov, Matthias Bernhard, Zsolt Toth, Sabine Stiller, Andreas Geier, M. Eduard Gröller, Gabriel Mistelbauer, "WithTeeth: denture preview in augmented reality," in *Vision, Modeling and Visualization*, The Eurographics Association, 2018, DOI: 10.2312/vmv.20181250.

This paper supports the *defect prevention* (i.e., the *plan* phase of the PDCA cycle) in the context of an augmented reality approach for previewing different dental treatment options. The scientific contributions of this paper include:

- an augmented reality approach for previewing different dental treatment options;

- a technique that achieves compelling visualizations without depending on complex reconstruction algorithms, expensive hardware, or tedious calibration procedures;

- a real-time system that permits user's free head movements and allows switching different teeth replacement options interactively.

Individual contributions of authors based on CRediT:

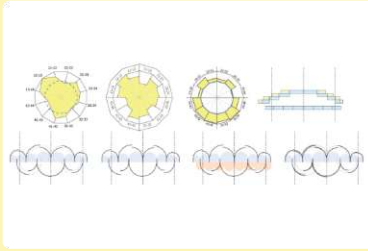|	Aleksandr Amirkhanov	Artem Amirkhanov	Matthias Bernhard	Zsolt Toth	Sabine Stiller	Andreas Geier	M. Eduard Gröller	Gabriel Mistelbauer
Conceptualization	×		×	×	×	×	×	×
Funding acquisition					×	×		×
Investigation	×							
Methodology	×		×					×
Project administration					×			×
Resources					×	×		
Software	×	×						
Supervision			×				×	×
Validation	×							
Visualization	×							
Writing—original draft	×	×	×					
Writing—review & editing	×	×	×				×	×

Paper B: *Aleksandr Amirkhanov*, Matthias Bernhard, Alexey Karimov, Sabine Stiller, Andreas Geier, M. Eduard Gröller, Gabriel Mistelbauer, "Visual analytics in dental aesthetics," *Computer Graphics Forum*, vol. 39, no. 7, pp. 635–646, 2020. DOI: 10.1111/cgf.14174.

This paper supports the *control of defective outputs* (i.e., the *do* phase of the PDCA cycle) in the context of an approach helping dental technicians to adjust a denture that fits a patient's appearance. The scientific contributions of this paper include:

- an approach that assists dental technicians in identifying the most aesthetically fitting denture and subsequently adjusting it;

- a technique that predicts the appearance of a patient with in-line designed dentures;

- a novel approach for quantifying several aspects of dental aesthetics;

- a technique for visualizing dental aesthetic aspects interactively during denture design;

- an approach that is integrated into the state-of-the-art digital workflow of dental technicians.

Individual contributions of authors based on CRediT:

	Aleksandr Amirkhanov	Matthias Bernhard	Alexey Karimov	Sabine Stiller	Andreas Geier	M. Eduard Gröller	Gabriel Mistelbauer
Conceptualization	×	×	×	×	×	×	×
Funding acquisition				×	×		×
Investigation	×						
Methodology	×	×	×				×
Project administration				×			×
Resources				×	×		
Software	×		×				
Supervision		×	×			×	×
Validation	×				×		
Visualization	×						
Writing—original draft	×	×	×				
Writing—review & editing	×	×	×			×	×

Paper C: *Aleksandr Amirkhanov*, Artem Amirkhanov, Dietmar Salaberger, Johann Kastner, M. Eduard Gröller, Christoph Heinzl, "Visual analysis of defects in glass fiber reinforced polymers for 4DCT interrupted in situ tests," *Computer Graphics Forum*, vol. 35, no. 3, pp. 201–210, 2016. DOI: 10.1111/cgf.12896.

This paper supports the *performance evaluation* (i.e., the *check* phase of the PDCA cycle) in the context of an application for material characterization and conformity verification. The scientific contributions of this paper include:

- a novel technique for defect extraction and classification in computed tomography images of glass fiber reinforced polypropylene;

- a set of techniques for visual analysis of material defects.

Individual contributions of authors based on CRediT:

	Aleksandr Amirkhanov	Artem Amirkhanov	Dietmar Salaberger	Johann Kastner	M. Eduard Gröller	Christoph Heinzl
Conceptualization	×		×		×	×
Funding acquisition						×
Investigation	×					
Methodology	×		×			×
Project administration						×
Resources			×	×		
Software	×					
Supervision					×	×
Validation	×		×			
Visualization	×					
Writing—original draft	×	×	×			
Writing—review & editing	×	×			×	×

Paper D: *Aleksandr Amirkhanov*, Ilona Kosiuk, Peter Szmolyan, Artem Amirkhanov, Gabriel Mistelbauer, M. Eduard Gröller, Renata G. Raidou, "Manylands: A journey across 4D phase space of trajectories," *Computer Graphics Forum*, vol. 38, no. 7, pp. 191–202, 2019. DOI: 10.1111/cgf.13828.

This paper supports the *improvement* (i.e., the *act* phase of the PDCA cycle) in the context of an approach to analyze and study complex 4D mathematical models describing biological processes. The scientific contributions of this paper include:

- design and implementation of a technique that supports mathematical domain scientists in understanding and analyzing the behavior of biologically meaningful 4D dynamical system trajectories;

- a novel technique to discover new knowledge within mathematical systems;

- a novel technique to illustrate findings of mathematical specialists for externalization and education;

- an interactive and integrated workflow for exploring and analyzing 4D dynamical systems trajectories.

Individual contributions of authors based on CRediT:

	Aleksandr Amirkhanov	Ilona Kosiuk	Peter Szmolyan	Artem Amirkhanov	Gabriel Mistelbauer	M. Eduard Gröller	Renata G. Raidou
Conceptualization	×	×	×		×	×	×
Funding acquisition							
Investigation	×						
Methodology	×	×	×				×
Project administration							
Resources		×	×				
Software	×			×			
Supervision					×	×	×
Validation	×	×	×				×
Visualization	×						×
Writing—original draft	×	×	×				×
Writing—review & editing	×	×	×		×	×	×

11

Within the scope of the dissertation, the following articles have also been published, but are not included in the upcoming chapters:
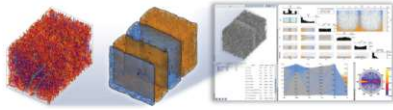
Johannes Weissenböck, Artem Amirkhanov, Weimin Li, Andreas Reh, *Aleksandr Amirkhanov*, M. Eduard Gröller, Johann Kastner, Christoph Heinzl, "Fiber-Scout: an interactive tool for exploring and analyzing fiber reinforced polymers," in 2014 IEEE Pacific Visualization Symposium, Mar. 2014, pp. 153–160. DOI: 10.1109/PacificVis.2014.52.
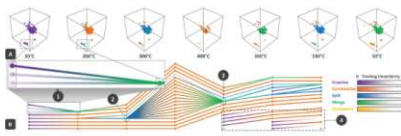
Andreas Reh, *Aleksandr Amirkhanov*, Johann Kastner, M. Eduard Gröller, Christoph Heinzl, "Fuzzy feature tracking: visual analysis of industrial 4D-XCT data," Computers & Graphics, vol. 53, pp. 177–184, 2015, ISSN: 0097-8493. DOI: 10.1016/j.cag.2015.04.001.
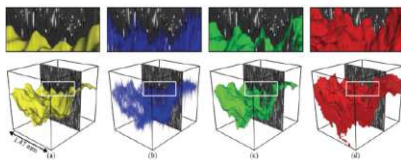
Aleksandr Amirkhanov, Dietmar Salaberger, Johann Kastner, Christoph Heinzl, M. Eduard Gröller, "Comparison of final fracture extraction techniques for interrupted in situ tensile tests of glass fiber reinforced polymers," in 7th Conference on Industrial Computed Tomography, 2017.
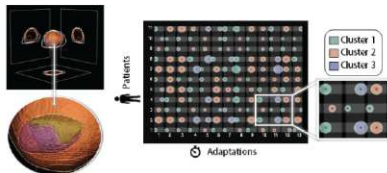
Renata G. Raidou, Oscar Casares-Magaz, *Aleksandr Amirkhanov*, Vitali Moiseenko, Ludvig Muren, John Einck, Anna Vilanova, and M. Eduard Gröller, "Bladder Runner: Visual analytics for the exploration of RT-induced bladder toxicity in a cohort study," Computer Graphics Forum, vol. 37, no. 3, pp. 205–216, 2018. DOI: 10.1111/cgf.13413.

CHAPTER 2

Related Work

Visual analysis of defects employs methods from the fields of visualization and visual analytics. In this chapter, we briefly discuss the main principles and foundations of these fields (Sections 2.1 and 2.2). Then, we discuss visualization and visual analytics methods relevant to the visual analysis of defects (Section 2.3). We group these methods into four distinct categories, which reflect the PDCA cycle steps: defect prevention (Section 2.3.1), control of defective outputs (Section 2.3.2), performance evaluation (Section 2.3.3), and improvement (Section 2.3.4).

2.1 Visualization

"Visualization is a method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. Visualization offers a method for seeing the unseen. It enriches the process of scientific discovery and fosters profound and unexpected insights. In many fields, it is already revolutionizing the way scientists do science."

McCormick et al. [195]

Visualization was established as a separate scientific discipline relatively recently. The rapid development of computer technology in the 20th and 21st centuries provided almost limitless possibilities for data simulations and image generations employing computer graphics. However, visualization as manually drawn figures, plots, or physical mockups existed a long time ago before the computer age. They were mainly created to study various natural, physical, mathematical, and other phenomena. These mockups reflect non-trivial processes, which are barely noticeable using the human visual or other perceptual system due to the complexity of the phenomena or its transience. Figure 2.1 provides several examples of visualization and physicalizations from different times.

13
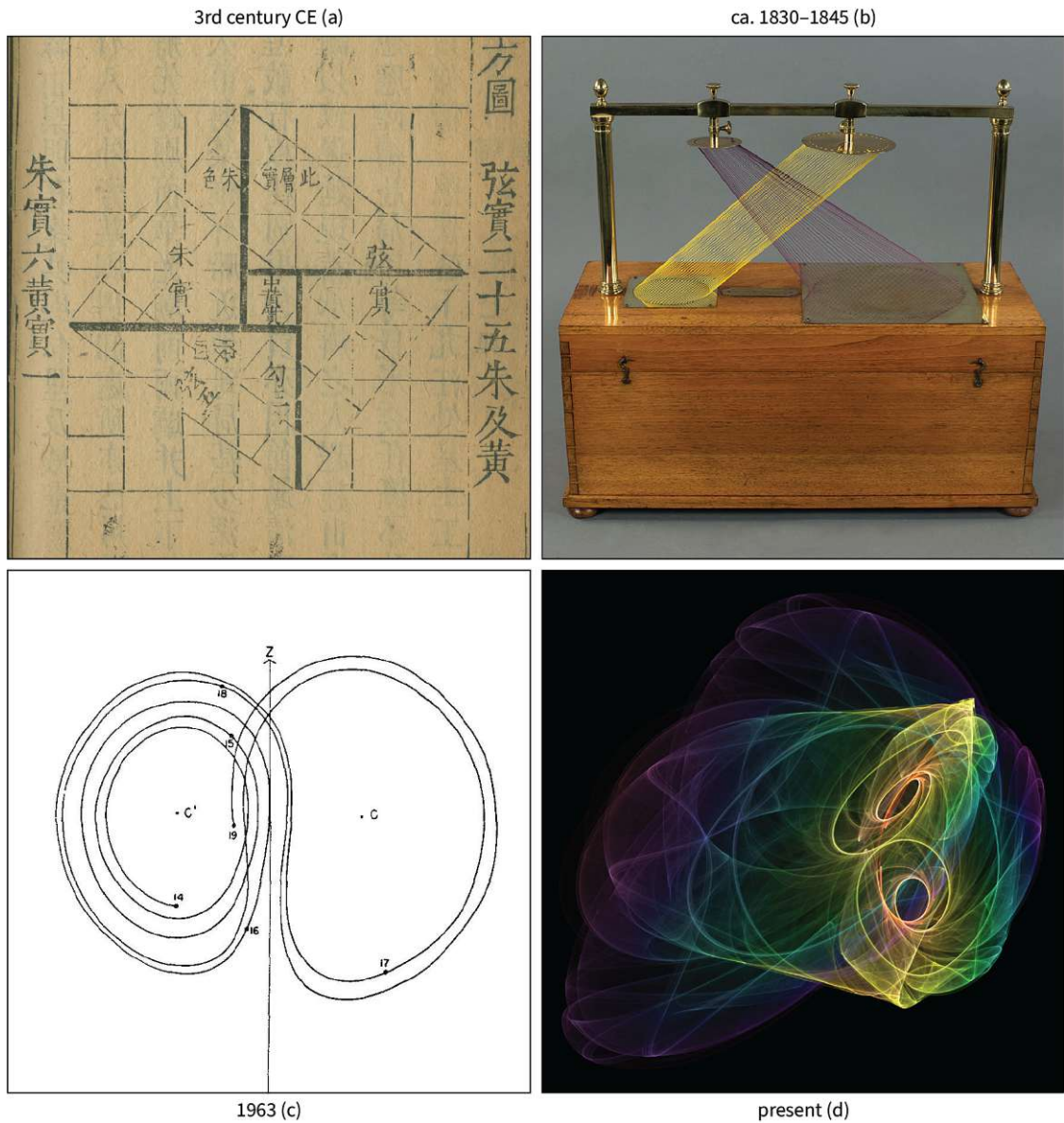
Figure 2.1: Evolution of visualization. *(a) 3rd century CE:* a diagram in the Zhoubi Suanjing that can be used to prove the Pythagorean theorem [81, 255]. *(b) ca. 1830–1845:* Théodore Olivier constructed a string model for pedagogical purposes showing the intersection of two geometrical shapes in 3D [13]. *(c) 1963:* one of the earliest visualizations of the Lorenz attractor. It was computed for a relatively small number of iterations due to the then limited computing power (Royal McBee LGP-30) [188]. *(d) Present:* a modern visualization of the Lorenz attractor obtained using the Chaoscope software [7]. Copyright notices: (a) © 2011 Mathematical Association of America, (b) Courtesy of the Union College Permanent Collection, (c) © 1963 American Meteorological Society.
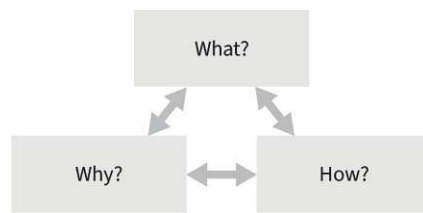
Figure 2.2: The three-part visualization analysis framework proposed by Munzner [205].

Scientific studies [151, 191] confirm that sight is the dominant human sense, and vision delivers more information to the brain than any other sensory system. The human visual system is a high-bandwidth channel connecting the human brain to the outer world, making it essential for cognition. Visualization employs the human visual system's capacity to deliver valuable information about an object to an observer. The efficiency of a visualization depends on the achieved results and used resources by the user. Therefore it is crucial to design intuitive, easy-to-read, informative, and computationally efficient visualization techniques.

For developing a visualization system, it is essential to analyze the available data, the target users, and the user tasks. Munzner [205] proposed a three-part analysis framework for a visualization technique (see Figure 2.2), consisting of three components:

- ***What*** *data does the user want to analyze?* Data is important for visualization. Different visual metaphors are best for representing distinct data types. For instance, time-oriented data visualization often benefits from a timeline or an animation to show changes. For static data, these visual representations are less common.

- ***Why*** *does the user intend to use a visualization tool?* Each visualization has an intended purpose. For example, visualization techniques can enhance data analysis, confirm or disprove hypotheses, deliver already known information to laymen or students. Naturally, these tasks require different visual metaphors to be used.

- ***How*** *are visual representations and interaction idioms are employed to show data?* The user can accomplish a required task by employing different approaches that have specific advantages and disadvantages. When creating a visualization, it is essential to reflect on made design choices, i.e., how a visualization shows the data, the pros and cons of the selected visualization approach.

Visualization supports the visual analysis of defects in multiple ways. Visualization improves the interface between the data and the user and provides efficient visual representations to describe (more or less) complex processes. Visualization also enhances knowledge discovery and the dissemination of interesting findings. Finally, visual data representations aim at increasing the users' trust and control over the visual analysis of defects.

2.2 Visual Analytics

"Visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces."

Thomas and Cook [74]

Visual analytics serves as an interface between humans and computers, providing (interactive) data representations to the user on demand. It supports analytical reasoning and facilitates the visual analysis of data. Visual analytics methods are often designed to address specific applications and user tasks. Therefore, visual analytics requires careful design planning to address user's needs and requirements. Visual analytics can be used to understand data, confirm or disprove hypotheses, and discover new knowledge.

Visual analytics differs from visualization by the aim. The visualization aim is broader than the one of visual analytics. While the visual analytics aim is to support analytical tasks or to advance knowledge through visual and interactive methods, the aim of visualization is to represent different data types graphically. Visual analytics employs visualization as one of its core components and can include additional areas such as machine learning, statistics, data mining, and advanced interactions. The table below shows differences between visualization and visual analytics:

	Visualization	Visual analytics
Aim	Focuses on graphical data representations to enable users to see the unseen.	Allows the users to visually solve tasks by providing advanced interactive and analytical techniques.
Complex and dynamic data	Optional	Essential
Data representations	Essential	Essential
Visual representations	Essential	Essential
Interactive visualizations	Optional	Essential
Interplay between human and machine	Rare	Essential
Automatic analysis techniques	Optional	Essential
Analytical reasoning and knowledge discovery	Optional	Essential
Knowledge externalization and dissemination	Essential	Essential

Visual analytics has the following distinctive characteristics [36, 74]:

- *Complex and dynamic data*—visual analytics provides approaches to manage complex and dynamic data. These approaches often employ automatic or semi-automatic data filtering and processing methods, which clean up visual scenes and provide

more specific visual representations, emphasizing particular data characteristics. Depending on the current analysis task, the user can control which data features to visualize. These adjustments infuse the visual data representation to help to reveal complex patterns and behavior.

- *Data representations*—visual analytics methods consider in which format to store and handle data. The data format should be efficient, scalable, and represent as much information as possible.

- *Visual representations*—visual analytics focuses on how important information in the data can be delivered through the human vision.

- *Interactive visualizations*—user interactions with a visual analytics system trigger an automatic update of data representations. This update should be performed with as small latency as possible to improve the user experience and accelerate the analysis procedure. Sometimes, a visual analytics system balances between the desired level of interactivity and the required precision of simulations.

- *Interplay between human and machine*—visual analytics provides an interface between the human and the machine, allowing the former to efficiently explore and analyze data. The user can interact with a system through conventional input devices (e.g., mouse and keyboard) or other interfaces (e.g., haptic devices, gestures). The system usability is crucial as it defines how efficiently the user can communicate with the system and receive the required information.

- *Automatic analysis techniques*—algorithms supporting the visual analysis can interactively recompute various data features and characteristics. Fast calculations in the background are preferred as they increase the usability and efficiency of visual analytics systems. Typically, analysis techniques can address only specific user tasks and work with fewer data types. Therefore, finding suitable algorithms for the task is one of the challenges of visual analytics.

- *Analytical reasoning and knowledge discovery*—visual analytics provides visual explanations of various phenomena by showing data features and patterns. As some patterns are easier to understand visually, visual analytics supports analytical reasoning. Knowledge discovery is also possible with visual analytics as visual representations of the data provide additional insights.

- *Knowledge externalization and dissemination*—visual analytics concern how to share the discoveries within and outside the community. For example, how to communicate discoveries to the collaborators and use them for educational purposes.

Visual analytics provides user-centric techniques to manage defects. The interaction between the user and the visual analytics system is of particular importance. Background calculations coupled with interactive techniques shorten and enhance the analysis process,

making it possible to identify defects as early as possible and develop a strategy to correct them. The visual analysis of defects also benefits from knowledge externalization and the dissemination capabilities of visual analytics systems.

2.3 Visual Analysis of Defects

We discuss the workflow for the visual analysis of defects with a simple example of a person, John, building a chair (Figure 2.3). The first step in the workflow is to *plan*. In this step, John realizes the need to build a chair and considers different options for its manufacturing. John imagines possible chair designs, and evaluates their advantages and disadvantages through a *planning preview*. He knows that he is not very skillful and cannot make a very complex design because he can fail. At the same time, he wants a comfortable chair. He compares different options in an *outcome comparison preview* and even considers hybrid solutions that would combine elements of distinct designs, analyzing the entire *parameter space*. After thinking a while, he finally chooses to pursue the easiest option. He is confident he will succeed in building it, and he does not foresee many risks. This described scenario corresponds to the *defect prevention* step of the workflow for the visual analysis of defects. To this step, the following areas of visual analytics are relevant: planning preview, outcomes comparison preview, and parameter space analysis. We discuss these areas in Section 2.3.1.

The second step is to *do*. In this step, John knows how he wants to build the chair and starts manufacturing it. Sometimes John makes mistakes, and he corrects them *interactively*. From time to time, John *progressively* tests the chair on loose parts or other minor defects, although the design is not complete yet. Sometimes, when these mistakes happen, he does not have much time, and he wants to fix them *in situ*, i.e., on-the-fly. However, he always considers several options, among which he chooses the most suitable one. This scenario corresponds to the *control of defective outputs* step of the workflow for the visual analysis of defects. To this step the following areas of visual analytics are relevant: interactive visualizations, progressive visual analytics, and *in situ* visualizations. We discuss these areas in Section 2.3.2.

The third step is to *check*. In this step, John has built the chair and wants to check its performance. First, he sits on the chair to make sure that it is stable and does not wobble. Then he brings it next to the table and checks that his laps do not touch the table. Lastly, he reviews the chair's surface to make sure it is finished well, without any splinters. He sees that there are a few minor flaws, but overall he is happy with the result. This scenario corresponds to the *performance evaluation* step of the workflow for the visual analysis of defects. To this step, the area of visual analytics that measures and visualizes features of different types of defects is relevant. We discuss this area in Section 2.3.3.

The fourth step is to *act*. In this step, John decides to build a few more chairs of the exact same design. However, the first chair took him quite some time, and he wants to build the new ones faster. He reflects on the whole manufacturing procedure, analyzes all
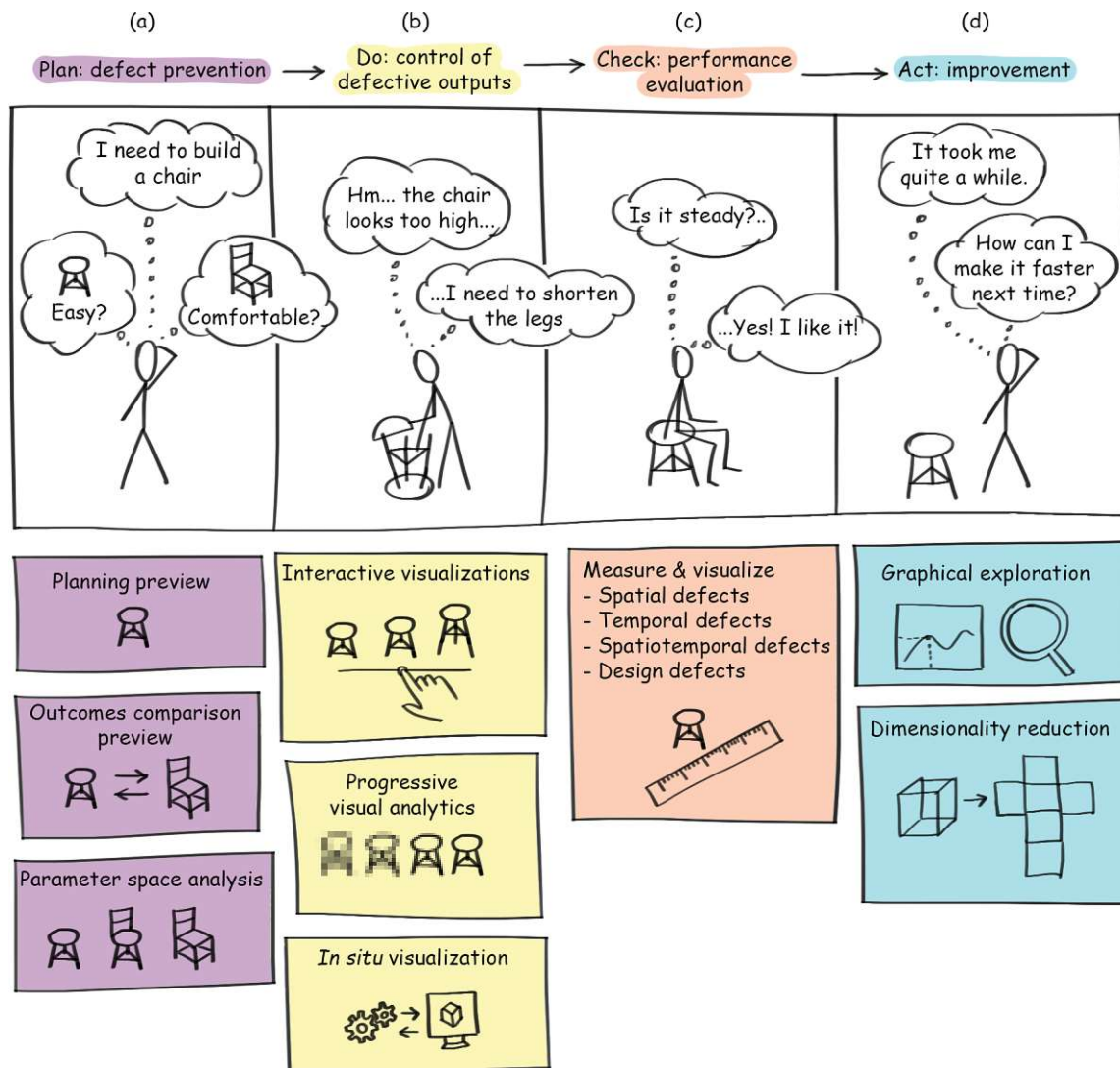
Figure 2.3: An example of the full cycle of the defect analysis workflow together with the corresponding areas of the visual analysis of defects.

the possible reasons why it took him so long and how he can build chairs more efficiently next time. John believes that if he manages to repeat the procedure with fewer mistakes, it will take him much less time. This scenario corresponds to the *improvement* step of the workflow for the visual analysis of defects. To this step, the following areas of visual analytics are relevant: graphical exploration, dimensionality reduction, and data exploration. We discuss these areas in Section 2.3.4.

2.3.1 Defect Prevention

Domain experts use defect prevention techniques to predict how the output of design and development processes will look for one or multiple scenarios (Figure 2.3 a). The techniques help identify steps to achieve desired outputs, forecast possible risks, and develop strategies to lower these risks. The domain experts' experience and knowledge are essential to estimate required resources and needed actions to achieve a goal.

Depending on the application, the number of considered scenarios is different. For example, in weather forecasting, only one scenario is possible, i.e., the user cannot change atmospheric conditions. In climate change, the user factor is significant, and various predictive models show changes based on the implemented strategies to reduce harmful anthropogenic emissions. In weather control, people can artificially adjust atmospheric conditions to meet their needs. Therefore, the number of possible scenarios is almost endless, and careful planning is required to achieve the desired goal foreseeing and minimizing negative consequences.

In this section, we group visualization techniques into three groups and discuss representative techniques. The first group previews a single scenario, increasing the awareness concerning the result. In this group, we discuss techniques for *planning preview*. The second group shows and compares results from multiple scenarios where the user can identify a preferable one. In this group, we discuss techniques for *output comparison preview*. The third group is parameter sensitivity analysis, and it allows the users to develop scenarios, preview results, and identify a strategy leading to the desired results. In this group, we discuss techniques for *parameter space analysis*.

Planning Preview

In the planning preview, the user inspects a single scenario to understand possible risks and consequences. We provide several examples from various applications to show how visualization techniques can be helpful within this context.

In medical treatment planning, Dick et al. [91] presented an interactive approach to visualize the stress distribution inside a patient's bone for hip joint replacement. The visualization shows principal stress directions and magnitudes and compares pre-treatment and post-treatment bone stress distributions of the bone (Figure 2.4). It helps domain experts to improve the understanding of complex hip joint mechanics to identify a proper prosthetic shape.

In weather forecasting, Wang et al. [272] proposed a visual analytics system to support domain experts. Domain experts often simulate atmospheric characteristics using a weather prediction model at multiple spatial resolutions to capture different scales of physical features. The proposed system employs multiple visualizations to support domain experts and improve their understanding of the correlation between parameter settings of the models and simulation results. These visualizations are nested parallel coordinates, heat maps for ensemble quality overview, and geographic representations.
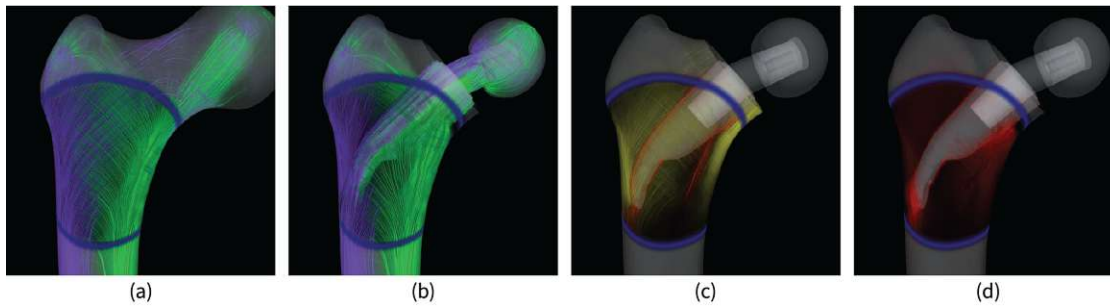
Figure 2.4: Interactive visualization of a patient's thigh bone for implant planning shows principle stress directions for (a) a pre-treatment case and (b) simulated implant surgery. The visualization also shows changes of (c) normal stress and (d) shear stress between the pre-treatment and post-treatment options [91]. © 2009 IEEE.

In materials science, Kratz et al. [173] presented a method and a case study for a tensor field exploration of mechanical parts. The method aims at assisting engineers in identifying weak spots of mechanical parts and improving their design to sustain higher stress before breakage. The visualization part of the method includes stress tensor lines, which are computed from simulations of the part's breakage process. The case study's outcomes show that the visualized tensor lines supported engineers in improving the design of the brake lever part. More precisely, engineers were able to design structures supporting the brake lever to increase the sustained stress level.

Output Comparison Preview

When multiple scenarios of a process are possible, domain experts preview and compare their outputs using visualization techniques to identify the preferable scenario. In this section, we provide several examples of comparative visualizations for multiple applications.

In radiotherapy treatment planning, clinicians seek strategies to irradiate tumor regions while avoiding healthy tissues. The patient's organs, such as the bladder and rectum, can change shape dramatically over time, making treatment planning challenging. Furmanová et al. [117] proposed a tool that can predict changes in a patient's organ shape and preview and compare multiple treatment plans with regard to these changes. The tool predicts a patient's organ shape using previously collected data from patients who underwent a similar radiotherapy procedure. Afterward, clinicians use the predicted organ shape to compare previously developed multiple treatment plans and identify the most appropriate one.

In climate research, Nocke et al. [209] presented SimEnvVis, a visualization framework for exploration and evaluation of climate simulation data. SimEnvVis provides several methods to support the inter-comparison of two simulation runs. The first method uses heatmaps and transfer functions to show the difference between two
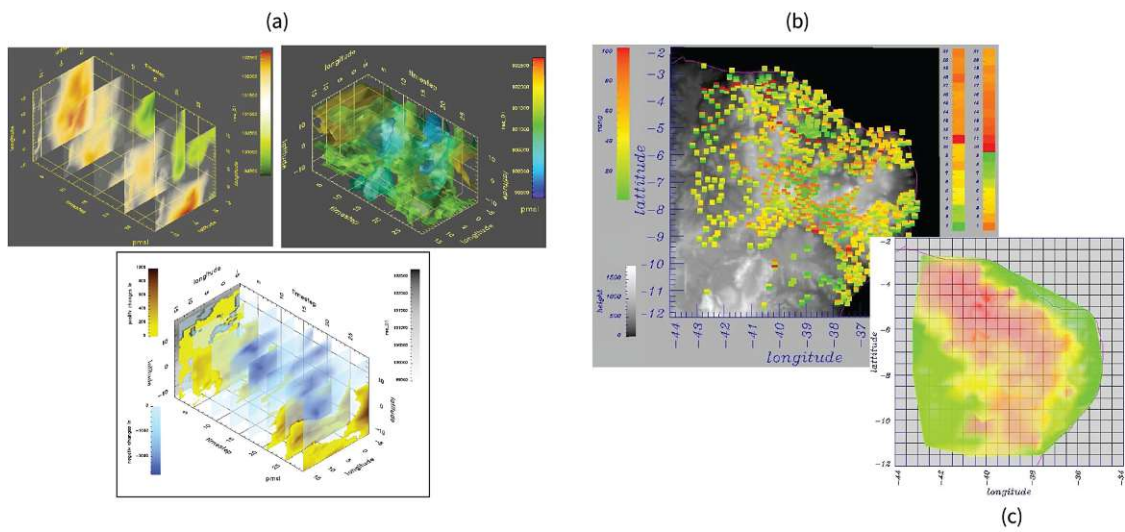
Figure 2.5: (a) 3D comparison of spatio-temporal sea level pressure conditions. Comparison of potential yield loss risk for maize using (b) square glyphs and (c) color-coding [209]. © 2007 IEEE.

simulations (Figure 2.5 a). The second technique employs square glyphs overlaid on a spatial map and color coding (Figure 2.5 b c). For glyphs, the top part represents one simulation run, while the bottom part represents another one. The third approach shows juxtaposed isosurfaces where domain experts can see both of them simultaneously. Additionally, it is possible to compute the differences in the isosurfaces' geometries and color-code the variation on the isosurfaces.

In materials science, Amirkhanov et al. [33] introduced a tool for finding an object placement inside an industrial CT device that provides imaging with the lesser artifacts. The tool discretizes possible object placements inside the device. Then, it simulates CT images for each object placement and performs penetration length and Radon-space analyses. Later, the tool visualizes analysis outcomes in 2D plots to help domain experts to analyze and compare images produced different object placement.

Parameter Space Analysis

Parameter space analysis performs multiple simulations to explain and study the behavior of a mathematical model. These simulations can be performed for one or multiple variables that define a model's state. The user specifies a range for every variable in which the model should be explored. Afterward, the models' variables are sampled, and multiple simulations are performed. Many individual simulations provide a rich outline of how the individual variables affect the object's performance. Often, these simulations must be filtered before being displayed to the user to avoid visual clutter. The user explores the simulations and identifies the most relevant variables where the object reaches its maximum performance.

Figure 2.6: Tuner uses three types of plots to show (a) the response value, (b) the uncertainty in estimation and (c) the expected gain from additional sampling [260]. © 2011 IEEE.
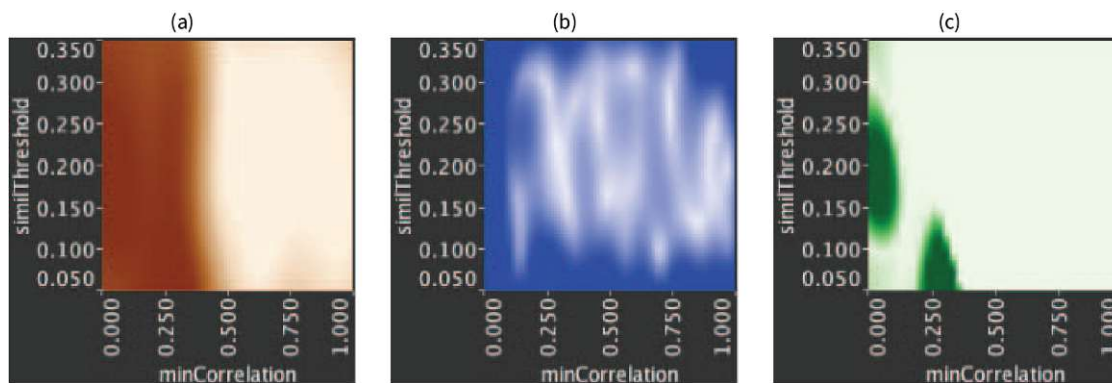
Pretorius et al. [222] introduced Paramorama to support life scientists in finding optimal parameters for image analysis algorithms. Paramorama starts with a user-supervised sampling of the image analysis parameters. The sampling parameters are hierarchically clustered and, together with their corresponding image outcomes, are visualized in three linked views. A node-link view provides an overview of the clustering hierarchy and serves as the starting point for the parameter space analysis. A refinement view shows a detailed representation of user-selected nodes in the node-link view. A reference image view supports the comparison of transformed images obtained during the analysis with an input image. The views support domain experts in a visual exploration of an algorithm's parameter space and improve the total quality of the analysis results. Additionally, Paramorama decreases the required user time and memory load to find image analysis parameters yielding desired results.

Torsney-Weir et al. [260] introduced Tuner to find parameters of image segmentation algorithms that provide closest ("good") results to the ground truth. Tuner replaces a tedious and manual procedure of finding optimal image segmentation parameters. To achieve this goal, Tuner employs a two-stage procedure. The first stage sparsely samples the parameter space and applies a statistical model to estimate a segmentation algorithm's response. The response value indicates how "good" the segmentation is. In the second stage, the user interactively explores the parameter space to identify areas where a segmentation technique provides "good" results. The user can interactively explore the parameter space by analyzing plots depicting the response value (Figure 2.6).

Raidou et al. [224] presented an approach to visually explore models for tumor control probability, which are commonly used in radiotherapy treatment planning. These models are statistical models that describe how a tumor is going to be treated (e.g., radiation dose distribution). The analysis of these models supports clinical users in an effective treatment planning that minimizes the exposure of surrounding healthy tissues to radiation. To support and improve the analysis, the presented approach incorporates the following

modules: (i) module for exploration of imaging-induced uncertainty (e.g., from magnetic resonance imaging scanners); (ii) module for parameter sensitivity analysis of tumor control probability models; (iii) module for inter-patient response variability; (iv) module for identifying a treatment plan starting from the desired result. The parameter sensitivity module inputs a range of model parameters for tumor control probability and computes the corresponding model outputs. The model sensitivity is visualized as a histogram that can be used to select and highlight areas in the image that satisfy specific criteria (dose or tumor control probability). These highlighted areas incorporate information concerning the variability as well.

2.3.2   Control of Defective Outputs

During the design and development process, domain experts use visual analytics techniques to identify and correct situations instantly leading to possible defects (Figure 2.3 b). These techniques should provide fast feedback to ensure that domain experts can react in time. Therefore, they are often interactive, employ fast calculations, and use real-time visualizations.

This section discusses the most common visual analytics techniques that support the control of defective outputs. In this section, we group visualization techniques into three groups and discuss representative techniques. The first group includes *interactive visualizations*. The second group discusses the topic of *progressive visual analytics*. The third group comprises *in situ visualization* approaches.

Interactive Visualizations

Interactive visualizations represent an investigated object and allow the user to adjust its state interactively. Using these visualizations, the user can generate and validate hypotheses on the fly by modifying the object state and evaluating how it affects its performance. In some cases, interactive visualizations track user interactions and incorporate the previous object's state for comparisons.

Schkolne et al. [245] presented a virtual reality application for designing DNA components such as double-helices, singe strands, molecules, etc. The application is taking advantage of tangible 3D input devices to create and edit DNA. While shutter glasses render a DNA model, which is floating above a table surface. Applying previous achievements in molecular visualization, the presented approach uses a virtual reality environment to design DNA structures. The outcomes showed that users often preferred the virtual reality environment to sketch DNA structures compared to the traditional manual approach as it provides a more satisfying experience.

De Llano et al. [86] presented another application for designing DNA nanostructures with a major focus on automatizing time-consuming, tedious operations. The presented approach allows combining and reusing existing nanostructures to create new ones and gives several visual representations (called scales) that work best for different tasks. The application makes it possible to smoothly transit from an atomic scale, where individual

atoms are visualized, to a high-level scale, where individual double strands are abstractly represented. The application incorporates real-time feedback on structural stability into a model and enables the design of DNA nanostructures on the fly, identifying weak structural elements.

Aliaga et al. [26] proposed a system for procedural generation and instant rending of architectural elements. The system can generate completely new buildings or modify existing ones on the fly. The user only needs to draw building blocks, and the system will automatically complete the building filling-in all necessary details in a preferred style. The interactive visualization allows the user to preview the result immediately and perform design modifications until the desired outcome is achieved. In general, the system simplifies the architecture visualization and facilitates fast content creation for various applications.

Krekel et al. [174] introduced a pre-operative planning system for shoulder replacement. The system considers the patient's unique anatomy derived from CT images and computes the corresponding bone-determined range of motion. The range of motion determination is highly efficient and based on the shoulder's joint collisions. Additionally, the range of motion is visualized to the user with motion envelops that indicate extreme post-operative shoulder positions. The system allows to adjust the prosthesis' design in real-time, and the consequence of every change can be immediately previewed in the visualization. This enables to determine the optimal shoulder replacement prosthesis taking into account patient-specific issues.

Cornel et al. [76] presented a visual analysis system of various flooding scenarios. The system combines simulation, analysis, and visualization modules and can interactively support domain users in decision making. The system utilizes a fast and efficient GPU implementation of water simulation using the finite-volume method [147]. The water simulation results are given on rectangular adaptive grids, which are prone to contain undesired artifacts due to discretization. These artifacts can sometimes be misleading to domain experts leading them to wrong conclusions. The presented system employs a novel surface reconstruction method for water height that reduces these artifacts. In general, the system enables a flood risk analysis and the design of protective measures by computing and visualizing flood maps (Figure 2.7).

Neugebauer et al. [208] presented a visual exploration system for blood flow inside vessels. The system helps to identify pathological vessels by analyzing anatomically guided blood flow characteristics at three different scopes. At a global scope, the inflow and outflow characteristics are explored. At an ostium scope, the flow characteristics and profiles are explored close to the ostium. At the local scope, flow characteristics in the parent vessel and inside the aneurysm are explored. The method can work with measured and simulated data, which is modeled as a Newtonian fluid [67]. The visual exploration and analysis of blood flow contribute to an improved risk-reduced treatment planning.

Figure 2.7: Visualizations of flood simulations with smoothly interpolated water boundaries [76]. © 2019 The Author(s) Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

Progressive Visual Analytics

With the increasing size and complexity of data, efficient analysis approaches are becoming more critical. Progressive visual analytics aims at handling big and complex data by including the user early in the exploration and analysis process [37]. At the starting point, the user can obtain first meaningful results, which are later improved with time. Progressive visual analytics methods either subdivide data into blocks or processes into batches. In the first case, initially, the user sees only a particular part of the data, which is later increased in time. While in the second case, the user can see the entire data but in a lower quality. The quality constantly increases with computations performed in the background.

Cui et al. [80] presented Sherpa, a system for computational steering utilizing the user's data exploration behavior. The system includes progressively updating visualizations, steering panel to start, stop, pause computations, and the data space view. The data space view depicts the data in a compressed 1D or 2D spatial representation. It enables data navigation and orientation, i.e., the user can zoom and pan to focus on specific data regions. Based on the current navigation region, the system prioritizes computations concerning data located inside the region and gives a lower priority to data outside of the navigation region.

An essential advantage of progressive visual analytics is that it provides early visualizations of results. However, many users of these systems are not visual computing experts, and they do not know how reliable the intermediate results are. To this end, Angelini et al. [37] introduced ten quality indicators for understanding intermediate analysis outcomes. These indicators compute progress, stability, and certainty for underlying computations and final visualizations.

Jo et al. [158] presented another approach to manage uncertainty arising from progressive data exploration using safeguards. During the data exploration, the user can build hypotheses based on the currently observed results and assert them by creating safeguards. For example, if the user analyzes a large car repair dataset, the user can observe that most
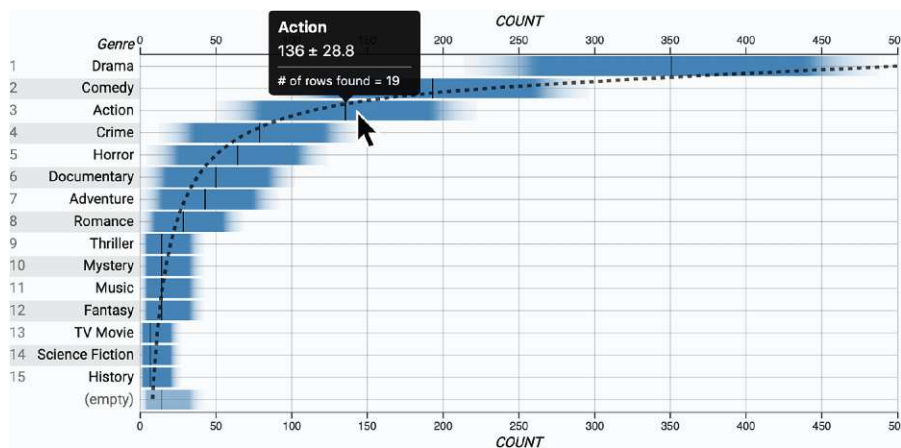
Figure 2.8: ProReveal uses gradients plots to show the uncertainty in intermediate results [158]. The figure is licensed under CC BY 4.0.

repairs are related to seat belts. In this case, the user can create a safeguard asserting this fact. However, if the most common defects will be replaced by airbag-related ones when more data is processed, then the user will be notified by the safeguard. With increasing data certainty that comes with time, the system automatically verifies safeguards and reports to the user if they are becoming invalid. Safeguards allow the user to ensure the correctness of conclusions and support data understanding (Figure 2.8).

In situ Visualization

In situ visualizations allow the user to plug into running simulations and visualize the intermediate results [46, 234]. The simulations compute information concerning a model state, which helps the user to predict possible undesirable outcomes and identify strategies to circumvent them. The simulations are performed in a parallel thread, allowing the user to interact with the system freely. For example, the user can steer the running simulations and preview changes in the simulation immediately.

Dutta et al. [94] presented an *in situ* technique to analyze flow instability in turbine engine compressors. Using the technique, domain experts can detect rotating stalls that can be destructive to the engine and consequently develop technologies to suppress them. The technique runs simulations to compute important statistical data of turbine aerodynamics from which anomalous regions are detected. These regions can indicate the inception of a rotating stall. The statistical data is visualized using uncertain isocontours in 3D (Figure 2.9). The combinations of extensive simulations and visualizations allow experts to perform the flow instability analysis efficiently with reduced cost of used storage space and fast calculations.

Woodring et al. [287] presented a system for an *in situ* analysis for high-resolution ocean models. The system enables an eddy analysis, which would not be possible otherwise due to the hardware limitations in storage space and input/output speed. A simulation
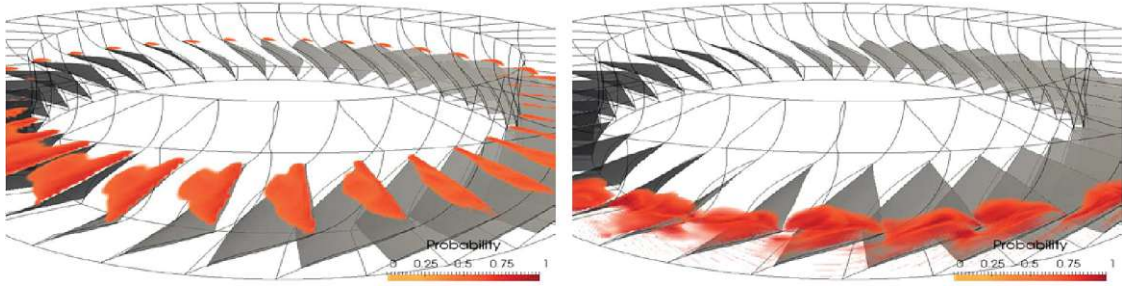
Figure 2.9: Uncertain isocontours are used to identify stall impacted regions [94]. © 2017 IEEE.

framework MPAC-Ocean computes the ocean system state, which the system employs to extract eddy fields using parallel computing. The system computes the analysis statistics from the edgy field and visualizes them in 2D plots (bar charts, line plots) and map overlays. The system improves the edgy analysis's performance and speed and allowing domain experts to work with large spatial grids.

Ayachit et al. [40] presented an interactive visualization toolkit ParaView Catalyst that provides a programmable interface to perform *in situ* visualizations. ParaView Catalyst allows users to perform simulations, generate and visualize intermediate data. The toolkit can be used for a wide range of applications such as analysis of ocean systems, rotorcrafts, or studying the atmosphere.

2.3.3 Performance Evaluation

Domain experts use performance evaluation techniques to measure output characteristics and ensure output conformity to the requirements (Figure 2.3 c). The main aim of performance evaluation is to verify output quality before it is delivered to the customer. During the verification, domain experts measure and characterize distinct object or defect features (e.g., size, weight, stiffness). The characterization methods depend primarily on the defect type (Section 1.1), as they require different approaches to be analyzed. Different characterization algorithms should be adopted for various applications. This section outlines and describes the most common characterization approaches arranged by feature type: *spatial*, *temporal*, *spatiotemporal* and *design space defects*.

Spatial Defects

Spatial defects can be characterized as objects in a geometrical space. The most common geometrical space is the Euclidean space, which can have various dimensions (e.g., 2D, 3D, 4D). Several structures can describe spatial defects, such as surfaces, binary images, or parametric models. The shape of a defect is often of analysis interest for the user, and it can be expressed by many metrics such as volume, roundness, and surface. Spatial defects are typically located inside the analyzed object, and their locations are essential
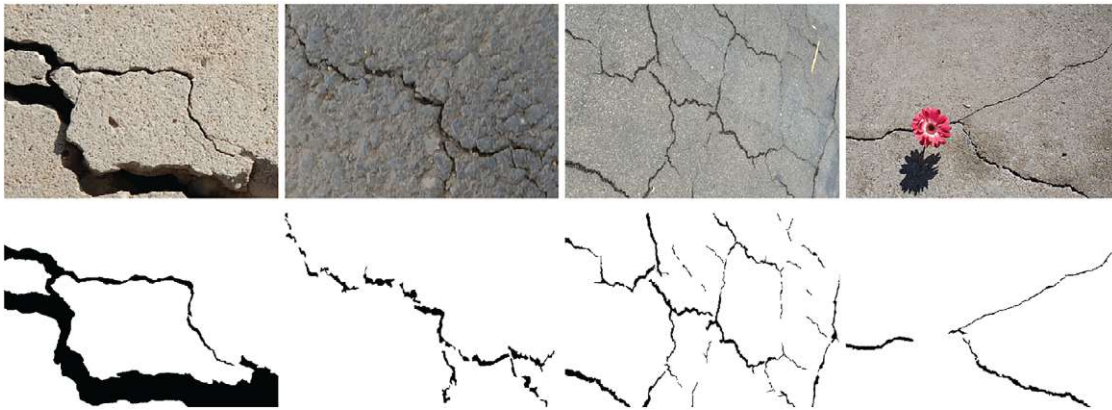
Figure 2.10: DeepCrack segmentations results [185]. © 2019 Elsevier B.V. All rights reserved.

to the user as they might contribute differently to the object's performance. When an object has many spacial defects of similar type, the distribution of defects is often used for characterization.

Segmentation is often used to identify and extract spatial defects from imaging data such as CT, MRI, or ultrasound. Segmentation is employed to identify regions or objects of interest in the original data with distinguishable features (e.g., structure, material, shape). Segmentation results can be represented as binary images or surface models. Often surface models are derived to identify structures of interests when subpixel or subvoxel accuracy is needed. In this case, the surface models are not limited or contracted by a regular pixel or voxel grid. Depending on the modality an image can have one or several data components. For example, CT images have one component (intensity) that reflects physical material properties, while color images such as photographs have three channels (red, green, and blue). Some of the most common and well-established segmentation techniques are thresholding [246], watershedding [270], region-growing [21], and neural network-based methods [214].

Wakaf and Jalab [271] presented a histogram-based method for defect detection in images. In contrast to the Otsu method, the proposed method can identify defects in grayscale images with unimodal, bimodal, and multimodal distributions. The method computes the background mask and sharpens the original image. Then it subtracts the mask from the sharpened image and calculates the histogram. Later it finds the maximum in the histogram value and uses it as the threshold value to separate defects from the background.

Liu et al. [185] presented an approach to segment cracks in images. The approach is based on a deep hierarchical convolutional neural network and can segment crack regions with a pixel-wise accuracy (Figure 2.10). Guilded filtering and conditional random fields are used to refine the predicted outcome of the neural network.

Connected component analysis is used to find connected structures in segmented binary images [244]. During this analysis, the image is traversed to assign to every pixel a corresponding connected component label. These labels are unique for every connected component and can extract a corresponding binary image for the component. Connected component analysis facilitates the quantitative analysis as it allows to numerically compute the number of individual defects such as micro-cracks, voids, or material inclusions in images.

Model-based segmentation is used to extract structures from the original data by defining a corresponding parametric model that describes them [95,112]. This model may consider the statistical variability of the structure and includes some assumptions concerning the defect shape. For example, some defects can be described as elliptical structures. Statistical information on size and proportions of the elliptical structures can be assumed or derived from the sampled datasets. Later, this parametric model is used to detect defects by adjusting parameters until it matches the structure presented in the image. Parametric models can be used to quantify and describe these structures for future analysis.

Temporal Defects

Temporal defects are features associated with time, and their presence at specific time points contributes to the object's performance. For example, these features can be the engine cycle number per minute or pipe pressure collected over time indicating a system's nonconforming behavior. Temporal defects depending on their connection to the time can have different aspects such as arrangement (linear / cyclic), scope (point-based / interval-based), scale (ordinal / discrete / continuous) [24]. There are many temporal defect visualization techniques such as tracking graphs, tile maps, and arc diagrams. Most of these techniques can be applied to a wide range of applications, including the visual analysis of defects.

Havre et al. [138] presented the *ThemeRiver* visualization to show trend and theme variations over time for an extensive collection of documents. The presented visualization uses a river metaphor to organize data in a timeline, which helps users reveal and understand data trends and patterns.

Mintz et al. [202] showed a series of Ozone measurements in *tile maps.* Every individual tile represents one day and indicates the corresponding measurement by color. Lighter shades indicate lower concentrations, and darker shades indicate higher concentrations. Tile maps support the identification and revealing of seasonal patterns. For example, using tile maps, it is possible to see that Ozone readings are much higher in summer.

Saito et al. [240] presented a method for visualizing one-dimensional time-oriented large-scale data. The method starts from a line plot and divides it into a small number of equally sized bands. Later, these bands are overlaid with transparency. Negative line plot values are mirrored and depicted with a different color (Figure 2.11). This representation of time-oriented data is compact and saves a significant amount of horizontal space.
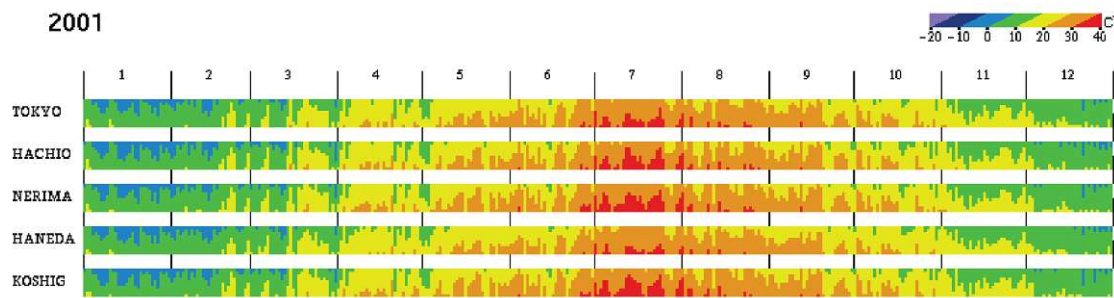
Figure 2.11: Compact line plots show the maximum temperature in different areas of Tokyo [240]. © 2005 IEEE.

Spatiotemporal Defects

Spatiotemporal defects combine both spatial and temporal defect types [24]. They can be described concerning space (e.g., location, position, orientation) and time (e.g., beginning and end time points). Often the correlation between time and spatial defect characteristics is interesting to analyze. For instance, during an in-situ tensile test, a test specimen is exposed to an increasing load. The load starts from a slight tension and is subsequently increased until the specimen is torn apart. During the in-situ tensile test, the specimen is measured using an ultrasound or CT device. Later the number of defects, their shape, and time of occurrence are analyzed to investigate the specimen properties. Frequently image registration is necessary to analyze spatiotemporal defects.

Image registration aligns images by bringing them into the same coordinate system. Image registration is typically performed by finding matching features in two or more images and computing a transformation that brings them into the same space in the new coordinate system. Transformations bringing images to the new coordinate system, can be linear and nonlinear. Linear transformations include rotation, scaling, translation, while nonlinear transformations comprise more complex deformations.

Jung et al. [160] presented an approach to register dental X-ray CT images containing rich information concerning inner dental structures, with 3D dental impression models with highly accurate dental surfaces. This approach supports modern computer-aided design software in dental treatment planning. For image registration, the approach first identifies dental surfaces in the X-ray CT image. Next, it uses an initial manual dental impression placement as an approximate registration. Last, this registration is refined using an iterative closest point algorithm.

Cao et al. [64] introduced a method for inspecting surface defects and dimension measurement. The method compares a reference image with the tested product images and identifies differences. For performance reasons, the tested image is separated into two areas: one containing the background and another one containing the complex object. This is the area with special patterns. To determine differences and identify
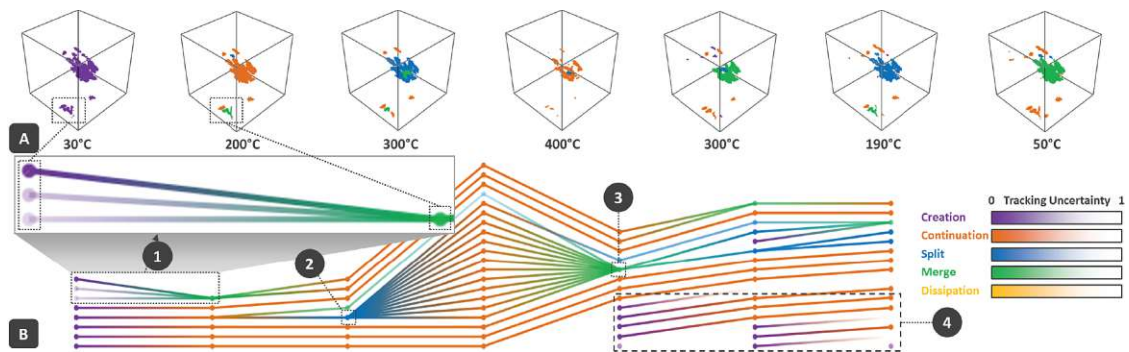
Figure 2.12: The fuzzy feature tracking graph shows the development of voids in an industrial material during a heating process [228]. © 2015 Elsevier Ltd. All rights reserved.

defects in areas with special patterns, image registration is employed. It is based on feature extraction and identifying matching features. Defects in the background area are identified using a gradient thresholding algorithm.

Feature tracking in multiple images allows characterizing an internal process of the tested object. Feature tracking starts with feature extraction, which can be accomplished in various ways, such as segmentation. The features in different images are correlated using a linear or nonlinear deformation. Then the correlated features are visualized and analyzed, supporting pattern discovery and knowledge discovery.

Reinders et al. [231] introduced a method for feature extraction, tracking, and visualization for time-oriented data. The method additionally can determine significant events or stages in the data (such as continuation, birth/data, or entry/exit). The method employs a graph viewer, a feature viewer, and two-linked views to visualize features over time. The graph viewer shows features in a graph as nodes, their relationship as edges, and event as icons. In the graph, the horizontal axis corresponds to time. The feature viewer shows features in 3D enabling interactive data exploration. The two-linked views connect the graph view and the feature view by interactive user selection and applying these selections in both visualizations.

Reh at al. [228] characterized internal and external changes of a test specimen (Figure 2.12). It enables tracking internal features in a series of X-ray CT scans. These features can be material cracks caused by an increasing thermal load or pores in natural materials induced by a drying process. The approach computes related features across multiple images. It visualizes them in a fuzzy feature tracking graph, listing all individual images features and showing their relationships, which can be either creation, continuation, split, merge, or dissipation.

Design Space Defects

Design space defects are associated with a specific set of object characteristics that prevent the object from performing correctly. For example, induced mechanical resonance can cause object destruction or a system failure. For example, the broughton suspension bridge was collapsed in 1831 due to mechanical resonance produced by soldiers walking in step. Design space defects often cannot be described as spatial structures.

Dentures are prosthetic devices replacing missing or damaged teeth. In clinical dental practice, articulators are employed to enhance denture design by simulating jaw movement and biting patterns [169]. This helps to identify and fix possible functional denture defects by analyzing complex static and dynamic occlusal relations. Articulators are adjusted for every patient individually to reproduce unique jaw movements. Nowadays, traditional mechanical articulators are being replaced by virtual equivalents, which are based on computer-simulated models. Virtual articulators have several advantages over mechanical ones: (i) they can reproduce jaw movements more accurately; (ii) they can work with digital denture models and do not require a physical denture prototype; (iii) they benefit from integration with modern computer-assisted design software (e.g., adjusting the denture shape can be performed interactively and validated immediately with the virtual articulator). Virtual articulators are integrated into modern computer-assisted design software and are often coupled with computer-assisted manufacturing (e.g., 3D printing, milling). The results of denture adjustments with virtual articulators can be previewed and evaluated concerning aesthetics [30].

Many software development organizations require to predict the number of defects and possible faults in their products [107]. This information allows organizations to improve the total product quality by discovering and eliminating software defects. Often, statistical models use size and complexity metrics to predict defects, and they are based on testing data. For example, El Emam et al. [98] developed a model that uses object-oriented design metrics to predict software defects. For model development, a commercial Java application was used. The validation was performed on subsequent releases of the same application. During the development, suitable metrics, which are associated with fault-proneness, were determined. These metrics are inheritance and export coupling. Based on the metrics, the resulting model could predict fault-prone Java classes and estimate the overall system quality. The prediction model achieves a good accuracy. However, the model does not account for fault severity.

Nuclear factor kappa-light-chain-enhancer of activated B cells (NF-κB) is a family of dimeric transcription factors. Mutations or defects in NF-κB pathways contribute to many human diseases such as cancer or immunodeficiency [78]. Mathematical modeling is often used to describe and analyze the NF-κB pathways. These models can be described through ordinary differential equations. The analysis of these models supports knowledge discovery and a better understanding of the NF-κB pathways and their regulation mechanisms. Krishna et al. [175] presented a reduced NF-κB model to analyze the spiky oscillations of the network, which contains only three variables. The model is
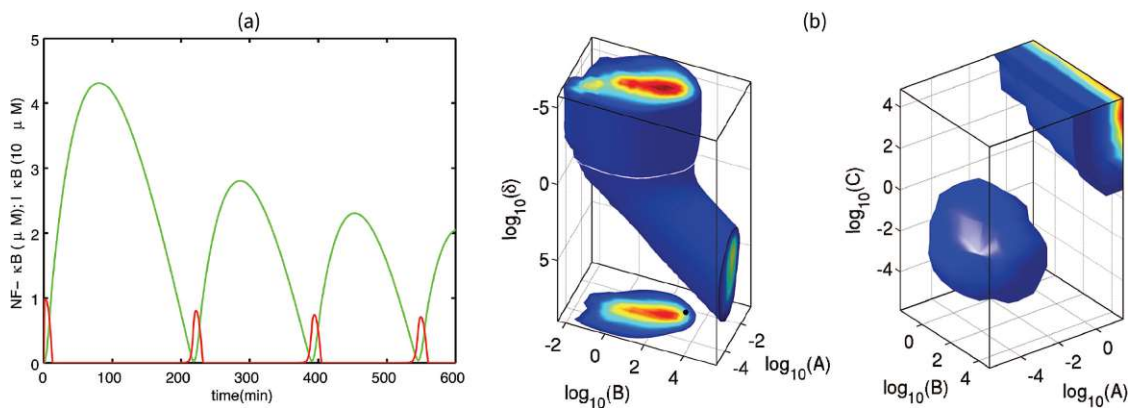
Figure 2.13: (a) A 2D plot shows NF-κB oscillations, and (b) 3D plots show robustness of spiky oscillations [175]. © (2006) National Academy of Sciences, U.S.A.

visualized with 3D volumes and various 2D plots to visually describe the NF-κB pathways (Figure 2.13). 3D volume plots are shown as phase portraits, where axes depict the variables of the model's ordinary differential equations. 2D plots are shown as both phase portraits and time series plots, where one axis indicates time.

2.3.4   Improvement

Domain experts use visual analytics techniques to improve the output quality in subsequent iterations of design and development processes (Figure 2.3 d). The improvement is made by adjusting the plan, do, and act steps of the PDCA cycle. To identify the direction of improvements, domain experts study and analyze new knowledge and experience gained in the last iterations. Often, design and development processes are complex and include many factors influencing the results. Therefore, models of high complexity are often used to describe the processes. Domain experts increase the understanding of these processes by employing visual analytics to identify new possibilities for improvements.

In this section, we describe common visual analytics techniques aiming at the improvement step of the visual analysis of defects, by categorizing them into two groups and discuss representative techniques. The first group discusses techniques for *graphical exploration*, based on the data type. Here, we include techniques with *single* or *multiple linked views*, as well as *interaction-based techniques*. The second group comprises techniques that deal with higher complexity through *dimensionality reduction*. Here, we discuss *feature selection*, as well as commonly employed strategies such as *principal component analysis*, *t-distributed stochastic neighbor embedding*, and *autoencoders*.

Graphical Exploration

Users visualize and analyze data and defects to recognize weak spots of design and developing processes and adjust them correspondingly. Visual analytics systems employ different graphical representations based on data type (e.g., 2D images, volumetric images, multidimensional data, time-oriented data, tensor data) and analysis goals (e.g., exploration, dissemination, solving an analytical task). In this section, we group techniques for graphical exploration into three groups. The first group, *single views*, includes techniques that use only one view to show the entire data. These techniques prioritize the clarity of representations. The second group, *multiple linked views*, employs several views showing distinct aspects of the data. These techniques fit well to show complex and dynamic data. The third group, *interaction-based techniques*, requires user interactions to get use of the visualization (e.g., magic lens). These techniques are typically tailored to a specific application, and they can solve tasks very efficiently (e.g., local image comparison).

Single Views

Various 2D plots can be employed to show data features, correlations, and trends. 2D plots provide a trade-off between the quantity and complexity of the displayed information and the simplicity of the visual representation. Often they can be extended to higher dimensions (e.g., 3D, 4D), but this causes additional challenges of either identifying an appropriate viewing direction or finding a suitable projection of the data back to the screen space. Nowadays, there is a wide variety of 2D plots [99, 108, 138], such as scatterplots, heatmaps, histograms, graphs, radar charts, and many others (Figure 2.14 a b). All these techniques have their strengths and weaknesses. Therefore selecting a 2D plot has to be done based on the data type and structure. For example, for time-oriented data 2D plots show temporal changes better than graphs or line charts.

Superimposed views combine several independent channels of information in one integrated representation [122, 123]. Often these channels contain different information and could be rendered separately. However, combining or superimposing these channels helps the user correlate information distributed in different channels and consequently enhance data understanding. Superimposed views can be rendered in two steps. In the first step, all information channels are aligned to the same coordinate system. In the second step, channels are blended to compose a new combined representation. Various blending methods emphasize different aspects of the data. For example, channels can be mixed using opacity, pixel intensity, or color saturation. In the next paragraph, we present several examples that employ superimposed views to visualize information concerning defects.

Parallel coordinates are used to visualize multivariate data [141, 152, 225]. Instead of plotting n-dimensional data as points in the corresponding n-dimensional space, they represent data as polylines passing through n vertically or horizontally aligned parallel axes (see Figure 2.14 c). Every parallel axis corresponds to a specific data dimension, and the location where a polyline intersects the axis designates its value in the corresponding
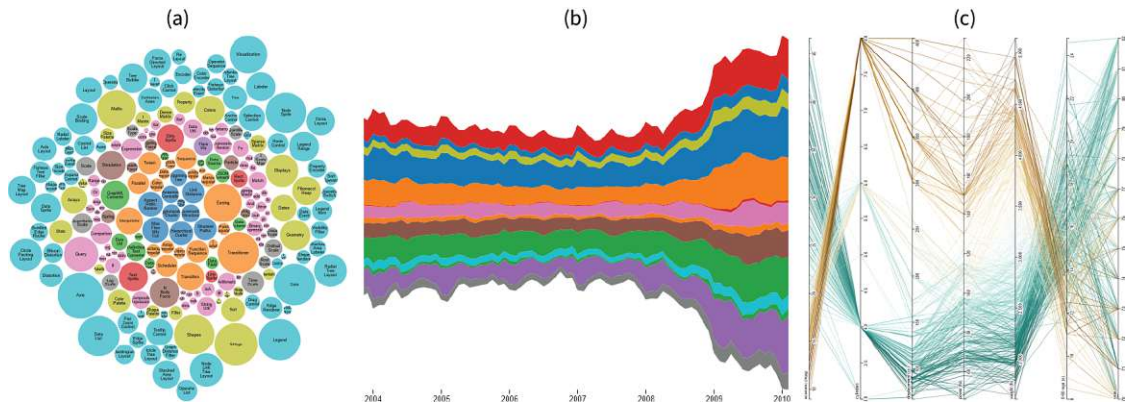
Figure 2.14: Example provided by the D3 library [57] of (a) bubble chart, (b) stream graph and (c) parallel coordinates.

dimension. Parallel coordinates visualize multivariate data in a single view and support the user in detecting data correlations and patterns. Different color encodings of the polylines can be used to enhance the visual representation of parallel coordinates. For instance, color can indicate a polyline feature, such as the value in a specific dimension. Additionally, parallel coordinates can be used for interactive data selection. This is done by defining a range on the parallel axes or by picking individual polylines. An essential advantage of parallel coordinates is that they linearly increase the complexity of the visual representation with the number of dimensions [152].

Immersive graphical exploration techniques provide exciting opportunities to interact with the data. Sorger et al. [250] proposed a system for the interactive exploration of large dynamic networks in virtual environments. Using a web-based 3D rendering engine and a virtual reality headset, the system gives a navigation method for large graphs. The graph is represented by a commonly used node-link diagram, where spheres denote nodes, and elongated cylinders connecting two spheres represent a relationship between the nodes. The method includes an overview exploration to see the whole graph and an immersive detail analysis on demand. To get an overview, the user needs to select a node using a virtual reality controller. The user will then be immersed in the graph by moving the camera inside the selected node.

Transfer functions are typically 1D or 2D functions that map data values to visual properties properties (e.g., color, opacity). They are used in direct volume rendering to separate different structures based on their intensity values [100, 104, 133, 186]. For example, they can visually separate soft tissues (e.g., muscles, brain) from hard tissues (e.g., bones) in medical images or pore structures (e.g., cracks, voids) from solid material structures (e.g., polymers, fibers) in industrial material images. Transfer functions map original intensity values of a volumetric image to color, which is later used by the volume rendering algorithm. Transfer functions may or may not contain information about the voxel opacity, allowing users to hide or fade out structures that are not of current research

interest. For instance, the surrounding air of an object of interest is often not desirable, and it can be removed by assigning zero opacity to the air intensity range. Transfer functions are often associated with volume rendering as they are essential for visual structure identification and separation. They can enhance the visual representation and provide an intuitive rendering. Additionally, they are valuable in knowledge dissemination and externalization to show understandable volumetric renderings to laypeople.

Fritz et al. [114] introduced a method to visually examine fiber and particle structures in CT images of composite materials. They use transfer functions to visually separate particle features (e.g., diameter or roundness). Additionally, they use directional transfer functions to distinguish fiber orientations. Direction sphere histograms are employed to visualize fiber orientation distributions.

Salama et al. [243] presented a volume rendering technique that uses high-level transfer function models to assist domain experts and laypeople in the visual analysis of data. The technique applies several primitives (trapezoids, paraboloids, and quadrilaterals) for selecting regions in 2D transfer functions to hide semantic information in the rendering. It allows building complex and, at the same time, easy-to-design transfer functions.

Multiple Linked Views

Linked views comprise several independent and synchronized visualizations of data [43, 235, 284]. Similar to the superimposed views, data for linked views can be preliminarily aligned and transformed into the same coordinate system. However, this alignment is not required. The arrangements of views can be either linear (e.g., side-by-side) or arbitrary (e.g., defined by the user interface layout). Interactions performed in one view are typically automatically applied to all other views to keep views synchronized. For example, a transformation of the camera position in one view will be automatically applied to all other views. The advantage of linked views over superimposition is that they typically do not overlay two or more information channels, which enhances the understanding of individual channels. The disadvantage is that linked view require more screen space than superimposed views.

Reh et al. [229] presented an approach to determine and visualize material porosity in X-ray CT data. Porosity reduces the material's durability, and therefore porosity level has to be carefully examined to ensure the product meets the intended quality standard. They presented a MObject visualization that superimposes defects from the whole scanned specimen or a region thereof. The visualization uses a color and opacity transfer function to fine-tune the mean porosity representation. Additionally, they employed a homogeneity visualization that overlays a map on a volume rendering of the specimen. This map shows how much the mean porosity shape in a region deviates from the whole specimen's mean porosity shape.

Weissenböck et al. [279] presented a tool for porosity analysis in CT scans of glass fiber reinforced polymers. The tool automatically segments the data with a number of different algorithms. The segmentation results are later visually analyzed. The visual analysis provides an overview on the segmentation results at different scales using an overview
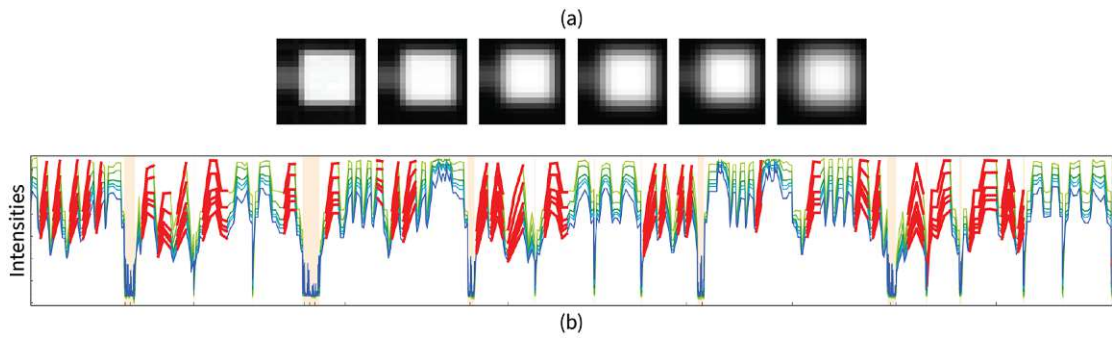
Figure 2.15: A use case shows differences between (a) multiple synthesized CT images using (b) dynamic volume lines [281]. © 2019 IEEE.

comparative matrix, scatter plots matrices, and a visual examination view. The visual examination view superimposes segmentation masks on the original data and provides an overview of how the segmentation techniques perform. The visual examination view also shows the minimum, average, and maximum porosity outlines. This allows the user to semi-automatically compare various segmentation techniques and find the most suitable one for the current analysis goal.

Amirkhanov et al. [32] presented a tool for metrological measurements and CT data analysis taking uncertainty into account. The tool computes predefined measurements of the scanned object using an extracted surface model. These measurements are visualized in linked views, which deliver information to the user concerning tolerances and uncertainties. The visualization module consists of a) measurement plots that represent measurements as lines in a 2D plot, b) 3D tolerance tags augmenting the 3D view with a symbol indicating which measurement metric is used (e.g., surface flatness, circularity, straightness), and c) a 3D view depicting the reference shape and the deviation from the target object.

Weissenböck et al. [281] presented dynamic volume lines for comparison of several CT images to identify mismatching regions. The comparison technique is based on a Hilbert space-filling curve going through each image voxel. Later, the curve's intensity profile is compared among the image set, and regions that exhibit largely varying behavior are identified. The tool consists of multiple linked views such as a Hilbert line plot and a 3D rendering view, enabling the analysis of differences in various visual representations (Figure 2.15).

Weissenböck et al. [280] introduced a tool for the interactive analysis of fiber structures in CT data. The tool focuses on the interactive exploration and analysis of fiber characteristics such as fiber length, diameter, volume, and orientation. The tool visualizes the initial volumetric image and provides visualization techniques to represent fiber characteristics. Fiber analysis needs to identify fiber classes, i.e., set of fibers having common characteristics. This is done using parallel coordinates that visualize fiber characteristics and can be employed by the user to identify fiber classes. Once the user

selects a new class, it is instantly displayed in all other views: scatterplot matrices, a 3D view, and a polar plot. Fiber clusters can further be visualized utilizing blob and metadata visualizations that augment the 3D view of the volumetric image.

Guo et al. [131] introduced a tool that supports visual exploration and analysis of complex traffic trajectory data. The tool comprises three linked views. The spatial view shows trajectories as lines. It supports multiple visual representations and user interactions. The temporal view is realized by improved ThemeRiver [138], and it shows traffic variations along the time axis. The multidimensional view applies *parallel coordinates* to support the user in analyzing advanced trajectory features such as speed and acceleration. Additionally, parallel coordinates support user interactions to brush and select interesting trajectories.

The Porosity Maps tool by Reh et al. [230] helps to analyze and explore the porosity in carbon fiber reinforced polymers. The tool targets X-ray CT images to quantify and determine the material porosity level and distribution. The porosity can be displayed in several visualizations, among which are 2D maps. These maps project segmented voids and cavities in the image to one of the axis-aligned planes (e.g., XY-plane or YZ-plane). Every pixel of the projection shows the total number of voxels belonging to voids in along the projection ray. These maps allow the user to compare the porosity levels and distributions for different specimens.

Jurčík et al. [161] presented a method for the interactive visual exploration of large sets of ligand trajectories. These trajectories help to understand protein function as they describe how protein reacts with other small molecules. The method includes multiple linked views. The chart matrix view shows trajectories in small multiples depicting various trajectory-profile properties (e.g., ligand binding energy). This view allows brushing to select individual trajectories and enable a drill-down exploration. The 3D view shows spatial properties for the currently selected trajectories. The selection chart panel view compares multiple trajectories, which are preliminarily selected in the chart matrix view. This view also allows the user to navigate previous trajectory selections to ensure a consistent analysis.

Zákopčanová et al [294] proposed a tool for the exploration and analysis of criminal data. The tool keeps track of available documents and progress history to support criminalists in the complex investigation process with high cognitive load. The system uses an overview dashboard containing all data relevant to the investigation and visualizations to provide an overview. The system also tracks all user interactions and shows it in a provenance graph, which enables the user to return to an early analysis stage at any time. A network analysis view shows objects or people and known connections between them. The network node layout is generated once when the data is loaded into the system. At any point, the user can introduce new objects into the network and assign them a credibility level (e.g., evidence, knowledge, assumption).

Interaction-Based Techniques

The magic lens is an interaction-based visualization that was initially proposed by Bier et al. [53]. It interactively combines several information channels using the visual metaphor of a moveable see-through lens [45, 252, 258]. Inside the lens, a different information channel than the outside of the lens or a magnified region is displayed. This channel combination allows a magic lens to show additional details on demand and, at the same time, to hide information not relevant to the current analysis goal. The user controls the lens position utilizing an input interface such as a mouse or keyboard. The lens typically has a rectangular or circular shape. However, it can have a more complex shape in some cases, such as a hexagon [192]. The user can also interactively control the lens size and shape.

The InSpectr tool by Amirkhanov et al. [31] is designed to analyze bi-model data where one channel is an X-ray CT image, and the second channel is spectral data. InScpectr uses multiple linked views to combine spectral and CT representations. The probing point spectrum view shows a 2D spectral plot for the user-defined point in the CT image. An aggregated spectrum view is used to define probing points for generating a combined spectral representation. Semi-transparent spectra line views provide an aggregated spectral representation for the entire data. Additionally, InSpectr enables selections in the spectral domain to highlight relevant voxels in a 3D view or pixels in 2D slices. Spectrum-based color transfer functions generate representations where color encodes spectral information in CT images. A spectral magic lens shows a spectral color image in a 2D slice. The magic lens can be smoothly moved with an input device (e.g., a mouse) over the 3D or 2D CT image. The magic lens has three different modes: overlay, offset, and side-by-side. These modes alternate the magic lens representation making it suitable for different usage scenarios.

The GEMSe tool by Fröhler et al. [115, 116] supports segmentation specialists and laypeople in exploring the parameter space of segmentation algorithms. GEMSe starts with sampling the parameter space by computing segmentation results with varying parameters. To provide an overview, segmentation results are clustered using maximum linkage. The resulting tree shows the clustering hierarchy and can be used to select cluster nodes. The detail view provides an annotated and enlarged cluster view with additional segmentation parameter information. The cluster example view provides a few cluster examples for the selected parental node. The histogram view shows input parameters and derived output. The scatter plot view indicates the correlation between input parameters and derived output. The magic lens tool enables the comparison between the original data and a cluster representation or the ground truth.

Malik et al. [192] presented a comparative tool for X-ray CT images. The tool employs various tile patterns to connect multiple images in one representation (Figure 2.16). Depending on the number of images to be compared and the analysis goals, the tile pattern can be of three different shapes: checkerboard pattern (two images), attribute blocks (four images), and hexagonal base tile pattern (arbitrary number of images). The
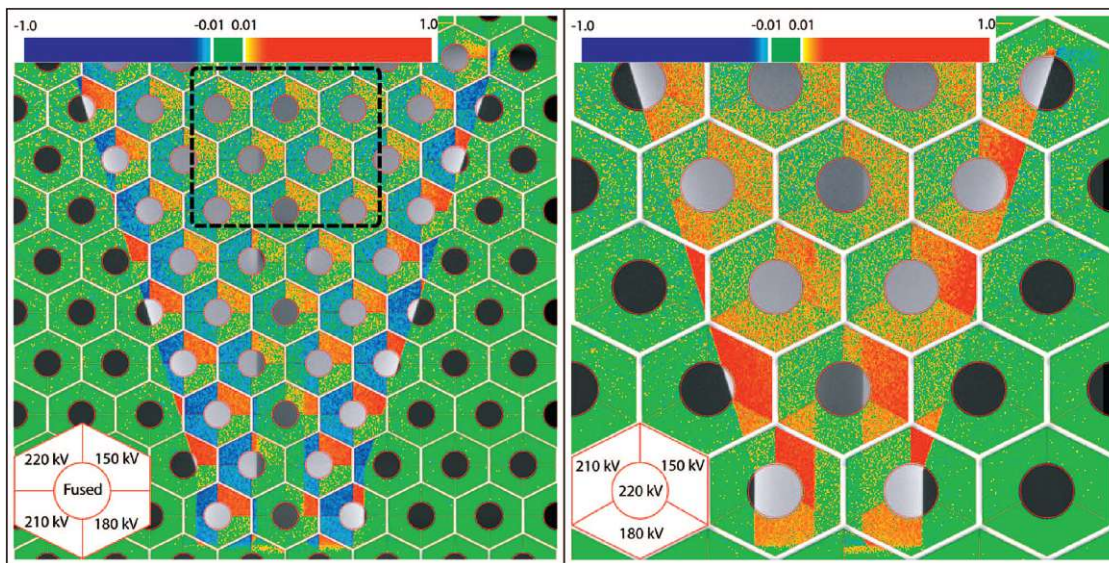
40

Figure 2.16: An example of multi-image view comparing five (left) and four (right) CT images [192]. © 2010 IEEE.

magic lens is applied to interactively display a single tile on a user-defined point. The user can regulate the size and shape of the magic lens. For the magic lens tool, quantitative information is displayed on the screen, such as mean and maximum deviations.

Dimensionality Reduction

High-dimensional data requires a big cognitive load for the user to understand. Additionally, the processing of such data is time-consuming and computationally expensive. In some extreme cases, data can contain thousands of dimensions, making it almost impossible for humans to comprehend and interpret. Also, modern technology does not scale well to visualize data containing many dimensions. Standard 2D displays often do not perform well if the number of dimensions is high. Nevertheless, many-dimensional data can still be visualized, but the mapping between the visualized data and the user's mental model is becoming intricate.

Dimensionality reduction is used if a original data visualization is too complex and a more straightforward representation is required. Dimensionality reduction techniques simplify data representation by computing a transformation of the data from the original high-dimensional space to a low-dimensional space. Thereby, dimensions can represent various aspects of the data, for example, space, time, feature, or variable. A dimensionality reduction technique can be either *linear* or *nonlinear*, depending on its assumption concerning the data [264]. Linear dimensionality reduction techniques seek a linear subspace of a lower dimension in the original data space. Nonlinear techniques search for a manifold that accurately represents the original data. In the next paragraph, we discuss typical dimensionality reduction techniques.

Feature Selection

Feature selection techniques reduce data dimensionality by identifying a set of relevant variables. The identification process can be supervised or unsupervised [69]. Supervised feature selection requires user input to identify which variables are selected for a future visual analysis. Which variables to select is application-specific and depends on the current analysis goal. With unsupervised feature selection, the set of relevant variables can be identified based on their characteristics such as standard deviation, extreme values, or mean value.

An example of supervised feature selection is provided in *FiberScout*, a tool that offers a visualization of fibers in X-ray CT images. FiberScout allows the user to detect regions with fibers of similar characteristics using parallel coordinates and scatter plot matrices. Fibers can have up to 25 or more characteristics (length, diameter, orientation, volume, etc.). To ensure scalability of the parallel coordinates and the scatter plot matrices, FiberScout employs an interactive fiber characteristic selection widget to identify only specific characteristics and disable the others [280].

A support vector machine is a statistical learning method that provides data classification and regression analysis [56]. Support vector machines are used for many tasks and applications such as medicine [39], handwriting recognition [42], face recognition [130], bioinformatics [290]. For example, Akay [25] proposed a method for breast cancer diagnosis using a support vector machine. The method showed a high classification accuracy and good performance.

Daikuzono et al. [82] presented a design for an electronic tongue device to measure the gluten level in food. The device employs four sensing units to measure food properties. The sensors produce high-dimensional data. However, the distinguishing system performance can decrease if all data points are considered. The system uses feature selection and multidimensional projection to address this problem. Feature selection was performed manually using parallel coordinates.

Principal Component Analysis

Principal component analysis (PCA) is a linear dimensionality reduction technique that computes a low dimensional representation of the data, which explains most of the variance [148, 264]. PCA finds a new linear basis, where the first coordinate has the most significant variance, and each subsequent coordinate has less variance than the previous one. Coordinates of the new linear basis are also referred to as principal components. Using PCA, the user can reduce data dimensionality by considering only a few of the first principal components and discarding the others. The user can choose the number of principal components based on desired data variability to be preserved.

Oeltze et al. [211] presented a visual analytics approach for evaluating perfusion data that indicates the regional blood flow inside human tissue. They first segment relevant data structures using techniques such as watershed and live-wire. Then they employ
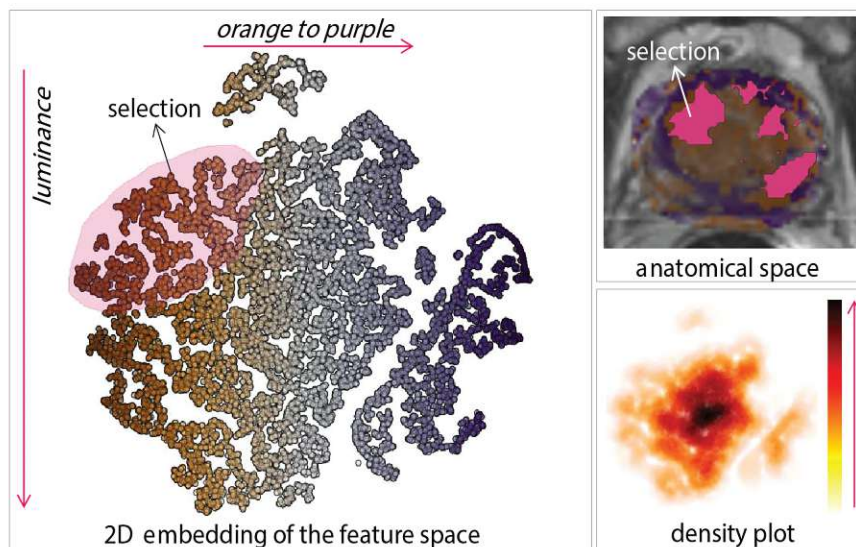
Figure 2.17: 2D embedding of the tumor features space can be used to select select semantic region in medical images [226]. © 2015 The Author(s) Computer Graphics Forum © 2015 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

a statistical analysis to detect data patterns and trends. To reduce the complexity of the multidimensional segmented data, they apply a PCA. The outcomes of statistical analyses are visualized in multiple views to guide domain experts in diagnosis processes.

t-Distributed Stochastic Neighbor Embedding

t-distributed stochastic neighbor embedding (t-SNE) is a nonlinear dimensionality reduction technique that is used for analyzing of multi-dimensional data [190]. It uses stochastic neighbor embedding [144] with a t-distribution to find a location for every data point in a two or three-dimensional map.

Raidou et al. [226] presented a tool for exploring and visualizing tumor regions in medical images. The tool first computes multiple tumor characteristics. Then it identifies those characteristics that can differentiate tumor regions from healthy tissues using t-SNE. The tool visualizes tumor characteristics to support clinical researches in tumor treatment planning (Figure 2.17).

Autoencoder

An autoencoder is a feed-forward neural network trained to minimize the error between the input and the output [27]. The autoencoder compresses data to a bottleneck layer, which contains fewer dimensions than the original data. The part of the neural network that compresses data to a low dimensional representation, i.e., the bottleneck layer,

is called the *encoder*. Another part that reconstructs the input is called the *decoder*. Autoencoders have many application scenarios such as removing noise from the original image, missing data imputation, face recognition, and dimensionality reduction.

Mei et al. [197] presented an automatic unsupervised learning-based approach for detecting fabric defects. The approach is based on optical textile inspection and can handle various fabrics and defect types such as scratches or oil spots. It uses a convolution denoising autoencoder network at multiple Gaussian pyramid levels to reconstruct image patches. The patch reconstruction is later used to predict where defects are present in the fabric with achieving pixel resolution.

Paper D: improvement

Paper A: defect prevention

Paper C: performance evaluation

Paper B: control of defective outputs
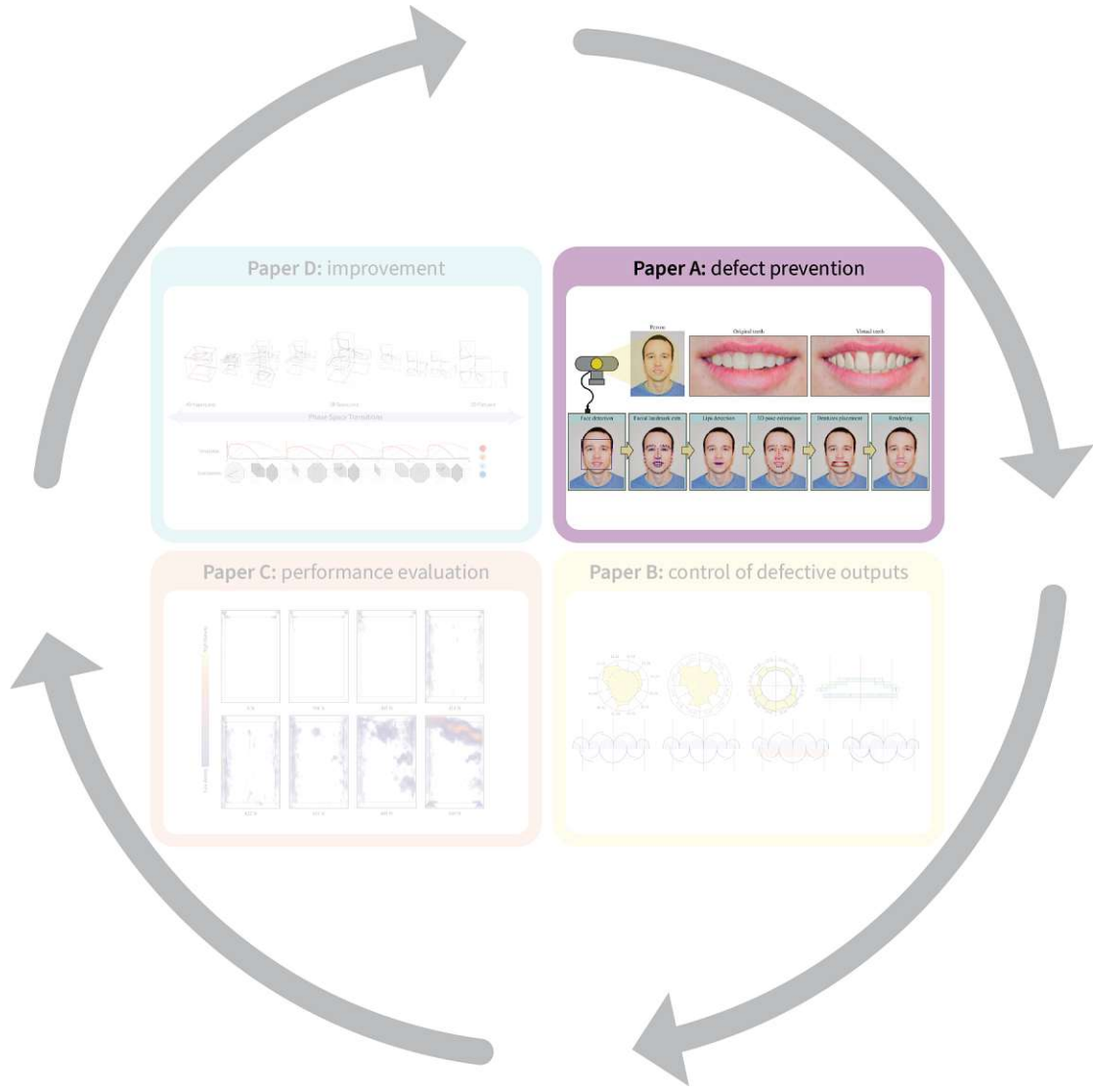
CHAPTER 3

Defect Prevention

This chapter addresses the defect prevention (i.e., the *plan* phase of the PDCA cycle) and is based on the publication:

Aleksandr Amirkhanov, Artem Amirkhanov, Matthias Bernhard, Zsolt Toth, Sabine Stiller, Andreas Geier, M. Eduard Gröller, Gabriel Mistelbauer, "WithTeeth: denture preview in augmented reality," in *Vision, Modeling and Visualization*, The Eurographics Association, 2018, DOI: 10.2312/vmv.20181250.

Dentures are prosthetic devices replacing missing or damaged teeth, often used for dental reconstruction. Dental reconstruction improves the functional state and aesthetic appearance of teeth. State-of-the-art methods used by dental technicians typically do not include the aesthetic analysis, which often leads to unsatisfactory results for patients. In this chapter, we present a virtual mirror approach for a dental treatment preview in augmented reality. Different denture presets are visually evaluated and compared by switching them on the fly. Our main goals are to provide a virtual dental treatment preview to facilitate early feedback, and hence to build the confidence and trust of patients in the outcome. The workflow of our algorithm is as follows. First, the face is detected and 2D facial landmarks are extracted. Then, a 3D pose estimation of the upper and lower jaws is performed and high-quality 3D models of the upper and lower dentures are fitted. The fitting uses the occlusal plane angle as determined manually by dental technicians. To provide a realistic impression of the virtual teeth, the dentures are rendered with motion blur. We demonstrate the robustness and visual quality of our approach by comparing the results of a webcam to a DSLR camera under natural, as well as controlled lighting conditions.

47

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Figure 3.1: Virtual full denture previews in augmented reality. Each pair of images consists of the original image (top) and the virtual dentures preview (bottom). Our proposed system works in real-time, mimicking a dental virtual mirror.

3.1 Introduction

Dental health care is increasingly using digital technologies in order to provide cost-efficient procedures for the design of personalized treatments with high quality and predictable results. A common pipeline, which has been established for computer-aided dental health care, involves three stages: scanning, computer-aided design (CAD), and computer-aided manufacturing (CAM). During scanning, the jaw of a patient is captured by taking 3D surface scans, either directly by inter-oral scanning, or indirectly from a plaster cast. After capturing the dentition of a patient in a 3D model, dental solutions, comprising bridges, crowns, dentures, healing abutments, or braces, are designed with CAD tools. These tools allow dentists to accurately adjust existing, replaced or treated teeth, as well as other inter-oral parts such as the alveolar bone of the patient. Finally, the CAD result is produced by CAM that supports using various materials for dentistry,

such as composite resin, ceramic, or metal alloys. While subtractive manufacturing through milling from a blank is the state-of-the-art approach, the development of additive manufacturing with 3D printers for digital dentistry is advancing.

Our work is concerned with the CAD of teeth replacement treatments, such as implants, partial, or full dentures. CAD tools offer a wide range of tools to assist a dentist or dental technician in designing replacement solutions. They replace traditional procedures of dental technicals that have previously been carried out with physical modeling materials such as acryl or plaster. Moreover, there are also digital versions of apparatuses for functional analysis, like the *Virtual Articulator* [54], which simulates traces of teeth movement. CAD tools, like the *Virtual Articulator*, are used to predict where teeth are potentially colliding, since they would wear off faster in such regions. Overall, there is an established set of tools assisting the CAD of functional dentures, implants, and other treatments. CAD software [1, 8] show to the user by means of visualizations how well the designed objects fit to the jaw of the patient. However, these visualizations do not provide visual feedback on how the dental solutions fit from an aesthetic point of view. It is difficult to formalize teeth aesthetics because it concerns characteristics like beauty and taste, which are rather subjective. Thus, the patient opinion is valuable for the aesthetic analysis.

We define the high-level task of our work as the mapping of a virtual teeth model to an image of the patient's face, thus providing a highly realistic simulation not distinguishable from reality. Through these simulations, dental technicians can communicate with the patients to find a solution according to their taste and preferences. We propose *WithTeeth*, a tool that embeds virtual teeth into a video stream of the patient's face captured by a camera in real-time. A virtual teeth preview is shown in real-time and supports various face positions, orientations, and facial expressions. The main contributions of our work are:

- A real-time system that permits free head movements and allows the user to switch interactively between several different teeth replacement options.

- A mixed-reality solution that achieves compelling visualizations without depending on complex reconstruction algorithms, expensive hardware, or tedious calibration procedures.

3.2 Workflow and Task Analysis

Many factors contribute to the beauty of a smile, for example, tooth and gingiva colors, teeth shape, texture, transparency, and position. Simulating the physical properties of all teeth materials is a challenging task and involves technical as well as artistic components. In this work, we define the aesthetic model as a function depending on two factors: tooth color and teeth shape. In a Euclidean space, any rigid object can be defined by a position vector and an orientation matrix. In order to find a mapping for the virtual teeth into

image space, we determine the position and orientation of the upper and lower dentures. Since the human teeth are rigidly attached to the maxilla and mandible, we specify our low-level tasks as follows:

(T1) Finding positions and orientations of the maxilla and the mandible to augment the input image with virtual teeth.

(T2) Provide a rendering of virtual teeth, which allows patients to choose the teeth base color.

WithTeeth detects and tracks the face of the patient, automatically extracting facial landmark points. These points provide an estimate of the 3D pose of the patient's upper (maxilla) and lower (mandible) jaws, as well as the lips' contour lines. The data are then processed to seamlessly place the properly oriented virtual teeth at the correct position on the face of the patient (T1). The visibility of the virtual dentures in the mouth is limited to the area inside the lips' inner contour when blending virtual and real images during rendering. *WithTeeth* allows the user to switch dentures of different color and shape on the fly to find the one best fitting to the face of the patient from an aesthetic point of view (T2).

Our virtual mirror can be implemented on consumer hardware consisting of a webcam and a personal computer with a screen (e.g., tablet computer). The implementation of the virtual dentures preview works in real-time. It is robust with respect to face orientations, shapes, and expressions. The patient can freely move inside the capturing area and switch between different alternative options of dental replacements.

WithTeeth enables dental technicians to communicate with patients through augmented reality (AR). Using the tool, domain experts show to the patients a simulated result of the dental treatment at early stages. For the patients, the tool allows an exploration of virtual teeth and to make a decision on the preferable teeth color and shape. Closing a gap in the CAD software currently available on the dental market, *WithTeeth* introduces a new workflow of the dental aesthetic analysis. The tool brings benefits to both dental technicians and patients. By presenting a denture design to the patients at an early stage, dental technicians can avoid cases where patients are not satisfied with the results. Additionally, patients using the tool can increase their confidence in the treatment procedure by actively participating in the design process. The workflow opens new possibilities for remote medicine, minimizing travelling expenses. The dental technicians can send a new denture design to patients while they are at home to get their feedback, saving energy and time.

3.3   Related Work

Precise teeth detection and denture fitting has been rarely investigated, and the literature related to dental procedures is limited. Our work is related to medical applications that involve capturing facial landmarks and AR. We hereby review these two topics, providing examples of dental applications, if available.

Facial capturing. Capturing 3D facial models is an important research topic in many applications, e.g., game or movie industries, as well as the medical domain. Nowadays, it is possible to create photorealistic models of the human head using special facial scanners or photogrammetry. Facial paintings or markers can be used for capturing high-resolution static or animated facial models [129, 149, 285]. These methods are often cumbersome to use and time-consuming as they require manual marker placement on the face. In our approach, it is important to have a very fast capturing algorithm, since we provide immediate visual dental feedback to the patients. Other methods use structured light and project patterns directly on the face. They do not require prior face markings [275, 278, 295]. However, the skin texture is modified, which entails a number of limitations for photorealistic rendering applications. Furthermore, the methods reduce patient comfort as the light might be directly projected onto the face. During the last decade, depth cameras, such as Kinect or Intel RealSense, have become popular for facial capture [85, 200]. These depth cameras can provide an accurate head tracking and capturing. Most depth sensing solutions bundle infrared projectors and sensors with RGB video cameras. Such a strong coupling significantly limits flexibility. Built-in cameras often have limited capabilities and upgrading is not possible without replacing the entire assembly. Sophisticated parametric models and approaches have been designed, with a particular focus on the human body and the head. These 3D facial parametric models are often used to reconstruct the head from a photography or a video [62, 63, 218]. Richardson et al. [233] presented a method to reconstruct the facial geometry from a singular image. The method uses a convolutional neural network trained on a synthetic dataset consisting of photo-realistic facial images with known geometric forms. Thies et al. [256] presented an approach to reenact a target video based on the facial expression of an actor, who is recorded live with a webcam. The methods mentioned above require a training set of photos and are sensitive to changes in light conditions. Nevertheless, the techniques are promising for real-time face capturing and tracking. Despite this, the topic of precise teeth detection and denture fitting has not been sufficiently investigated. Wu et al. [288] presented a teeth reconstruction technique that is based on a set of photographs or a video for detecting the shape and position of the teeth. Their approach requires a database of high-quality teeth for training and is sensitive to light conditions. In addition, their approach does not work for patients with only a few or no teeth, who usually are the target group for dental prostheses. Detecting the lips is another important aspect for providing a plausible virtual denture. Eveno et al. [101] have presented an active contour algorithm for segmenting lips in 2D images. Their approach detects the boundary of the mouth and then fits a spline to the contour of the lips using a parametric model.
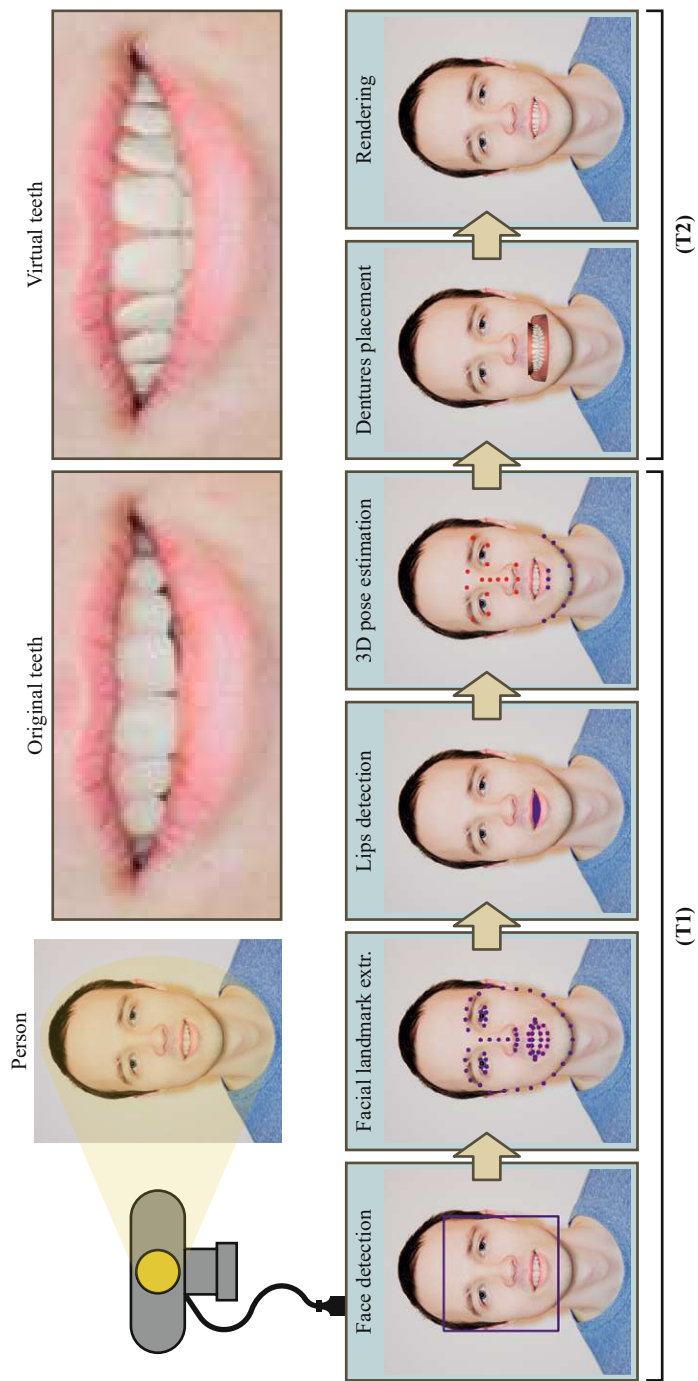
Figure 3.2: *WithTeeth* pipeline, showing all the employed steps.

Garrido et al. [118] presented an approach for 3D reconstructing the shape of the lips, based on monocular RGB videos. A comprehensive summarization and classification of the state of the art has recently been published by Zollhöfer et al. [297].

Augmented reality. Many AR applications use virtual mirror approaches, e.g., in commercial products [223] and anatomical education applications [198]. Chu et al. [72] presented an approach where users can try different types of jewelry. Rahman et al. [223] proposed a virtual mirror approach to preview how different types of cosmetics fit the user's face.. In the field of anatomical education, Blum et al. [55] use a depth camera to align a volumetric dataset to the human body. Stefan et al. [253] presented a system for bone anatomy education. Users are presented with a puzzle consisting of bones to assemble, employing their own body as a reference frame. Moreover, Mercier-Ganady et al. [199] visualize brain activity in real-time. Their approach teaches users to control different mental states, such as concentration or relaxation. In the dental domain, Rhienmora et al. [232] discussed a dental training simulator in AR. This simulator includes a virtual mirror for indirect vision and utilizes haptic force feedback. For dental reconstruction, a commercial software is available from Kapanu AG [10]. The software enables previewing dental reconstructions in AR, however, there is no publication explaining the technique. Other medical AR applications include image-guided surgery, e.g., the work of Salah et al. [242] for brain tumor resection. An example of focus-and-context visualization in medical AR is the work of Kutter et al. [178], which inspects bone structures to support trauma surgery. Virtual mirrors have been employed in numerous previous works [50–52, 207]. Illustrative visualization has been employed in the past in combination with AR, in order to show vascular structures around tumors [135].

3.4 Methodology

In this section we describe our method in detail. It consists of the following steps: face detection, facial landmark extraction, lips' inner contour detection, 3D pose estimation of maxilla and mandible, dentures placement, and finally rendering. An overview of our approach is depicted in Figure 3.2. In the face detection step, the face region is identified in the image. Then, facial landmarks are extracted in the face region. Using the facial landmarks the inner contour of the lips is detected as the virtual dentures have to be cropped there. Then, the 3D poses of the maxilla and mandible are estimated from the facial landmarks, in order to determine the positions of the lower and upper dentures. In the last step, the original teeth are replaced with the virtual dentures, resulting in a realistic dental preview in real-time.

3.4.1 Input

Our approach requires certain equipment and environmental settings. The goal of the work is to build a robust system for previewing dentures that require only consumer hardware, such as a computer and a webcam. We are targeting low-cost webcams, however, our system can work with more expensive and better quality cameras as

well. For the facial capture, the recording environment is very important to obtain a good performance. We recommend using environments with bright and indirect light. Bright light allows the recording device to use a low ISO sensitivity, which results in a better image quality. In cases where indirect light is not available, the light source should illuminate the face from the camera direction. The face of the patient should be homogeneously illuminated without any hard shadows. A good contrast between the face of the user and the background is recommended, while the face should not be occluded by any other objects.

The approach works for videos as well as for single images. For videos, an additional step is required to filter the noise in the input video stream produced by the camera sensor. To reduce noise, we apply a Kalman filter [162] that predicts an actual value based on a set of observations. For simplicity, we subsequently explain the steps of our pipeline for a single image, except at places where we are talking about the Kalman filter.

3.4.2   Face Detection

Face detection is the first step of the *WithTeeth* pipeline. In the input image, all human faces, which are present in the field of view of the camera, are detected using a histogram of oriented gradients [83, 106]. The face detector module searches for faces that are oriented towards the camera. Empirically, the angle between the face and the camera direction should be less than 45°. Otherwise, the face will not be captured appropriately, as shown in Figure 3.9. Due to performance reasons, the input frame is downsampled by a factor of five for the face detection. The frame itself remains untouched for the next steps. The downsampling factor depends on the resolution of the camera and how far the user is located from the device. We empirically found that for a webcam with a resolution of 1920×1080 pixels and a horizontal field of view of 70.5°, a downsampling factor of five is sufficient to detect a face at a distance of up to one meter from the capturing device.

3.4.3   Facial Landmark Extraction

In the second step, landmarks are extracted from the facial region. We use a 68-point facial model for the landmark detection (see Figure 3.3), which has been trained on the ibug 300-W dataset [239]. Due to noise in the input image and limitations of the facial landmark detection technique, the points include some error. To reduce noise in the points between the video frames, a Kalman filter is applied [162]. It reduces the noise in the measurements of the landmarks and predicts their actual values. For each landmark point in a frame, the Kalman filter is applied separately. The filter has two main parameters: the process-noise covariance and the measurement-noise covariance. These parameters are adjusted for each specific camera and the recording environment. For the webcam that is used for all our results, we set the process-noise covariance to two and the measurement-noise covariance to four. After the facial landmarks have been extracted, we detect the lips to determine the mouth region.
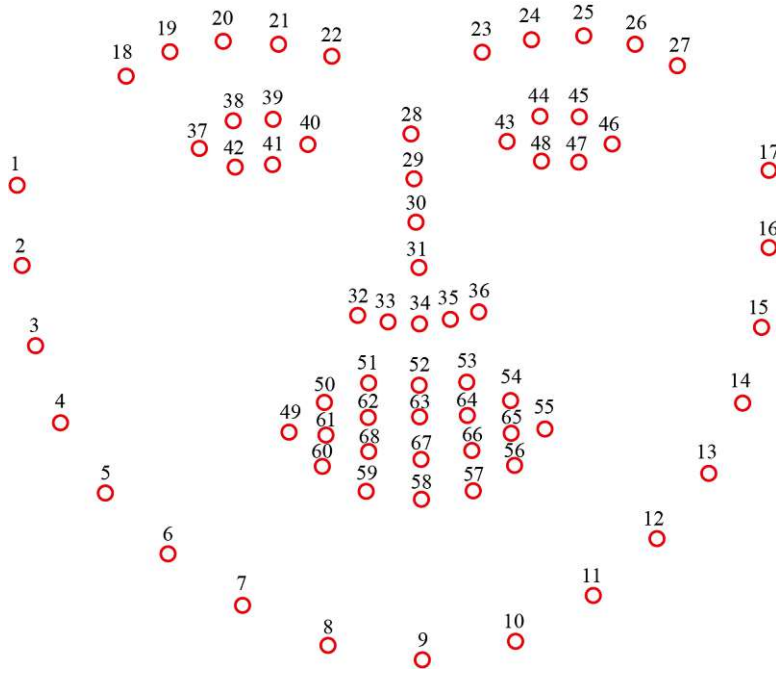
Figure 3.3: The facial landmark extraction provides 68 facial points. The outer contour of the lips has more points than the inner contour. We use a heuristic to predict points on the inner lip contour using information from the outer lip contour.

3.4.4 Lips Detection

The facial landmark extraction provides 68 points distributed over the face including the region of the lips (see Figure 3.3). We found that the number of points on the inner lip contour is insufficient to smoothly replace the original teeth with the virtually rendered dentures. Especially points at the corners of the lips are missing, where the curvature is often the highest. To address this, we developed a heuristic to increase the number of points on the inner contour of the lips. We also observed that the inner contour has a similar shape as the outer contour of the lips. Using this characteristic, we add four additional points that are located between points p_{61} and p_{62}, p_{64} and p_{65}, p_{65} and p_{66}, p_{68} and p_{61} (see Figure 3.4). The subsequent steps determine a point between points p_{61} and p_{62}:

$$\mathbf{a} = p_{51} - p_{49}, \qquad \mathbf{b} = p_{50} - p_{49}, \tag{3.1}$$

$$\mathbf{c} = p_{62} - p_{61}, \qquad \mathbf{n} = \begin{pmatrix} -\mathbf{c}.y & \mathbf{c}.x \end{pmatrix}^{\mathsf{T}}, \tag{3.2}$$

$$P = p_{49} + \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{|\mathbf{a}|} \cdot \hat{\mathbf{a}}, \tag{3.3}$$

$$C = p_{61} + 0.5 \cdot \mathbf{c}, \tag{3.4}$$

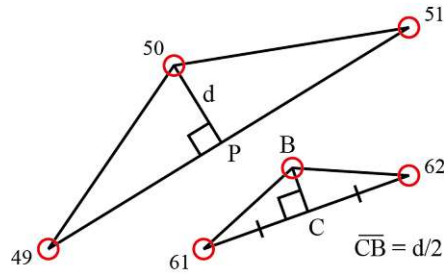$$B = C + 0.5 \cdot \hat{\mathbf{n}} \cdot \|50 - P\|, \tag{3.5}$$

55

Figure 3.4: Example of our heuristic to increase the number of points on the lips' inner contour. A point between points p_{61} and p_{62} is added, based on the positions of points p_{49}, p_{50}, and p_{51}.

with $\hat{\mathbf{v}}$ being the unit vector of $\mathbf{v}$. We proceed analogously for the three other pairs of points.

After the reference points of the lips' inner contour are detected, we calculate the mask of all the pixels inside the polygon formed by the points. Then, we apply a morphological erosion and a normalized box filter, with kernel size of 5×5 pixels, to the mask. The erosion prevents the virtual dentures from leaking into the area of the lips. The Gaussian blur softens the transition from the real to the virtual region, thus making it less distractive.

3.4.5 Pose Estimation

The pose estimation of a 3D object in a 2D image is done by finding the 3D-2D point correspondence. To create 3D models of the maxilla and mandible, we use the head mesh from the BodyParts3D dataset [204] (see Figure 3.5a). The maxilla model consists of 15 points that are located on the nose, eye, and eyebrows. The mandible model consists of 12 points that are placed on the chin and the lower lips. To estimate the pose of the maxilla, we use an iterative method based on Levenberg-Marquardt optimization [182] by fitting the 3D model to the facial landmarks detected in the previous steps. The method minimizes the reprojection error between the points of the 3D model and the facial landmarks. It provides a transformation matrix and a translation vector for the points from the model coordinate system to the world coordinate system. To estimate the pose of the mandible we perform a similar procedure as with the maxilla. As only difference, we use the extrinsic guess and provide to the method the rotation matrix and translation vector concerning the maxilla as approximate initial position of the mandible. We assume that the mandible has a similar position and orientation as the maxilla, which applies to most of the cases.
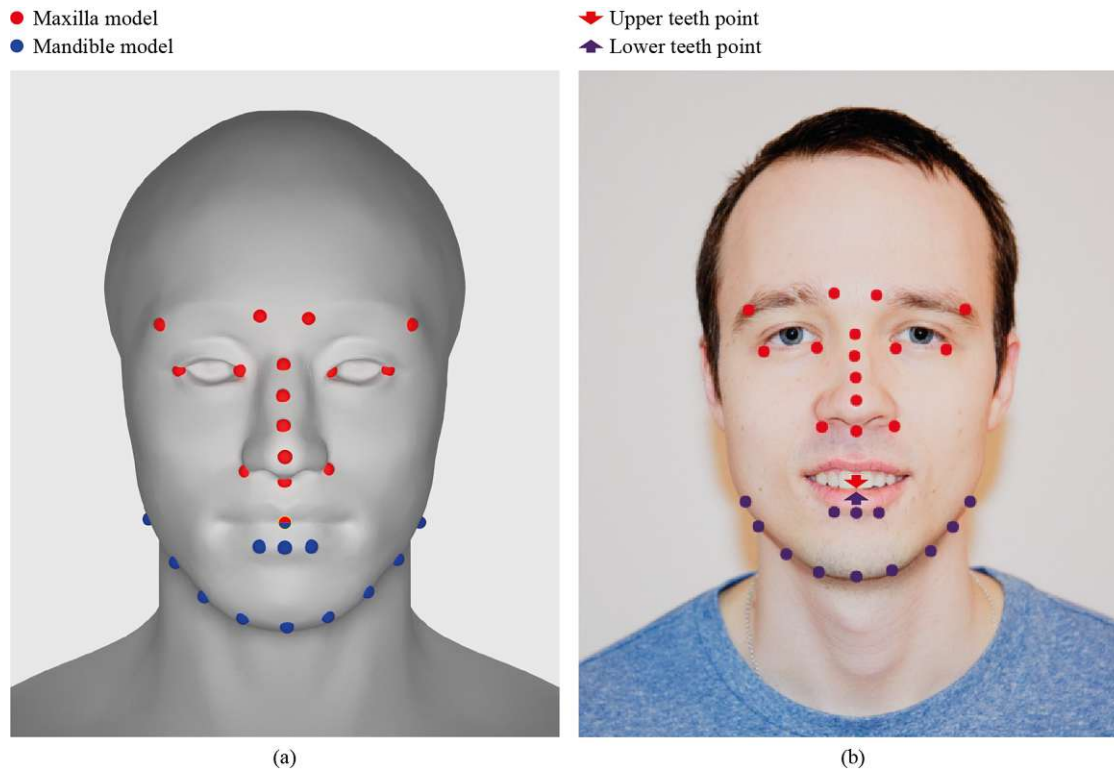
Figure 3.5: We have developed two 3D models of the human head. The red points indicate the model of the maxilla and the blue points indicate the model of the mandible. For the maxilla we find more stable points that are almost motionless during facial emotion or smile. For the mandible, points are placed on more dynamic regions. Therefore, the quality of the pose estimation for the mandible is worse than for the maxilla.

3.4.6 Dentures Placement

We determine the occlusal plane to properly fit the virtual teeth to the original ones (see Figure 3.6). The functional occlusal plane goes through the molar and premolars and indicates the relative positions of the teeth with respect to the head. The functional occlusal plane is always parallel to the interpupillary and Ala-Tragus lines. The interpupillary line connects the centers of the eyes' pupils (see Figure 3.6a). The Ala-Tragus line connects the wing of the nose with the small pointed eminence of the outer ear (see Figure 3.6b). Capturing the interpupillary line is not necessary for our method. We assume that the previous 3D pose estimation of the maxilla provides us with a transformation matrix that is already aligned to the interpupillary line. To find the Ala-Tragus line, the user has to place four points on an image of the patient's head, which is shown in a profile view. These points are at the following locations: chin, forehead, wing of the nose, and pointed eminence of the ear. The line defined by the first two points is referred to as *facial plane*. It depicts the orientation of the head in space. The
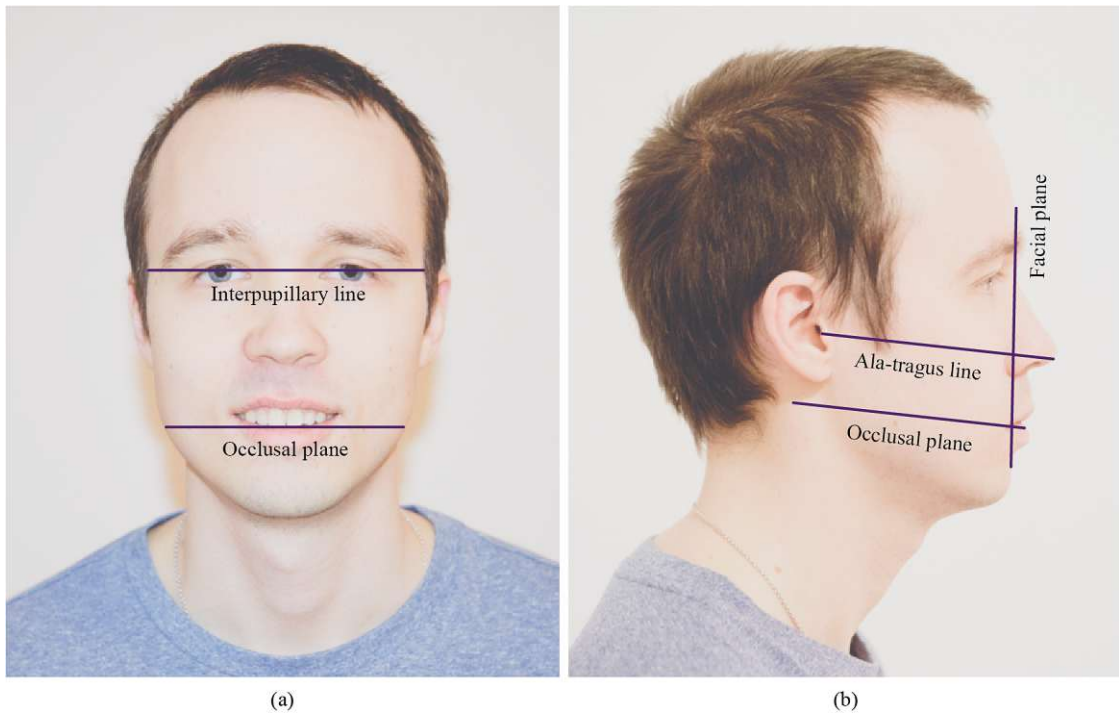
Figure 3.6: The occlusal plane is parallel to the interpupillary line (a), and to the Ala-Tragus line (b).

line defined by the third and fourth points indicates the Ala-Tragus line. Afterwards, the angle between the occlusal plane and the facial plane is measured and used to place the dentures, as subsequently described.

After determining the transformation matrices and rotation vectors for the maxilla and mandible, we verify the mandible position. Typically, the mandible transformation is noisier and has a larger error than the one of the maxilla. This is because the mandible is the moving part and the maxilla is attached to the skull. The point where to find the virtual teeth is defined within the facial 3D models (see Figure 3.5). In some cases, this point has to be adjusted to match the anatomy of the person. We provide a method to shift the point from its original position in order to achieve a good fitting. In addition, we check that the mandible is at about the same distance to the camera as the maxilla. We apply the initial rotation to each denture in order to match the occlusal plane. Subsequently, we apply transformations of dentures to the world coordinate system. In the next step of the *WithTeeth* pipeline, we realistically render the dentures to provide a natural and aesthetic impression.

3.4.7 Rendering

The composition of the denture rendering is shown in Figure 3.7. It consists of textured foreground and background planes, a camera, and the virtual teeth placed in between these two planes. The camera looks through the foreground plane towards the background plane. The foreground plane presents the texture of the patient's face with the mouth region transparently rendered in order to see through the mouth. The teeth are placed so that the camera can see them in the mouth region. The virtual teeth are placed in the same way on the screen as the original teeth. The background plane presents the texture of the patient's face as well as the foreground plane, but it does not remove the mouth region. As a consequence, the tongue remains visible in the result image. The light source is located close to the camera and casts light on the teeth. It is important to match the light color with the real light. In our tool, it is possible to modify the intensity of the light as well as the color temperature. In this way, we can adjust the virtual teeth to match the real ones in appearance.

3.5 Implementation

The face detection is implemented using the *Dlib* library [167]. *OpenCV* [58] is used for pose estimation and the *Unreal Engine* is taken for rendering the virtual dentures [15]. *Dlib* and *OpenCV* offer state-of-the-art algorithms for face detection and facial landmark detection. The *Unreal Engine* enables fast prototyping for rendering and works as the binding framework for all the involved libraries. Moreover, *Dlib*, *OpenCV*, and *Unreal Engine* allow us to compile the application on many platforms such as Windows, macOS, Linux, iOS, and Android. Being able to execute the application on multiple platforms is a crucial requirement. It allows the dental technicians to run the application on their workstations and the patients to preview the simulation results on portable and affordable devices like mobile phones or tablet computers. There are other libraries for face detection and facial landmark extraction such as Apple Vision [3], ARCore [4] or ARToolKit [5]. However, some of them are platform-dependent, for example, Apple Vision supports only macOS and iOS; and ARCore supports only Android.

We have created two different materials in *Unreal Engine* for the teeth and the gingiva. Both materials use the subsurface shading model [12] implemented in Unreal Engine. The material of the gingiva has the following parameters: Base color (R, G, B) = (0.694, 0.251, 0.243); Subsurface color (R, G, B) = (0.694, 0.251, 0.243); Roughness = 0.23; Specular = 1; Subsurface amount = 0.2. The material of the teeth has the following parameters: Base color (R, G, B) = (0.847, 0.426, 0.066); Subsurface color (R, G, B) = (1, 1, 1); Roughness = 0.13; Specular = 1; Subsurface amount = 0.19. An exemplary rendering of virtual dentures, using our proposed approach, is presented in Figure 3.8. To provide a realistic impression of the teeth during movement, we use motion blur. Ambient occlusion is also used to give the teeth a more volumetric and natural appearance.
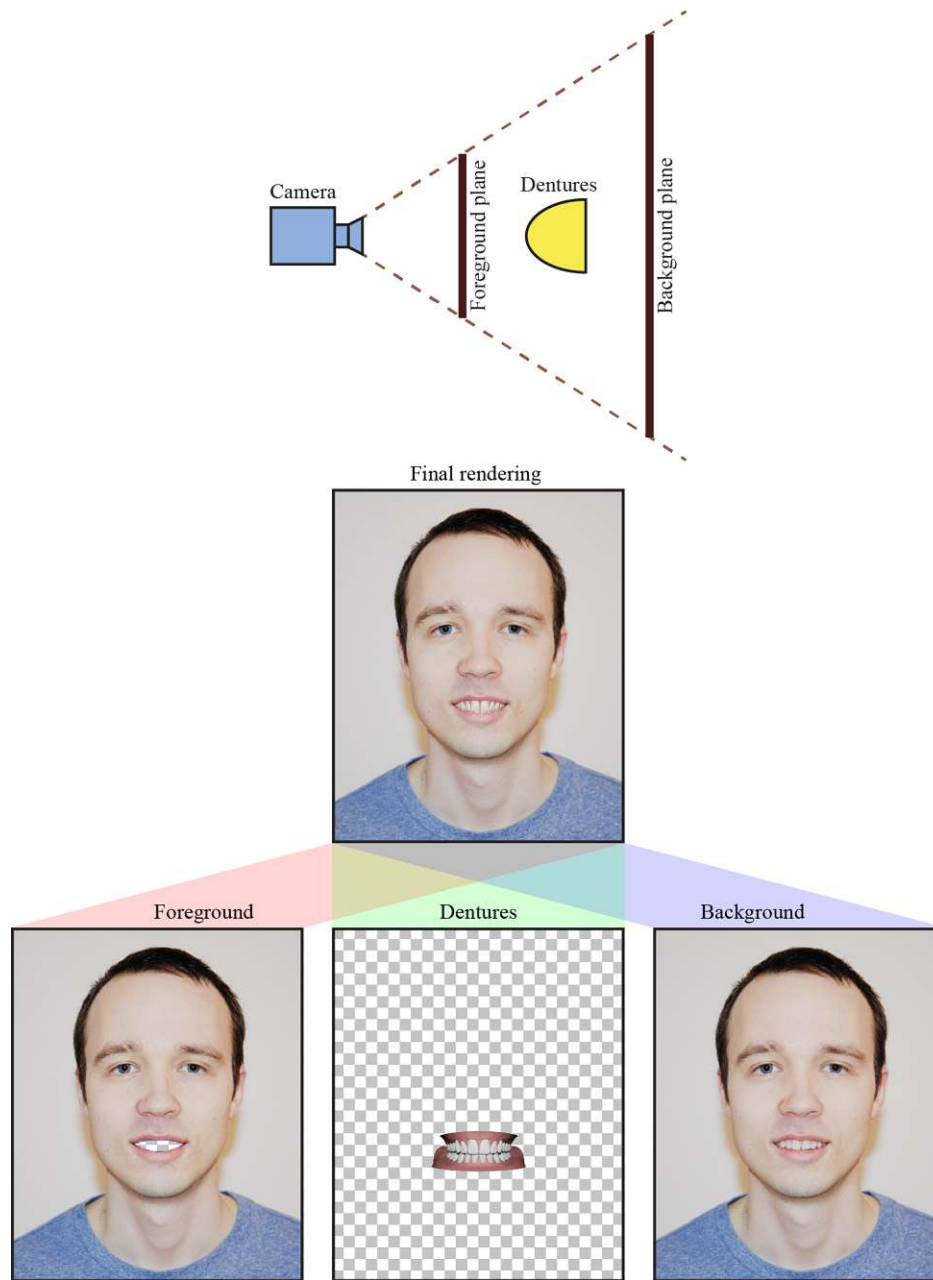
Figure 3.7: Rendering the 3D scene involves the camera, the foreground plane, virtual dentures and the background plane. The intrinsic camera parameters should match the parameters of the real-word capturing camera (field of view, aspect ratio). The virtual dentures are positioned, at the screen position of the real teeth. The distances of the foreground and background planes to the camera are defined so that the virtual teeth are always in between these two planes.

Figure 3.8: Example of dentures rendering.

3.6 Results

In this section, we summarize the most significant results of our approach, illustrating application scenarios and demonstrating its functionality, and performance.

Figure 3.1 shows three participants with their original teeth (top) and with virtual dentures (bottom). Pose estimation works for a large range of head positions (arbitrary yaw, pitch, and roll). Our approach works for participants with glasses (see second and third columns) and with different facial shapes and sizes.

Figure 3.9 illustrates the performance of our approach on different facial orientations (rows). Each step of the *WithTeeth* pipeline is displayed in a separate image (columns). In the **first** row the face is oriented towards the camera. The face detection method finds the facial region in the correct place and the facial landmarks are precisely extracted. The mouth region is detected with sufficient quality. The borders of the lips are not exact. However, no real teeth are visible in the final result. Markers of the pose estimation approach are placed at the correct positions and close to facial landmarks. The rendered dentures look natural like real teeth. The **second** row demonstrates the results if the face is slightly turned away from the camera. Each step is performed without visible deviations in quality from the first row.

In the **third and fourth** rows no quality degradation is visible in the images as well. In the **fifth** row, the accuracy of the inner contour detection of the lips is starting to degrade. Since some key points of the 3D maxilla and mandible models are occluded, the pose estimation step is not performing so well, resulting in a significant decrease in quality of the denture placement step. This is visible in the result image where the dentures are displaced from the original teeth position. The results are becoming even worse in the **sixth** row, where the head is turned further away from the camera. The lips'

inner region is becoming less accurate. It includes part of the right cheek and more teeth are becoming visible from the left side. The lips detection is failing due to the imprecise facial landmark extraction. In the image with the facial landmarks (see column 3, row 6 in Figure 3.9), the feature points around the right corner of the mouth are imprecise. Although the region leaks into the right cheek, this region looks fine in the final rendered image. The reason is that the inner region of the lips is used only to occlude the virtual dentures. Also, locations like the right corner of the eyebrow or the right corner of the right eye are not placed exactly since their correct positions are occluded. As a consequence, they are placed on the edge of the face. Imprecise landmark detection leads to an error in the pose estimation and, therefore, to an erroneous denture placement.

As the first four rows in Figure 3.9 present good results we conclude that our approach works well if the head of the user has not an extreme angle away from the camera. Mostly, this is due to the facial landmark detection step that yields worse results if the head is turned further away from the camera. We present our tool as a virtual mirror for previewing dentures. Since it is not possible to see oneself in the mirror from extreme viewing directions, our virtual mirror provides high-quality previewing.

3.7 Limitations and Future Work

The major contribution of this work is the development of a pipeline for pre-treatment dental previewing that can produce realistic results in real-time. The pipeline consists of multiple steps, and each of them can be individually improved. In this section, we discuss limitations of the pipeline steps and outline possible improvements for future work.

Facial landmark detection. The current module of facial landmark detection was trained on frontal faces and, therefore, does not work well if the head is turned far away from the camera. A solution to this limitation might be training another module for finding facial landmarks for such head positions or extending the current training database with more annotated images of the head in such positions.

Also, facial landmarks are prone to noise in the input image and they are detected at slightly different positions for each frame in the video. This noise is propagated to subsequent steps and causes jittering of the augmented dentures. To reduce the noise in the input signal we use a Kalman filter that predicts the correct positions of landmarks based on previous observations. This filter provides good results if the face undergoes just small movements in the frame or the motion is predictable, e.g., the face is moving on a line. However, if the face movements are too fast and random, the error of fitting the dentures is increased. For example, if a fast motion just starts, the dentures will be moving slightly slower than the head and they will be close to their original position. The same happens when a fast movement stops. The Kalman filter continues predicting the movement and dentures will sway a bit further along the movement trajectory. This limitation does not allow patients to fully immerse into the mixed reality scene. In most of the cases, though, patients do not move fast and often explore the virtual teeth using rather static positions.

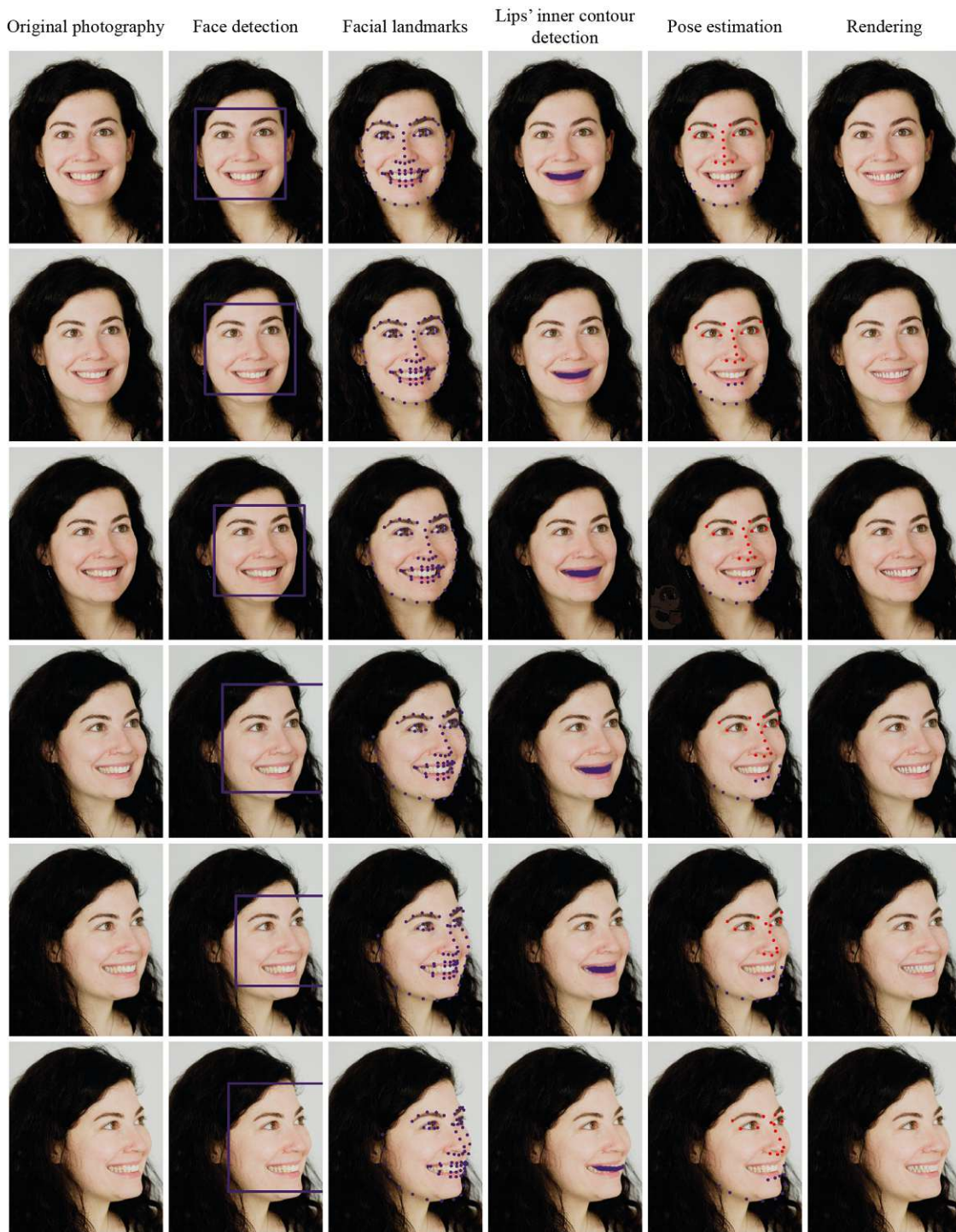| Original photography | Face detection | Facial landmarks | Lips' inner contour detection | Pose estimation | Rendering |

Figure 3.9: *WithTeeth* pipeline for different face poses of the same person.

Lip detection. The current lip detection method is based on the facial landmarks. In some extreme facial expressions, the landmarks are not detected correctly, e.g., if the mouth is wide open. The facial landmark detector is based on a machine learning approach and strongly depends on the training set. To avoid this drawback, more sophisticated algorithms can be used such as, for example, boosted edge learning.

Dentures placements. The placement of the dentures depends on the pose estimation of the upper skull and the chin. We developed 3D models to find the positions using a publicly available dataset of the human body. The pose estimation performance can be improved, e.g., by personalizing the models. For example, the 3D face of the user can be captured using a photogrammetry approach or a depth camera. Then, the facial landmark points will be searched and personalized on the head of the user mesh. This will potentially improve the dentures placement step. It might require additional equipment or steps in the pipeline. In the current approach, we use a Levenberg-Marquardt optimization that fits the facial mask consisting of a set of 3D points to 2D facial landmarks. The pose estimation can be further improved using a 3D-to-3D optimization, requiring depth information of the facial landmarks that can be obtained using, for example, a depth camera.

In some cases, only a single tooth of the patient has to be treated or replaced. The current approach does not support to preview the dental treatment outcome for a single tooth. The individual tooth can be occluded by other teeth, thus the tooth visibility or the visibility mask has to be determined for a correct fitting. This can be calculated by using an individual teeth extraction based on the head position and a priori knowledge of a typical tooth layout. Additionally, the teeth visibility mask will help to hide a patient's teeth if they are shorter than the dentures.

During testing, we found that in some cases it is useful to zoom in to the teeth to see small details. In the current implementation this is not possible. If the camera is placed close to the mouth of the patient, other important parts of the face, such as eyes or chin, are outside of the camera view. Thus, some facial landmarks cannot be used for detecting the face orientation. This limitation can be overcome, by adding more landmarks to the mouth region.

Rendering. The current rendering approach can be improved by measuring physical parameters, e.g., the bidirectional reflectance distribution function, of real teeth and adjusting rendering settings accordingly. To make the lighting conditions realistic, environmental maps can be used. We consider a precise color previewing of dentures as future work, because it requires all the rendering improvements mentioned above. In addtion, precise color previewing requires a calibrated camera or a recording environment with controlled light. Also, shadows from lips can be implemented and ambient occlusion can be taken into account more accurately.

An informal interview with domain experts revealed the need of dental technicians for simulating skin deformations caused by wearing dentures of different sizes. The dental state, shape, and size of teeth have a big influence on the shape of the upper and lower

lips. For example, if teeth are missing, the skin will refract towards the mouth of the patient. Or vice versa, if the teeth are located too far in front of the mouth the lips will be bigger and more visible. The skin deformation has a big influence on the patient appearance and has to be taken into account in demonstrating the outcome of a dental treatment. We consider a realistic model of skin deformation as an interesting direction for future work.

3.8 Conclusions

We presented *WithTeeth*, a virtual mirror approach, targeting the dental industry. Using our method, dental technicians can demonstrate possible treatment outcomes in advance to the patients to get their feedback. Patients are involved in the design procedure that might increase their confidence and satisfaction in the outcome. Virtual dentures can be exchanged on the fly and various characteristics can be controlled such as tooth color and size. Our tool works in real-time and requires only basic equipment comprising a computer with display, and a consumer-level webcam or a DSLR. Also, our approach does not require a priori training. The lighting parameters must be adjusted manually to match the real lighting of the environment. We described our method and provided implementation details where necessary. We discussed limitations and suggested directions for further research. Our approach, following modern trends in the dental industry, shows how digital technologies may reform and improve the dental reconstruction.

Paper D: improvement

Paper A: defect prevention

Paper C: performance evaluation

Paper B: control of defective outputs
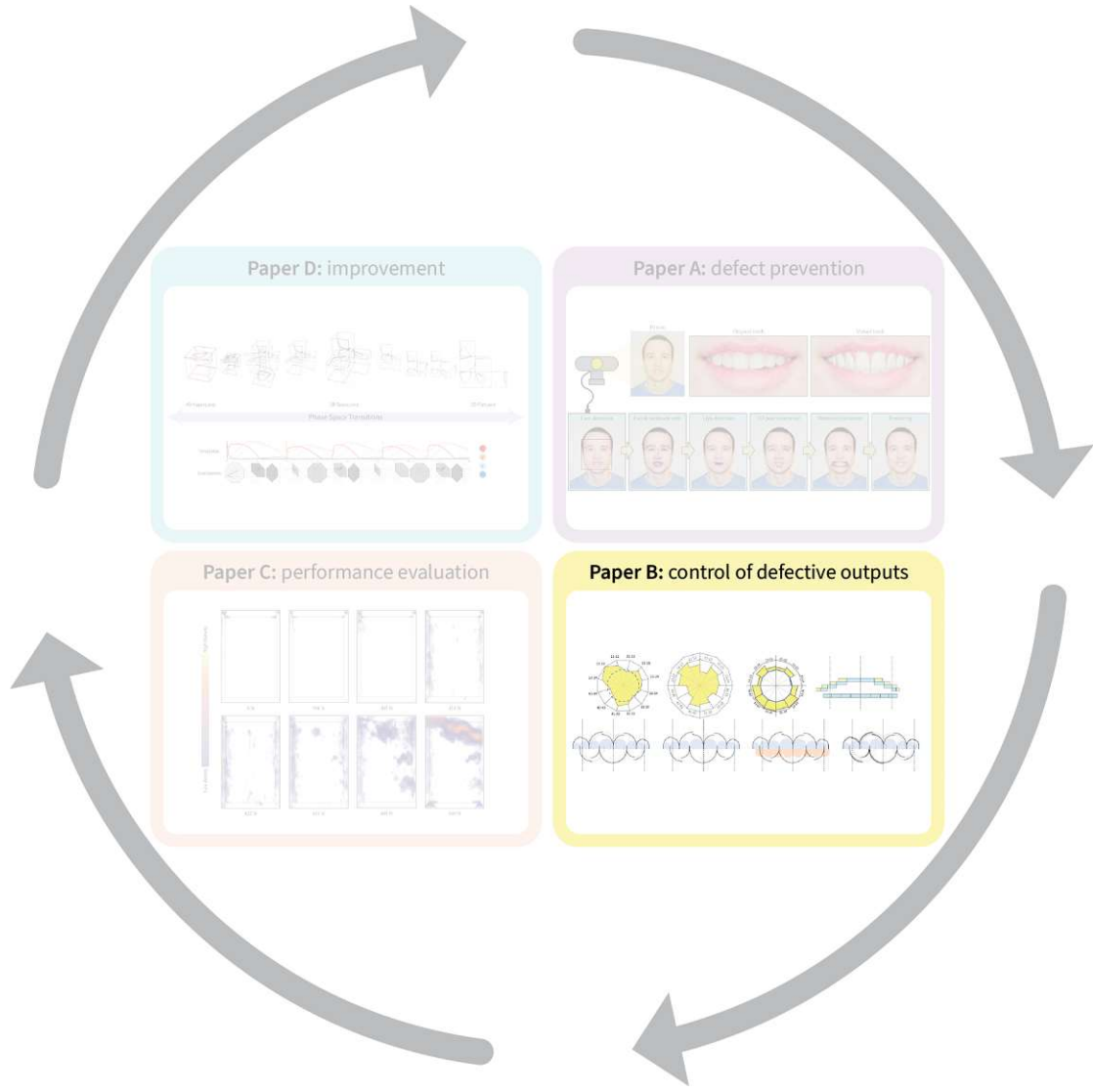
CHAPTER 4

Control of Defective Outputs

This chapter addresses the control of defective outputs (i.e., the *do* phase of the PDCA cycle) and is based on the publication:

Dental healthcare increasingly employs computer-aided design software, to provide patients with high-quality dental prosthetic devices. In modern dental reconstruction, dental technicians address the unique anatomy of each patient individually, by capturing the dental impression and measuring the mandibular movements. Subsequently, dental technicians design a custom denture that fits the patient from a functional point of view. The current workflow does not include a systematic analysis of aesthetics, and dental technicians rely only on an aesthetically pleasing mock-up that they discuss with the patient, and on their experience. Therefore, the final denture aesthetics remain unknown until the dental technicians incorporate the denture into the patient. In this work, we present a solution that integrates aesthetics analysis into the functional workflow of dental technicians. Our solution uses a video recording of the patient, to preview the denture design at any stage of the denture design process. We present a teeth pose estimation technique that enables denture preview and a set of linked visualizations that support dental technicians in the aesthetic design of dentures. These visualizations assist dental technicians in choosing the most aesthetically fitting preset from a library of dentures, in identifying the suitable denture size, and in adjusting the denture position. We demonstrate the utility of our system with four use cases, explored by a dental

technician. Also, we performed a quantitative evaluation for teeth pose estimation, and an informal usability evaluation, with positive outcomes concerning the integration of the aesthetics analysis into the functional workflow.

4.1 Introduction

Oral diseases affect nearly 3.5 billion people worldwide and pose a major health burden for many countries [17]. Only 41% of Europeans have all their natural teeth, and a third of the rest wears partial or full, removable dentures [18]. In this chapter, we focus on full denture reconstructions. Wearing dentures often brings various difficulties to their owners, such as day-to-day pain or embarrassment. This leads to significant discomfort or other severe consequences, such as long-term social distancing. Furthermore, dental healthcare is expensive and there is a strong correlation between access to dental care and satisfaction with dentures [17, 18]. Therefore, cost-efficient and accessible solutions are required.

Dental healthcare increasingly employs digital technologies to design and manufacture high-quality and cost-efficient dental prosthetic devices. The modern dental reconstruction workflow (see Figure 4.1) requires two appointments with the patient [11, 254]. During the *first appointment*, a dental technician captures the dental impression of the patient (see Figure 4.1 a). The impression is a negative imprint of hard and soft mouth tissues, i.e., the residual teeth, and gingiva. Impressions can be obtained traditionally, using a dental tray and a special thermoplastic impression material, or digitally, using an intraoral scanner [193]. An accurate impression is the foundation for a patient-specific denture. Once the dental impression is acquired, dental technicians use dedicated CAD software to extract several important features from it. First, they determine the *occlusal plane* that covers the incisal edges of the anterior teeth, and the *occlusal surfaces* of the posterior teeth. Second, they remove the jaw undercuts from the impression mesh to ensure a comfortable denture insertion and removal. Next, the denture border lines are drawn on the impression to outline the extents of the denture (see Figure 4.1 b).

One way to design a full denture starts with the selection of a suitable denture model from a preset, referred to as the *teethmaster*. This is based on the previously extracted features and the aesthetic preferences of the patient. The selection is done from a representative range of teethmasters, based on criteria, such as configuration (one or two molars), bow size (shape of the alveolar line), and the teethmaster functionality concerning statics. Further refinements are made with specific CAD tools that adjust the shape or position of the teeth, to satisfy functional and aesthetic criteria. Then, a prosthesis is generated by modeling the gingiva, which is the interface between the teeth and the denture base. The denture base is derived from a patch, cut from the impression scan with the denture border line, and, thus, fits the jaw of the patient. Fine-grained refinements are made by editing the gingiva to convey a natural appearance (see Figure 4.1 c).
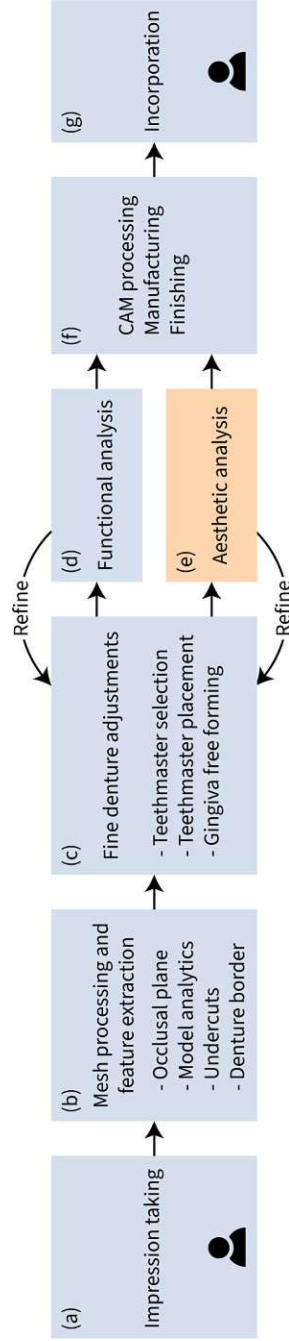
Figure 4.1: The modern workflow of dental technicians, as described in Section 4.1. For the first time, we integrate interactive aesthetics analysis into the workflow, enabling dental technicians to assess functional changes of the denture shape from an aesthetic point of view.

Additionally, the digital workflow allows specialists to conveniently analyze the functionality of designed dentures for the entire jaw of the patient (see Figure 4.1 d). With a *virtual articulator*, mandibular movements can be simulated on a digitalized physical model of the mandibular joints and the masticatory apparatus. During the *functional analysis*, issues related to the denture design, such as teeth collisions, are identified and corrected by changing the teeth shape, and the layout or the position of the teeth. Besides having well functioning and comfortably fitting dentures, patients are also interested in aesthetically pleasing denture designs, which greatly affect their overall appearance. Designing a denture that adheres to the aesthetic requests of the patient is a challenging task (see Figure 4.1 e), which is currently accommodated through the use of the aforementioned teethmasters. Finally, the prosthesis is manufactured utilizing CAM technology (see Figure 4.1 f). Depending on the used materials, dental technicians may apply various finishing procedures, such as sanding or polishing. Fine denture adjustments are performed during the *second appointment*, when the prosthesis is delivered to the patient (see Figure 4.1 g).

During the design of dentures, dental technicians consider various facial and dental features for a functional, as well as natural-looking and aesthetic, result. While the functional analysis is based on objective criteria, design decisions that concern aesthetics are rather subjective, and rely on the intuition and experience of the dental technician. We contribute to the denture design process with a variety of aesthetic criteria (e.g., proportions) and measurement strategies (e.g., smile lines analysis), implemented as computational tools, and integrated into a denture CAD software. To this end, aesthetic features have to be extracted from the predicted appearance of a patient with new dentures, quantified, and visually conveyed.

In this work, we base our approach upon *predicting* the appearance of a patient with in-line designed dentures, *quantifying* several aspects of dental aesthetics, *visualizing* these aspects interactively during denture design, and *comparing* them. Our approach is integrated into the state-of-the-art digital workflow of dental technicians.

4.2 Task Analysis

At an early stage of our work, we conducted an informal interview with our collaborating dental technicians. During the interview, we identified two *tasks* (*T1*, *T2*) that should be addressed to improve the current dental workflow concerning the aesthetics analysis of dentures. Every task corresponds to a specific step in the workflow and contains a list of *requirements* (*R 1–R 11*), as summarized in the following:

T 1 Denture Aesthetics Overview and Assessment: the dental technician selects the most suitable teethmaster for the patient from a large library.

R 1 Provide a preview of the dental treatment outcome.

R 2 Compare different dentures and assess each case.

R 3 Provide an overall score for each denture to quickly identify and compare suitable dentures.

R 4 Propose a visualization to identify and show the main problem of each denture.

R 5 Propose a visualization to show the golden proportions for the upper teeth.

T 2 Fine Adjustment of Selected Teethmaster: the dental technician adapts the selected teethmaster to the unique jaw anatomy of the patient, and improves it aesthetically.

R 6 Provide interactive feedback on all denture adjustments.

R 7 Propose a visualization that includes facial proportions.

R 8 Propose a visualization to show the six smile lines.

R 9 Propose a visualization that includes the three different scales with golden proportion, i.e., the eye scale, mouth scale, and teeth scale.

R 10 Provide specific feedback helping to identify, and localize what is the major problem with the denture, and how to correct it.

R 11 Integrate the entire solution into the current workflow of dental technicians.

4.3 Related Work

Our work concerns various topics from visual computing, visualization of the predicted outcome, visual analytics for treatment planning, and facial and dental aesthetics. In this section, we summarize the related work of these topics.

Teeth Pose Estimation: Aichert et al. [23] presented a method for image-based teeth tracking in a video. They used computed-tomography scans of the patient's jaw to estimate the position of the maxilla and mandible. Initially, the teeth must be placed manually. Wang et al. [273] discussed a similar approach to track teeth in a stereo video for augmented reality navigation in maxillofacial surgery. Later, they also introduced a method for a single-camera setup [274] for oral and maxillofacial surgery. Yang et al. [292] developed a jaw kinematics model from a stereo video. They used intra-oral scans to track the maxilla and mandible pose. The aforementioned approaches require that a significant part of the teeth is visible and they address different applications. Zoss et

al. [298] track the jaw movements, based solely on the visible skin surface. Their approach requires training data and may provide inaccurate results if the patient has unique jaw movements, which cannot be triangulated inside the convex hull of the training set.

Predicted Outcome: Zachow et al. [293] assessed different therapeutic strategies of craniomaxillofacial surgery. In their work, they used tomography images of patients and a finite-element method on various scenarios of treatment outcomes. Lee et al. [181] developed a 3D surgical simulation system for rhinoplasty planning. They use a 3D feature-based volume morphing technique to preview various treatment scenarios. Kim et al. [166] proposed a tool to simulate outcomes in breast plastic surgery. With an image-based method, they got a 3D model of the patient and simulated the appearance of the patient's breast using an example-based algorithm. Keeve et al. [165] employed a 3D finite element tissue model to predict soft tissue changes from the realignment of underlying bone structures. Chabanas et al. [68] predicted deformations of facial soft tissue from bone repositioning in maxillofacial surgery. The method is based on a generic 3D finite element model and it can provide simulations of face deformations under muscle actions. Gladilin et al. [120] realistically simulated facial expressions of emotions for craniofacial surgery planning. The aforementioned approaches deal with various applications and cannot be used to predict dental treatment outcomes. Amirkhanov et al. [28] (Chapter 3) presented a virtual mirror approach to preview full dentures in augmented reality. Their method estimates the teeth pose based on facial landmarks and replaces natural teeth in a video stream with a denture rendering. The method is limited to provide a preview and cannot be used by dental technicians to assess aesthetic changes caused by fine denture adjustments.

Visual Analytics for Surgery Planning: Krekel et al. [174] developed a technique for the visualization of a predicted motion range for shoulder replacement. Every change of the treatment planning interactively updates the predicted motion range of the patient, allowing surgeons to identify the most suitable treatment scenario. Dick et al. [91] presented a stress tensor field visualization of bones for hip joint replacement. The method is interactive and assists surgeons in determining the optimal implant design. Chen et al. [70] developed a system to assist knee study and operation. The system includes a 3D knee model reconstruction, motion simulation, biomechanical visualization, and virtual surgery. Meehan et al. [196] presented an interactive technique for planning craniofacial surgery. Their method computes non-linear soft-tissue deformations, due to bone realignment. Nadjmi et al. [206] proposed a semi-automatic method to virtually define the dental occlusion. They use a rigid-motion physical-engine to ensure the impenetrability of dental meshes. All approaches mentioned above are tailored to specific applications and cannot be used for the aesthetics analysis of dentures.

Facial and Dental Aesthetics: The facial attractiveness [216, 217, 257], as well as smile attractiveness [145, 159, 263], were intensively studied in psychology. Recent developments in the machine learning domain also include an automatic facial attractiveness assessment. For example, Eisenthal et al. [97] apply machine learning for computing a facial attractiveness score. Xu et al. [289] predicted facial beauty from a photo, using a

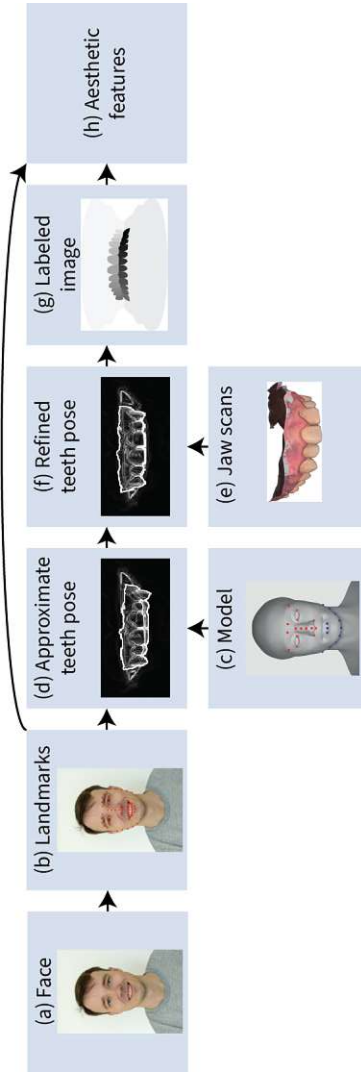Figure 4.2: Metric extraction starts with face detection and landmark extraction (a b). Later, we approximate the teeth pose by solving a 3D-to-2D point correspondence between the head model and facial landmarks (c d). We refine the teeth pose using the impression of the patient (e f). Denture previews are generated by replacing the natural teeth with denture renderings, and we extract aesthetic features from these images (g h).

convolutional neural network. Fan et al. [103] computed facial attractiveness, using a very deep residual network. Liu et al. [184] presented a tool for face beautification. Their tool manipulates the input face image to increase its aesthetic score. However, most of the work on facial attractiveness assessment and beautification does not consider dental aesthetics.

4.4 Method

We present a solution for the interactive analysis and exploration of dental aesthetics. To support the aesthetics analysis, we first estimate the teeth pose in a video frame (see Section 4.4.1). We then extract relevant metrics and visualize them to assist dental technicians in the aesthetics analysis of denture and decision making (see Section 4.4.2). According to *R 11*, the method has been integrated into existing CAD software for designing full dentures from the impressions of patients.

4.4.1 Teeth Pose Estimation

The teeth pose estimation procedure is crucial for the entire aesthetics analysis, as it enables denture preview and aesthetics metric calculation. We estimate the teeth pose in two steps. In the first step, we approximate the teeth pose using only facial landmarks (see Figure 4.2 a–d). In many cases, this is sufficient to preview the denture. We refer to it as the *approximate teeth pose*. In the second step, we optimize the approximate teeth pose in image space (see Figure 4.2 e, f) using the dental impression of the patient. We refer to it as the *refined teeth pose*. The refined teeth pose estimation requires that parts of the patient's original teeth are visible. In cases where only small portions of the original teeth are visible, the refined teeth pose might be inaccurate. To achieve a robust outcome, we compare the refined teeth pose and the approximate teeth pose. If they are dissimilar more than a predetermined threshold, we use the approximate teeth pose for the current frame. If a patient does not have original teeth, it is difficult to create a denture preview because the patient's lips are deformed, especially when smiling. Although a denture preview is possible with the approximate teeth pose, the lip deformation changes the shape of the smile and thus, leads to inaccurate results. In such a case, we suggest to use a temporary denture for the patient during the video recording. This temporary denture restores the shape of the lips and can be also used for the refined teeth pose estimation.

Approximation

Following our previous work in Chapter 3 [28], we compute an approximate teeth pose by solving a 3D-to-2D point correspondence between the facial landmarks and the 3D face model. We use the maxilla and mandible meshes from the BodyParts3D database [204] to create the respective models. The maxilla model contains points located on the upper part of the face (see red points in Figure 4.2 c) Similarly, the mandible model contains points located on the lower lip and the chin (see blue points in Figure 4.2 c). For every

video frame, we detect the face region and extract the facial landmarks (see Figure 4.2 b) using the Dlib library [167]. These landmarks correspond to the points placed on the maxilla and mandible models. We solve the 3D-to-2D point correspondence using a Levenberg-Marquardt optimization [182]—first, for the maxilla (see Figure 4.2 d). For the mandible, we follow the same procedure as for the maxilla, but we use the earlier detected maxilla pose as a starting configuration. This improves the accuracy of the estimation, as the mandible model contains fewer feature points and is more difficult to detect, compared to the maxilla.

Refinement

We refine the approximate maxilla pose in image space, using the jaw impression of the patient (see Figure 4.2 e). To measure how well the estimated teeth position matches the actual teeth position, we compute the following features for every video frame: tooth edges, tooth color, and tooth luminance. We define the refinement procedure as an optimization problem with the following optimization function:

$$T = w_e M_e + w_c M_c + w_l M_l, \tag{4.1}$$

where T is the maxilla pose defined by a 3D position vector, and by three Euler rotation angles; M_e, M_c, M_l are matching metrics for edges, color, and luminance; w_e, w_c, w_l are corresponding weights for the metrics.

To solve Equation 4.1, we use a pattern search optimization strategy [146] that starts from the initial maxilla pose and refines it by searching a better pose in a neighborhood of the parameters. For the current pose and all candidates, we iteratively compute the function value. If we find a better placement, we use this as the initial pose for the next iteration. Otherwise, we increase the radius of the investigated parameter neighborhood and repeat the search process. If the neighborhood has reached the maximal radius, we stop and consider the currently best pose as the optimum.

Edge Matching: We perform edge matching in the gradient image space. First, we apply Sobel edge detection [163] with the same parameters on both images, i.e., the denture image and the natural teeth image are filtered with the same kernel. Second, we remove all non-dental edges like the lips, and face skin irregularities, using the facial landmarks. This avoids solutions of the optimizer, where the dental objects snap to non-dental objects. Third, we multiply the source and destination image edges to see how well the edges for the source and destination images are matching:

$$M_e = \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} I_s(x,y) I_d(x,y), \tag{4.2}$$

where W is the image width, H is the image height, I_s is the intensity of the source image, and I_d is the intensity of the destination image.

Color Matching: We convert the source image from the *RGB* to the *Lab* color space, which consists of three channels: lightness, green-red, and blue-yellow. We use the *Lab* color space, because this color representation is convenient to separate teeth pixels from

surrounding pixels belonging to lips, gingiva, and tongue. Moreover, the teeth are well distinguishable from the lips in the green-red channel. We calculate the color matching using the following formula:

$$M_c = \frac{w_a\sigma_a + w_b\sigma_b}{I_{max}}, \tag{4.3}$$

where σ_a is the standard deviation of the green-red channel, σ_b is the standard deviation of the blue-yellow channel, w_a, w_b are the corresponding weights, and I_{max} is the maximal intensity value.

Luminance Matching: Teeth are typically brighter than their surrounding area and thus can be well differentiated in the lightness channel. We compute the luminance matching as follows:

$$M_l = \frac{\mu}{I_{max}}, \tag{4.4}$$

where μ is the mean intensity value of all pixels belonging to teeth, and I_{max} is the maximal intensity value.

4.4.2 Aesthetics Analysis

Designing dentures requires knowledge and, especially, experience. Considering the user tasks and literature in cosmetic dentistry, we propose a solution for the aesthetics analysis of dentures. This solution enables dental technicians to assess denture aesthetics and to identify and correct possible design flaws. During dental design, technicians often concentrate on a small part, e.g., a single tooth, and must not forget to consider the overall dental aesthetics. Our approach addresses this by tracking several states of the denture and visualizing them on demand, to inspect how changes affect the aesthetics. Additionally, our method computes aesthetic metrics in the image space for the current denture design. With our aesthetics analysis method, dental technicians can perform fine denture adjustments, i.e., they can transform and rotate every individual tooth. Together with the tooth manipulation, we deform the gingiva with interactive Laplacian mesh editing [251] to match the new teeth layout. The control vertices are those which are inside the teeth meshes. Their position is kept fixed relative to the respective reference tooth which contains them. The deformation's region-of-interest is obtained by applying a dilation on all control vertices with a distance of 5 mm. Interactive refresh rates are achieved by splitting the Laplacian solving into a pre-computed factorization pass and an update pass. Also, immediately after the manipulation of the denture, we recompute all metrics in real-time, providing responsive feedback. This addresses *R 6*.

Denture Preview

In the denture preview, we augment the video of the patient with the current denture model. We identify the mouth region from the landmarks and find the upper and lower lip smile lines. Using these lines, we mask out the mouth region in the video frame with a dark red color that mimics the color of the tongue and gingiva. We compute the teeth pose and apply the corresponding rotation matrix and the translation vector to

Figure 4.3: The denture preview shows the face of the patient, augmented with a denture. The denture can be changed or adjusted during the aesthetics analysis. On the right-hand side, there are three dentures of different size: small (S), medium (M), and large (L).

the denture model. Then, we render the denture model and combine the rendered image with the video frame. Figure 4.3 shows a denture preview for three dentures of different sizes. The denture preview addresses *R 1*.

Best Viewing Angle: The frontal view of the face is one of the most important views for the aesthetics analysis, as it highlights problems related to the denture placement or the teeth symmetry, for instance. We developed a procedure that searches for the most frontal frame in the video. For each video frame, using the teeth pose, we compute the angle between the view direction of the teeth and the camera. We select the frame with the smallest angle and use it as a suggestion for the aesthetics analysis.

Facial Proportions

Dental technicians use facial proportions to identify a suitable denture size, shape, and position [183]. Figure 4.4 a–k shows the most important facial lines. Further in the text, we will refer to these lines with the corresponding letter assigned in Figure 4.4. The ratio of the distance between lines *(b)* and *(d)* and the distance between lines *(d)* and *(e)* (see Figure 4.4 l, top) should be approximately equal to the ratio of the sum of the widths of the first three teeth (central incisor, lateral incisor, and canine) to the half-width of the mouth (see Figure 4.4 l, middle). These two ratios should also be approximately equal to the ratio of the width of the central incisor to the width of the lateral incisor (see Figure 4.4 l, bottom).

We use this facial view to guide the dental technicians during the aesthetics analysis. The facial proportion view augments the denture preview with facial lines (see Figure 4.10 a–c and Figure 4.11 a–g). Vertical lines help dental technicians to identify an appropriate denture size, and horizontal lines assist them to define the denture placement. The facial proportion view addresses *R 7*.

Dental Proportions

Teeth proportions are often used in cosmetic dentistry as an aesthetic metric. Often dental technicians use the golden ratio to define natural proportions between teeth [187]. For natural teeth, the ratio of the central incisor width to the lateral incisor width is approximately equal to the golden ratio (see Figure 4.4 n). Also, the ratio of the lateral incisor width to the canine width and the ratio of the canine width to the first premolar width is approximately equal to the golden ratio. The visible tooth area depends on the view angle. The visible tooth widths or proportions are, therefore, view-dependent and must be calculated separately for each video frame.

To extract the teeth proportions, we render the denture model into a labeled image (see Figure 4.2 g). We then use a threshold segmentation to obtain a mask for each tooth and calculate the extents of the region masks. Horizontal extents indicate teeth boundaries, and all together they depict teeth proportions. The dental proportions view augments the video frame with vertical dental lines, and both the face of the patient and the denture are visible (see Figure 4.10 d). This view highlights the current teeth proportions and allows dental technicians to assess the symmetry and size of the denture during fine-tuning. Because dental technicians can use dental proportions to detect and correct improper denture placement and size, this view meets *R 10*.

Dental Histogram

The amount of visible gingiva has a significant influence on the smile aesthetics [88]. To compute a dental histogram, we identify the mouth region using facial landmarks and render the visible part of the denture behind the lips as the labeled image (see Figure 4.2 g). Using this labeled image, we segment the gingiva and teeth. Since dental technicians are interested in the horizontal line separating the gingiva from teeth, we compute the number of gingiva and teeth pixels for every row of the image. For every row in the image, we compute the number of gingiva pixels and the number of teeth pixels. This provides us with two distributions, which we visualize in the dental histogram (see Figure 4.11 h, i).

Typically, dental technicians aim to have the gingiva with 2–3 mm of visible height [88]. Based on the dental histogram, we define the dividing line between the gingiva and teeth at the location where the numbers of gingival and teeth pixels are equal, i.e. where we have an evenly mixed distribution of pixels between the two materials. Since the dental histogram provides a suggestion for the accuracy of the denture placement, it satisfies *R 10*.
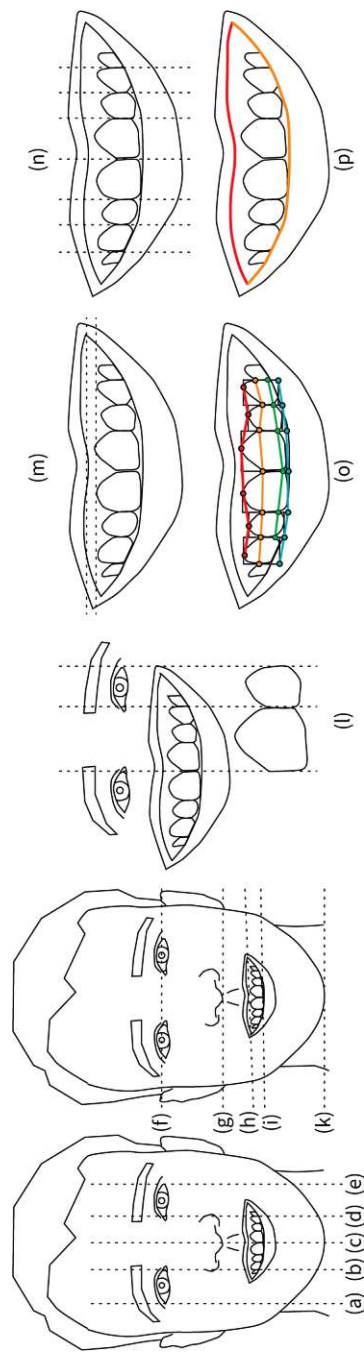
Figure 4.4: Facial and dental metrics used in the aesthetics analysis: the two lateral palpebral raphe lines (a), (e), the two intercommissural lines (b), (d), the symmetry line (c), the interpupillary line (f), the acanthion line (g), the intercommissural line (h), the incisor line (i), and the mentum line (k); eyes, mouth, and teeth are in harmony as they have similar internal proportions (l); lip line height (m); proportions between teeth (n); from top to bottom: cervical, papillary, contact points, incisal smile lines (o); upper lip and lower lip smile lines (p).

Smile Lines

Smile lines are important to quantify dental aesthetics and to identify main flaws in the denture design [89, 187]. They connect features, which are relevant for the aesthetics analysis of a smile. The line shape provides a visual guideline for the dental technician on the smile aesthetics. To address *R 8*, we compute the six aesthetic smile lines presented by Câmara [61]. Figure 4.4 o, p lists and illustrates these smile lines, and Figure 4.5 a–d shows these smile lines computed on the face of a person with our method. The *upper-lip smile line* and the *lower-lip smile line* are computed directly from the facial landmarks. The other lines are computed from the labeled image of a denture (see Figure 4.5 e, f).

Cervical Smile Line: The cervical smile line connects the upper corner of the visible parts of the upper teeth. To calculate this line, we extract the tooth contours and search for the point with the lowest value on the Y-axis (vertical axis, pointing downwards) for each tooth. We connect these points to obtain the cervical smile line.

Papillary and Contact Points Smile Lines: The papillary smile line connects the tips of the gingiva between the canines and the lateral incisors, and between the maxillary lateral incisors and the central incisors. To compute the papillary and the contact points smile lines, we apply a dilation operator to the teeth masks and compute their overlapping areas. The kernel of the dilation operator should only modify the object along the X-axis (horizontal axis) to ensure the correct position of the points. After computing the contours of the overlapping regions, we determine the upper and lower contour points. The papillary smile line is defined as the line connecting the upper points, while the contact points line connects the lower points.

Incisal Smile Line: The incisal smile line connects the edges of the anterior maxillary teeth. To compute this line, we extract the teeth masks and calculate the corresponding bounding boxes. From the bounding boxes, we find the lowest edges, and we connect them.

Upper-Lip and Lower-Lip Smile Lines: We compute the upper and lower lip smile lines only from the facial landmarks. Following our previous work in Chapter 3 [28], we increase the number of points on the lips.
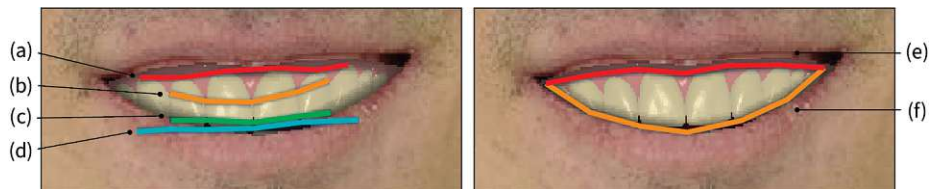

Figure 4.5: Six aesthetics smile lines. Cervical (a), papillary (b), contact points (c), incisal (d), upper-lip (e), and lower-lip (f) smile lines.
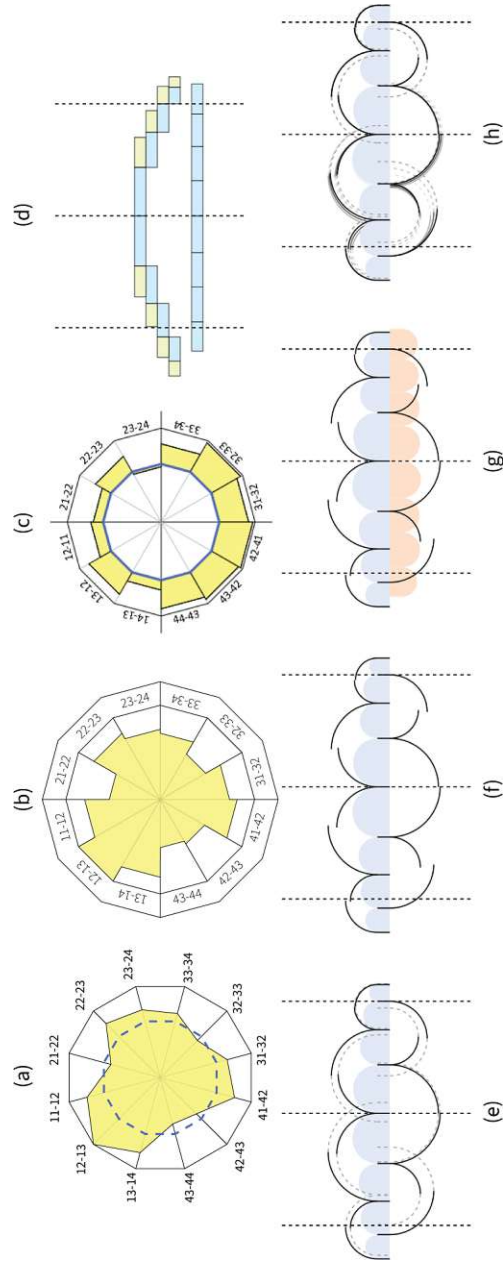
Figure 4.6: The evolution of the design of the abstracted facial and dental proportions view in chronological order: the radar visualization (a)–(c), the stairway visualization (d), and the modulor visualization (e)–(h).

Harmony View

The face has repetitive proportions at different scales, i.e., the eyes, mouth, and teeth have similar proportions [183] (see Figure 4.4 l). To evaluate and compare them, we designed the *harmony view*. The harmony view shows eye, mouth, and teeth subviews, and their corresponding internal proportions (see Figure 4.10 e). Dental technicians can choose one as the reference subview, and we align all other subviews to the reference one. In the subviews, we show the actual proportions as small blue lines and the reference proportions as red lines. The employed harmony view addresses *R 9*.

Abstracted Facial and Dental Proportions

Dental technicians often use the golden ratio to approximate the desired proportions for the upper teeth [183]. Lower teeth are, in many cases, not visible. Therefore, dental technicians usually adjust them to achieve a good functional state of the denture. Nevertheless, the lower teeth proportions are also important for the aesthetics analysis. Together with our collaborators, we designed three visualizations for abstracted facial and dental proportions—namely, a *radar*, a *stairway*, and a *modulor view*. We compared and assessed these techniques to select the most suitable one for integration in the dental CAD system. This was found to be the modulor view, as we discuss below. In this section, we briefly explain the techniques and our methodology for choosing the most appropriate one.

The **radar** visualization displays the proportions of the upper and lower teeth in a circular arrangement that mimics the teeth layout (see Figure 4.6 a–c). Next to each bar, we show a label indicating the ratio of the teeth pair that the bar represents. The ratio of a teeth pair is defined as the ratio of the widest to the thinnest tooth and is always between zero and one. The design in Figure 4.6 a, c includes a blue polygonal line that indicates the golden proportion. The line assists technicians in estimating how much the actual proportion deviates from the desired one. The advantage of the *radar* visualization is an intuitive and clean representation of teeth proportions that allows technicians to compare various dentures in a short time. The main disadvantage of the *radar* visualization is that it includes only dental proportions, but no facial ones.

The **stairway** visualization depicts the proportions for the upper and lower teeth with horizontal bars. It also includes three facial lines—the *symmetry line*, the *left eye line*, and the *right eye line*. Figure 4.6 d shows an example of a stairway visualization. Every bar consists of two rectangles and represents a proportion of the upper teeth pairs. The blue rectangle shows the actual tooth width and the yellow rectangle shows the required width of the next tooth so that both are in perfect proportion to each other, e.g., in the golden ratio. We shift the upper teeth bars down for the next teeth pairs to avoid overlaps. At the bottom, we show the sizes of the lower teeth, which consist only of blue rectangles representing the actual teeth widths. The advantage of the *stairway visualization* is that it shows dental as well as facial proportions. However, it shows the estimated perfect proportion only in one direction, e.g., for the lateral incisor it shows
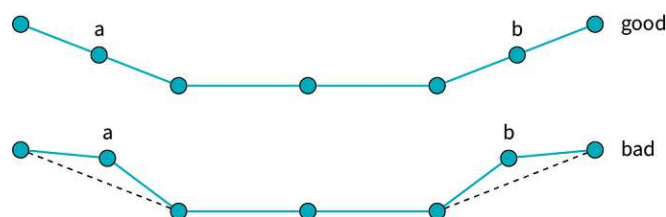
Figure 4.7: The shape of the incisal smile line affects the smile score. The upper and lower rows show a good and a bad incisal smile line shape, respectively.

only the canine tooth, but not the central incisor proportion. We can resolve this by displaying yellow bars on both the left and right sides. In some cases, however, this leads to overlapping bars. In an informal interview with our collaborating dental technician, we noticed that the visualization is less intuitive than, for instance, the *radar* visualization, and typically requires a steeper learning curve for an inexperienced dental technician.

The **modulor** visualization encodes the teeth in glyphs and the teeth proportions in arcs (see Figure 4.6 e–h). Every ratio of a tooth pair has two arcs. For example, the left central incisor and the left lateral incisor are typically in the golden proportion. If they are not in the golden or required proportion, this can be interpreted in two ways, depending on which tooth is considered to be the reference one. If the lateral incisor is the reference then the central incisor is too wide. If the central incisor is the reference then the lateral incisor is too narrow. To accommodate this ambiguity in the modulor visualization, we show the ratio between two teeth as two arcs. One arc takes the size of the first tooth as the reference, while the other arc takes the size of the second tooth as the reference. If the teeth are in perfect proportion to each other, these two arcs fit together and look like one continuous arc, without any interruption. Otherwise, the vertical distance between the two arcs indicates the extent of the ratio violation and the direction in which the adjustment must be made to improve the denture aesthetics. The design of the visualization was inspired by the work of Le Corbusier, the Modulor [75]. The advantage of the modulor view is that it shows all dental and facial proportions, as well as bi-directional proportions, and it provides an easier and cleaner reading of the proportions.

In all three presented visualizations, we depict previous dental states as visual trails (see Figure 4.6 h for the modulor view). This allows dental technicians to perform fine denture adjustments while focusing on the denture itself, and then assess how additional changes affected tooth proportions and smile aesthetics. Abstracted facial and dental proportions enable the assessment and comparison of different dentures (*R 2*). They can be used to identify main dental issues, e.g., asymmetric dental layout (*R 4*). Additionally, they show golden proportions for the upper teeth (*R 5*).

Aesthetometer

Using the extracted facial and dental features, we calculate the *total smile score*. Since smile aesthetics are subjective and may vary depending on many different parameters, this *score* is also subjective. However, in case of subtle differences, this score is particularly useful for quantitatively comparing two denture designs. The total smile score consists of three different scores, and is computed according to the formula:

$$S = S_t + S_l + S_i, \tag{4.5}$$

where S_t is the teeth proportion score, S_l is the symmetry score of the aesthetic lines, and S_i is the incisal smile line score.

The teeth proportion score reflects how well the current teeth proportions match the reference proportions. We calculate this score by the following formula:

$$S_t = \sum_{i=1}^{N} w|r_i - r_{i,d}|, \tag{4.6}$$

where N is the number of teeth ratios, w is a user-defined weight, r_i is the i-th ratio, and $r_{i,d}$ is the desired ratio for the i-th pair of teeth.

Since all aesthetic lines should be symmetric in the frontal view, we use symmetry to highlight good denture designs. We compute the symmetry score as following:

$$S_l = \sum_{i=1}^{6} \frac{\sum_{j=1}^{\lfloor L_i/2 \rfloor} |p_{i,j} - p_{i,L_i+1-j}|}{D_i}, \tag{4.7}$$

where L_i is the number of points on i-th line, $p_{i,j}$ is the j-th point of i-th line, p_{i,L_i+1-j} is the antagonist point of the j-th point of the i-th line, D_i is the length of the i-th line.

We compute the incisal smile line score, as following:

$$S_i = 1 - \frac{w}{D}\left(|a - a_{best}| + |b - b_{best}|\right), \tag{4.8}$$

where w is a user-defined weight, a is the left evaluated point of the incisal smile line, a_{best} is the best location of the left evaluated point, b is the right evaluated point of the incisal smile line, and b_{best} is the best location of the right evaluated point (see Figure 4.7), and D is the length of the incisal smile line.

We visualize the smile score in the aesthetometer, using a gauge visual metaphor. The aesthetometer looks like a car speedometer or a pressure gauge (see Figure 4.12 c, g). If the score is high, the arrow points towards the right. If the score is low, the arrow points to the left. At the bottom of the aesthetometer, we display the numerical value of the smile score. To track changes of the gauge, we display previous values as trails that fade over time. A similar trail concept for gauge widgets was presented by Matkovic et al. [194]. The aesthetometer addresses *R 3*.
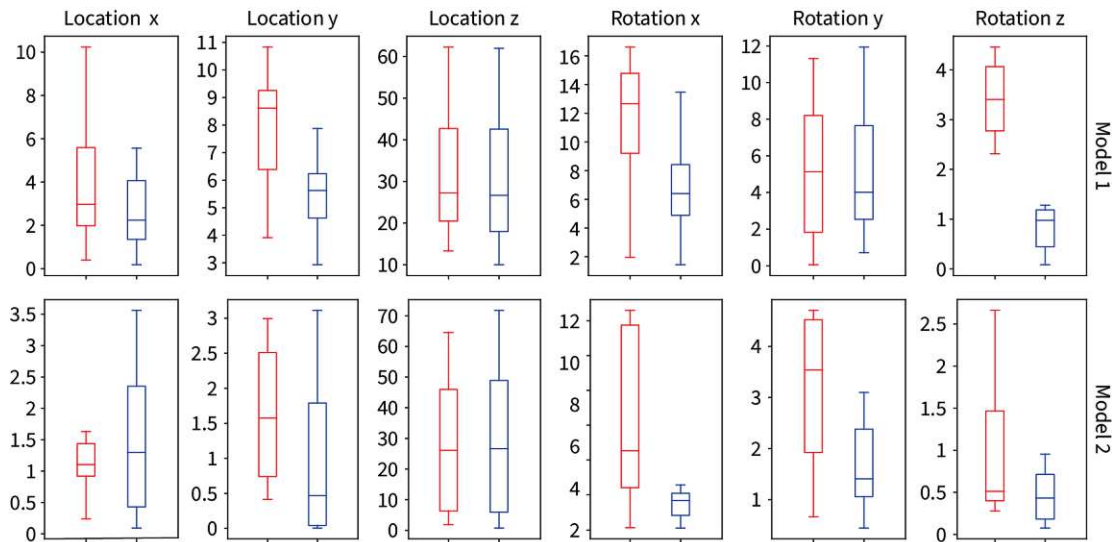
Figure 4.8: The upper and lower rows show the error for the first and the second gold standard models, respectively. Box plots depict the mean value and the distribution of the error for the approximate teeth pose estimation (blue), and the refined teeth pose estimation (red) in six dimensions (X, Y, Z coordinates and X, Y, Z Euler rotation angles). The coordinates are measured in millimeters, and the angles are measured in degrees.

4.5 Results and Evaluation

After designing the system, we performed a quantitative evaluation of teeth pose estimations. Besides, we conducted an informal interview with a dental technician. Together with the dental technician, we investigated four use cases to assess various components of the system and to identify how well the proposed solution solves the real-world tasks.

4.5.1 Teeth Pose Estimation

To evaluate the accuracy and precision of the teeth pose estimation technique, we acquired gold standard data using photogrammetry. We reconstructed two textured face models of a person from a set of high-resolution photographs, but with different facial expressions. Using a single full-frame DSLR camera, we took 19 and 25 photos of the person's face from different angles, for the first and second models respectively. To minimize facial motion between photos, we did the entire procedure in a short time frame (6 and 4 seconds, respectively). We subsequently manually aligned the scanned teeth model of the same person with the head model so that the teeth model matched the contours of the teeth that were visible on the texture of the head model. Then, we rendered each head model with the visible natural teeth from different view positions and directions, defined by the matrix transformations $m_1, ..., m_n$. Consequently, we obtained $n = 30$ and $n = 12$ images that look similar to the original photos and were used as input for both

teeth pose estimating techniques. For both models, we compared the approximated and the refined teeth pose against the gold standard, i.e., the position of the aligned teeth model with the additional transformations $m_1, ..., m_n$. Figure 4.8 shows the mean error and the error distribution of both teeth pose estimation techniques (approximated and refined) in all six dimensions (3D location vector and 3D rotation vector) for the two head models. The overall accuracy of the presented imaged-based teeth pose estimation (refined) is higher than that of the landmark-based teeth pose estimation (approximate).

4.5.2 Use Cases

Use Case A—Choosing the Best Aesthetically Fitting Denture: Our collaborating dental technician has a large library of teethmasters among which he wants to identify the one that best fits the patient. He receives a prerecorded video of the patient, consisting of 340 frames, and he uses the denture preview (see Figure 4.9 a) to assess the aesthetics of the individual dentures. He goes through the video to preview the overall denture fitting. Later, he identifies a suitable frame for the aesthetics analysis and the comparison of dentures using the best viewing angle identification procedure. As a direct visual comparison might be challenging, he enables the modulor view (see Figure 4.9 c) to analyze the facial and dental proportions. Afterward, he switches to a different teethmaster (see Figure 4.9 b), and receives the updated denture preview, as well as the modulor view. Based on the facial shape and the denture proportions, the dental technician compares and assesses all available teethmasters to decide which one fits best for the patient. In this case, the modulor view shows the smallest overall discrepancies for the third denture (see Figure 4.9 c, bottom). This is the best fitting denture, concerning the frontal teeth.

Use Case B—Finding the Suitable Denture Size: The size of the denture must match the size of the face and the facial proportions of the patient. Using the facial proportions (see Figure 4.10 a–c) as well as the dental proportions (see Figure 4.10 d), our collaborating dental technician identifies the correspondence between the denture and the face. If the denture has the proper size, the (b) and (d) lines (see Figure 4.4) should go through the border between canines and first premolars (see Figure 4.10 a, c). If the denture is too small or too large, he scales the denture up or down or, in some cases, changes the denture preset. Afterward, he uses the harmony view to crosscheck the proportions at different scales. In the harmony view (see Figure 4.10 e), he can choose the reference proportions, e.g., the distance between eyes and the eye width. In this case, the selected denture size fits well.

Use Case C—Adjusting the Denture Position: The denture position and orientation play an important role in the smile aesthetics. Also, they directly define the size of the visible gingiva surface and the dental surface. Dental technicians employ the term of red-white aesthetics to describe this relationship. Using the facial proportion view, our collaborating dental technician adjusts the vertical denture position. He looks at four horizontal lines, namely lines (f), (g), (i), and (k) (see Figure 4.4), which should form two golden ratios (see Figure 4.11 a–d). Besides, he adjusts the denture orientation using lines

(f), (h), and (i) (see Figure 4.4). If lines (i) and (h) are not parallel, the orientation of line (i) must be interpolated between the orientation of line (f) and (h) (see Figure 4.4). Then, he analyzes the red-white aesthetics in the dental histogram view that shows the distribution of the gingiva and dental pixels in the image. For example, if the number of gingiva pixels is too small (see Figure 4.11 h), dental technicians can move dentures down to increase their number and reach the target proportion (see Figure 4.11 i).

Use Case D—Fine Denture Adjustments: During the fine denture adjustments or teeth editing, dental technicians modify the shape of individual teeth and the gingiva to improve the functional aspect of the denture. Our collaborating dental technician uses a virtual articulator to simulate the chewing and biting patterns of the patients and identify functional issues of the teeth, which often result in changing the teeth layout and editing. To preview and assess these changes concerning aesthetics, dental technicians can use a combination of different views. Our collaborating dental technician uses the smile lines view (see Figure 4.12 a) to get a summary on the current teeth layout, and checks if there are some issues related, for instance, to an asymmetric teeth placement or incorrect smile arc. Using the modulor view (see Figure 4.12 b), he assesses the dental and facial proportions. If the proportions of teeth are sub-optimal concerning the aesthetics, the dental technician can modify the teeth shape, size, and layout. The modulor view also encodes information about several previous changes as trails, enabling dental technicians to analyze their previous manipulations. If dental technicians are unsure about the current smile aesthetics, they can use the aesthetometer (see Figure 4.12 c, g). Using the harmony view, dental technicians see the correspondence of proportions at different scales (see Figure 4.12 d, h). Every tooth adjustment results in an automatic metric recalculation and update of the views. Figure 4.12 e–h show updated views, for adjustments in the left frontal incisor and the left canine, from the view of the patient.

4.5.3 User Experience

To evaluate the user experience, we completed all four usage scenarios together with our collaborating dental technician and asked him to rate the fulfillment of each task requirement regarding *effectiveness* (es), *efficiency* (ey), and *satisfaction* (s). For ranking, we applied a Likert scale from 1 (--) to 5 (++). Requirements related to *T 1—R 1* (es5 ey3 s2), *R 2* (es4 ey3 s2), *R 3* (es3 ey4 s3), *R 4* (es4 ey4 s3), *R 5* (es4 ey4 s4)—received positive grades in effectiveness and efficiency. This shows that the proposed solution can be successfully used to achieve the goal in a reasonable amount of time. However, the technician reported that the user interface is overloaded. The technician highly appreciated the different views for abstracted facial and dental proportions and noted that they are understandable and self-explanatory. The requirements of *T 2—R 6* (es4 ey4 s3), *R 7* (es4 ey4 s3), *R 8* (es4 ey4 s3), *R 9* (es3 ey4 s3), *R 10* (es4 ey3 s3), *R 11* (es3 ey3 s3)—received all positive grades regarding effectiveness, efficiency, and satisfaction. The technician reported that some operations required too many user operations to achieve the goal and some important facial lines were missing. We added these facial lines shortly after the evaluation. The technician appreciated the integration of the aesthetics
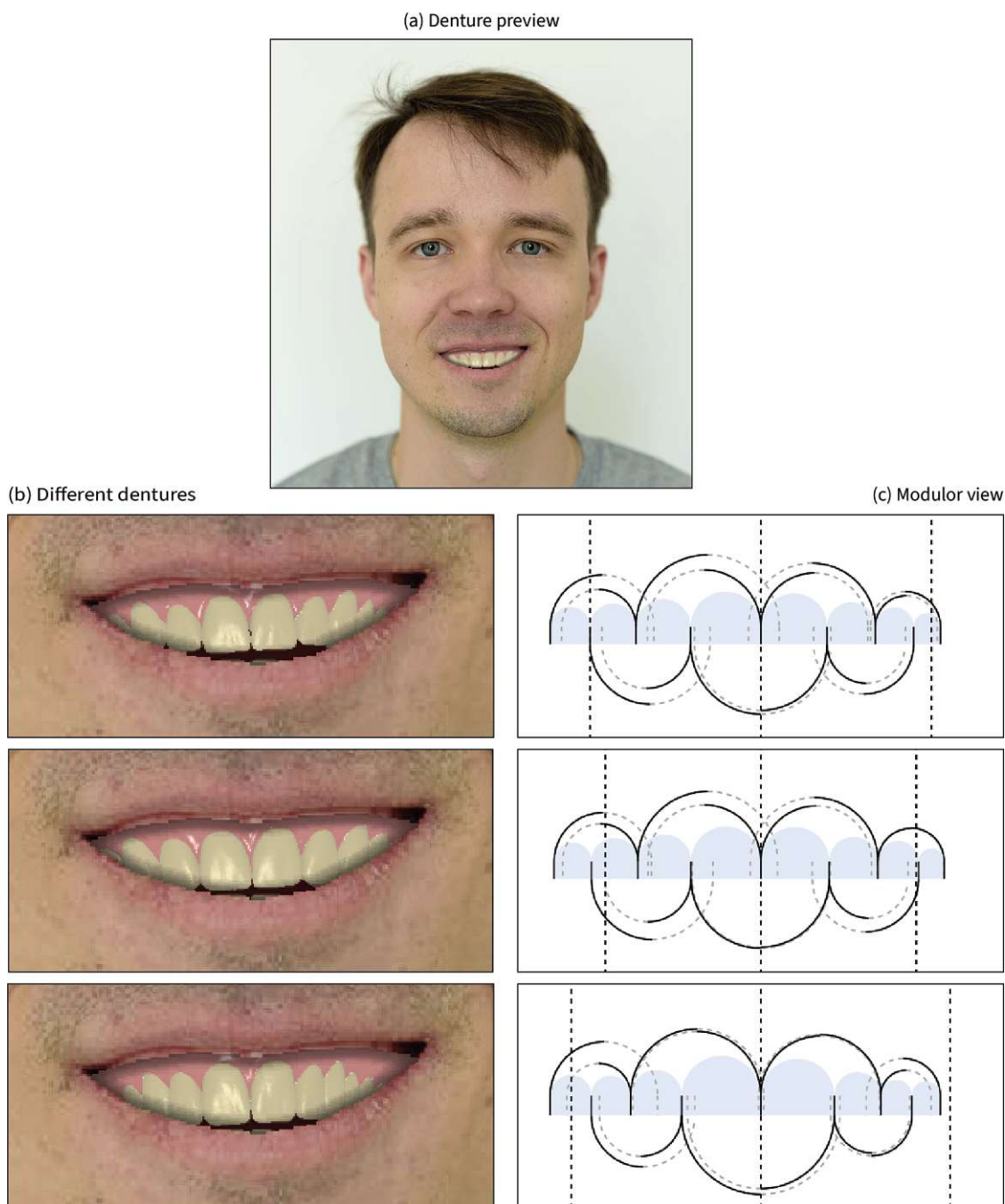
Figure 4.9: Comparison and assessment of different teethmasters conducted in Use Case A (Section 4.5.2) with the aid of the denture preview (a)–(b), and the modulor view (c).
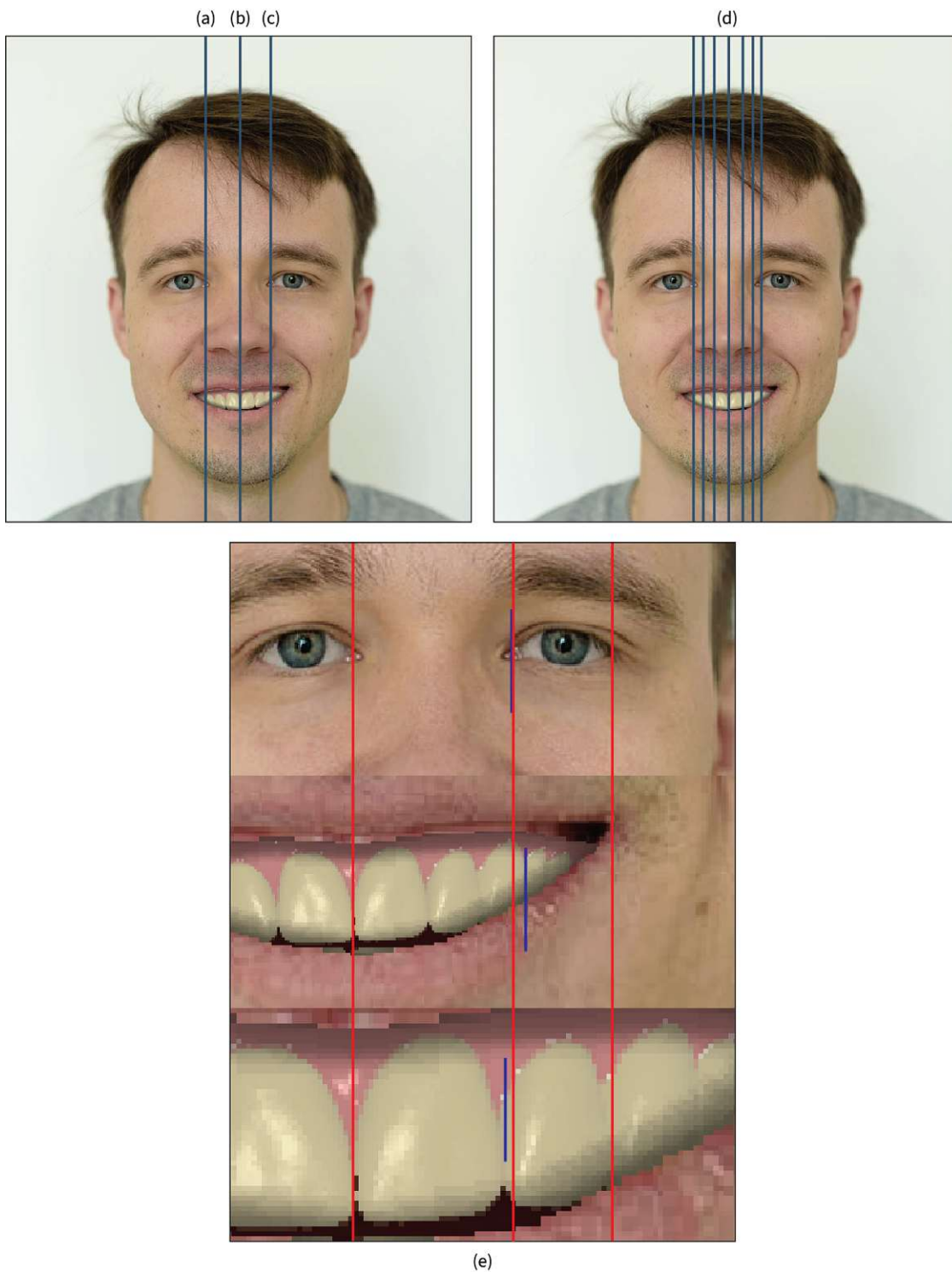
Figure 4.10: Analysis of the denture size conducted in Use Case B (see Section 4.5.2), using the facial proportions view (a)–(c), the dental proportions view (d), and the harmony view (e).
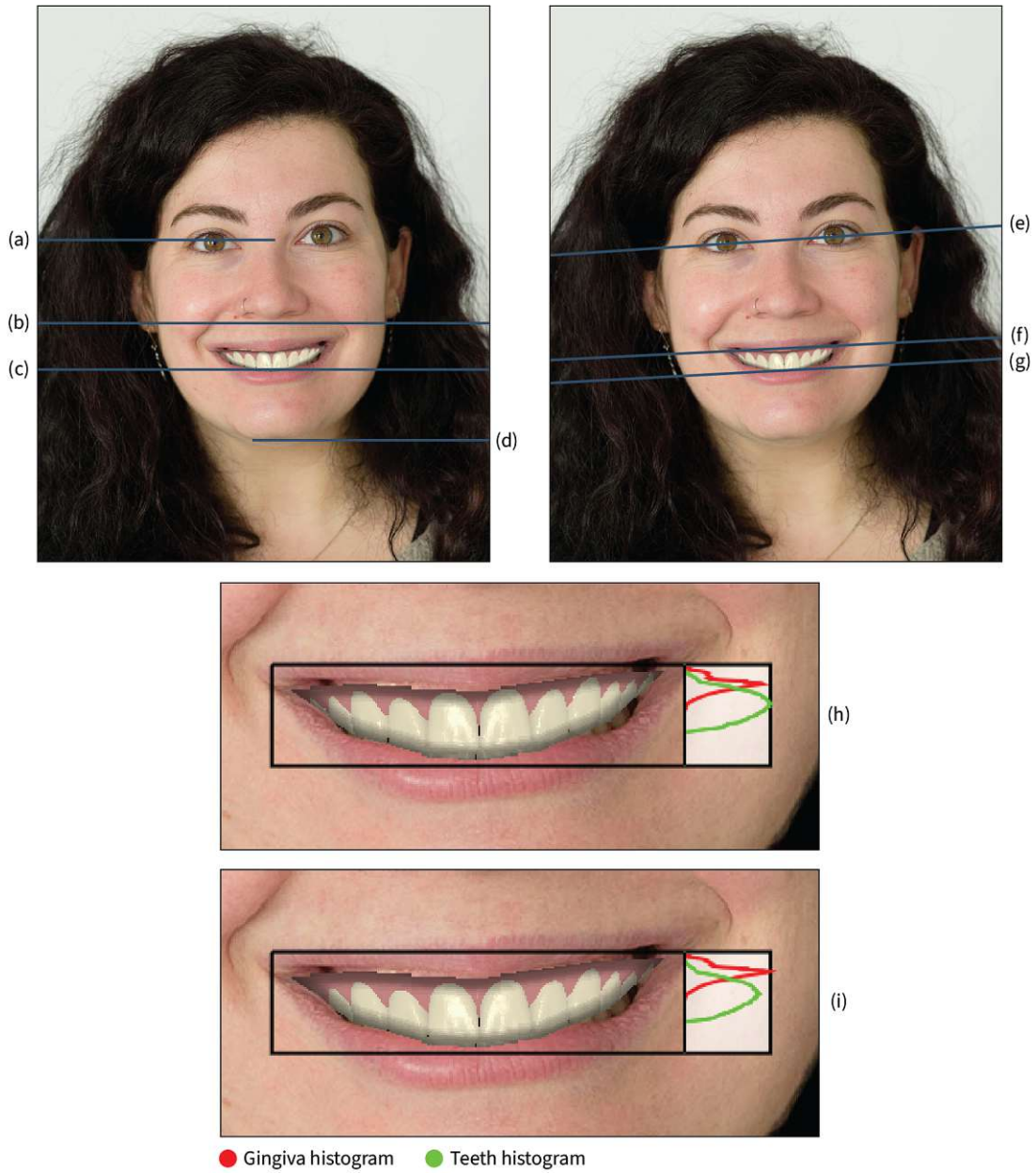
Figure 4.11: Analysis of the denture position conducted in Use Case C (see Section 4.5.2). Facial proportion views (a)–(g) identify the correct denture position. The dental histogram (h)–(i) provides an overview of red-white aesthetics.

Figure 4.12: Analysis of fine denture adjustments conducted in Use Case D (see Section 4.5.2). The smile lines (a), the modulor (b), the aesthetometer (c), and the harmony view (d) show the current aesthetics state. These views are automatically updated after every tooth editing operation (e)–(h).

analysis into the dental CAD workflow. He commented that the aesthetics analysis can be distributed over various steps of the workflow to avoid the cluttered user interface. For example, only some visualizations are ne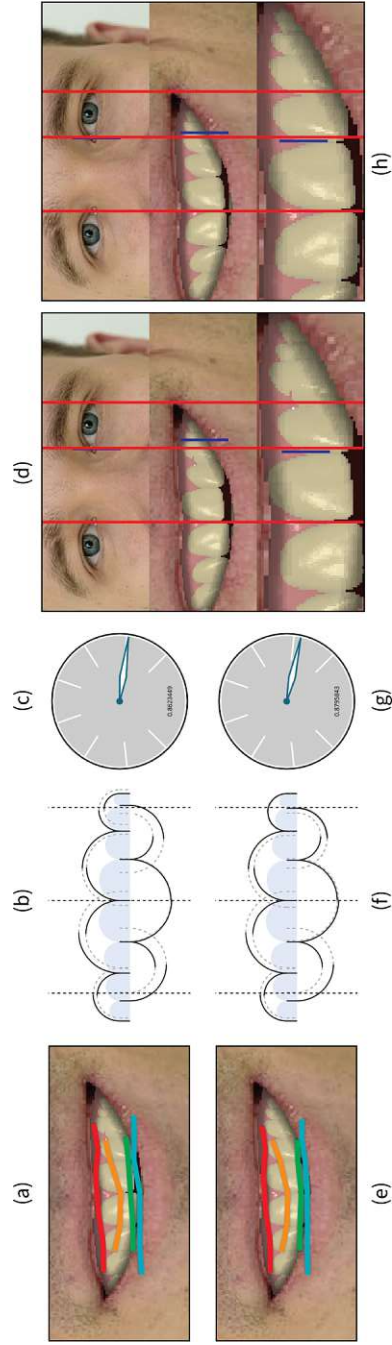cessary for the denture selection step. Besides, we applied the *System Usability Scale* evaluation scheme, which confirmed the previously mentioned observations on the system.

In the open discussion session of the evaluation, we talked about the limitations of the solution and the direction for future work. The main limitation is the complexity of the system and the overloaded user interface. A simpler solution with fewer visualizations should improve the learning curve of the system and facilitate the analysis. Also, smart suggestions about which views are helpful for the current analysis step are important. As future work, we plan to design an additional lateral side view for the aesthetics analysis of the patient. This view can be especially useful for the dental technicians to preview the denture placement outcome, as the denture modifies the lip shape. In this view, the shape of the lips must be modified, according to the denture position. Additionally, the fine denture adjustments should be done directly in the abstracted views. This should improve the performance of the designing process as some teeth editing operations require more user interaction. For example, to bring a tooth to the front, the camera view must be changed a couple of times.

4.6   Conclusions and Future Work

The modern dental reconstruction workflow enables the design and manufacturing of high-quality and cost-efficient dentures. For the first time, we integrated an interactive full aesthetics analysis into the current workflow, to provide a functional and an aesthetics analysis side-by-side. This combination allows dental technicians to preview the effect of any change in the denture design, both in terms of functionality and aesthetics. In this work, we made the first step towards a systematic smile design by quantifying several aspects of aesthetics, and by visualizing them during the denture design process.

There are several promising directions for future work. Firstly, the proposed method targets dental technicians as prime users of the system. Including the patient directly in the analysis procedure will increase the confidence and trust of the patient in the treatment outcome. In this case, the system can be used as a platform for communication between patients and technicians. Secondly, the current approach requires a video of the patient to preview the denture. Additional lateral views can be beneficial for the analysis. Providing a 3D recording of the facial performance of the patient, on one side will increase the complexity of the system; on the other side, it will enable additional views, e.g., the side view. Thirdly and finally, enabling the aesthetics analysis for partial dentures, e.g., comprising one tooth or two teeth, is a promising direction. In this case, the dental aesthetics of the patient are constrained not only by the facial features and the jaw anatomy, but also by the remaining natural teeth.

Paper D: improvement

Paper A: defect prevention

Paper C: performance evaluation

Paper B: control of defective outputs
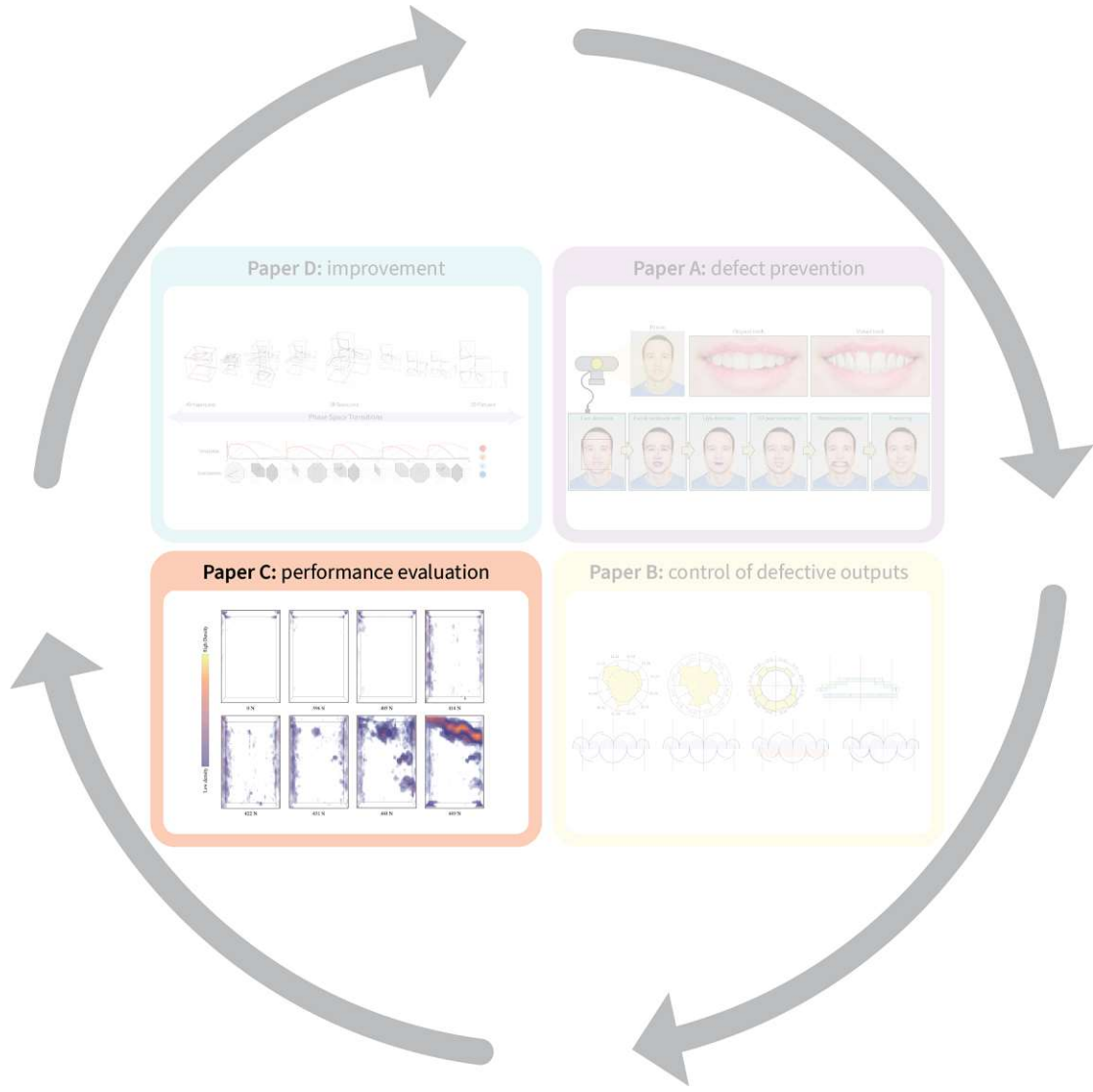
CHAPTER 5

Performance Evaluation

This chapter addresses the performance evaluation (i.e., the *check* phase of the PDCA cycle) and is based on the publication:

Material engineers use interrupted *in situ* tensile testing to investigate the damage mechanisms in composite materials. For each subsequent scan, the load is incrementally increased until the specimen is completely fractured. During the interrupted *in situ* testing of glass fiber reinforced polymers (GFRPs), defects of four types are expected to appear: matrix fracture, fiber/matrix debonding, fiber pull-out, and fiber fracture. There is a growing demand for the detection and analysis of these defects among material engineers. In this chapter, we present a novel workflow for the detection, classification, and visual analysis of defects in GFRPs using interrupted *in situ* tensile tests in combination with X-ray computed tomography (CT). The workflow is based on the automatic extraction of defects and fibers. We introduce the automatic Defect Classifier assigning the most suitable type to each defect based on its geometrical features. We present a visual analysis system that integrates four visualization methods:

- the Defect Viewer highlights defects with visually encoded type in the context of the original CT image,

- the Defect Density Maps provide an overview of the defect distributions according to type in 2D and 3D,

- the Final Fracture Surface estimates the material fracture's location and displays it as a 3D surface,

- the 3D Magic Lens enables an interactive exploration by combining detailed visualizations in the region of interest with overview visualizations as context.

In collaboration with material engineers, we evaluate our solution and demonstrate its practical applicability.

5.1   Introduction

Currently, composite materials attract a lot of interest from industry. Utilizing GFRPs enables the manufacture of cost-effective products with tailored mechanical properties. GFRPs consist of two components: glass fibers and a polymer matrix. Glass fibers provide the material with stiffness and durability. The polymer matrix bonds the fibers, protects them from external damage, and provides resiliency to the material. In order to achieve an improved material strength, the fibers are oriented during the manufacturing process. Low weight and high durability of GFRPs make them especially suitable for production in areas such as the aerospace and automotive industries. In order to produce GFRPs that satisfy industrial requirements, material engineers are highly interested in finding and analyzing the correlations between material features (e.g., the number of fibers, their directions and diameters, etc.) and the resulting mechanical properties, such as strength and ability to withstand high loads. This research knowledge is then utilized for optimizing the manufacturing process to produce superior composite materials.

3D X-ray computed tomography (3DCT) is an inspection method that is widely used for the non-destructive testing of composite materials such as GFRPs. Typical 3DCT scanners comprise three major components: X-ray tube, rotary plate, and detector. The inspected specimen is placed on the rotary plate between the tube and the detector, whereupon the scanning process starts. The X-ray tube directs X-rays through the specimen and onto the detector. The detector registers a 2D X-ray attenuation image also called projection image. The rotary plate is then turned and the process is repeated until a full 360° rotation of the specimen is completed. Tomographic reconstruction is performed using the set of 2D projection images to produce a 3D (volumetric) attenuation image. 3DCT has two main advantages over other non-destructive methods: 1) it allows the user to explore all internal and external structures with just one scan, and 2) it provides a high imaging resolution capable of resolving individual fibers and small defects. However, 3DCT cannot be directly used for the inspection of the specimen under various loads. The specimen has to be static during the scanning procedure, which can last up to several hours. To circumvent this limitation, interrupted *in situ* tensile 3DCT tests can be used.

Material engineers rely on interrupted *in situ* tensile testing to study the damage mechanisms in composite materials. During interrupted *in situ* tensile testing, a specimen is scanned under various loads created by the tensile stage. Typically, the first scan is
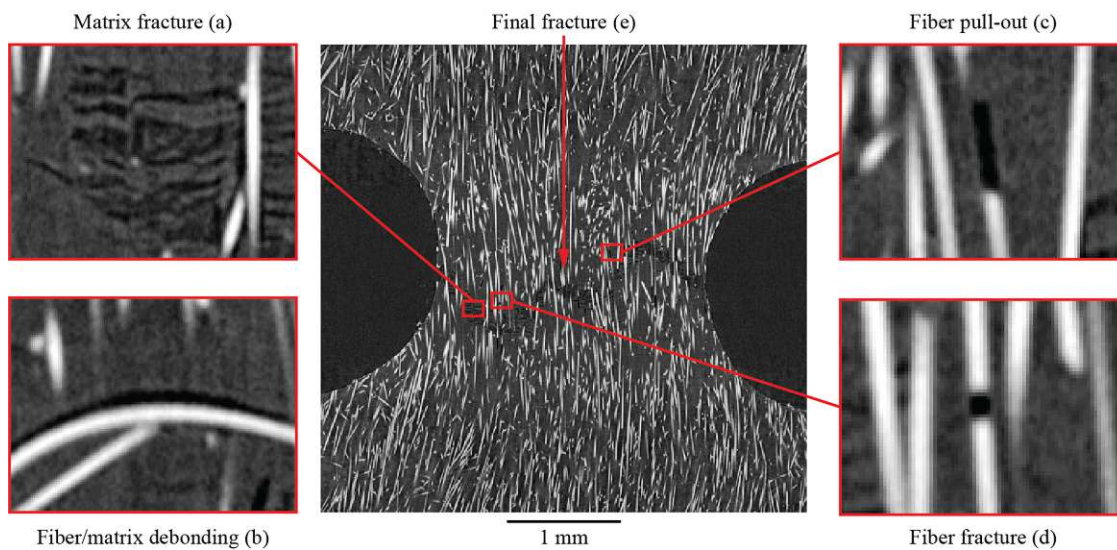
Figure 5.1: Four defect types on a 2D slice of a 3DCT volume coming from an interrupted *in situ* tensile test.

performed without any load. For each subsequent scan, the load is incremented until the specimen is completely fractured. The series of 3DCT scans will henceforth be referred to as 4DCT data, where the fourth dimension is the time or the amount of force applied to the specimen during the scan. If interrupted *in situ* testing is performed, defects of four types are expected to appear in the GFRPs: matrix fractures, fiber/matrix debondings, fiber pull-outs, and fiber fractures (see Figure 5.1). Detection, classification, and visualization of these defects is of a high interest to material engineers. Friedrich [113] described and illustrated each defect type (failure mechanism). However, the provided description does not formalize differences between defect types to an extent that is sufficient for an automatic defect classification. Therefore, in collaboration with our industrial partner Borealis AG, a major supplier of plastic materials for engineering applications in the automotive industry, we further detailed the defects' description. It resulted in the following list of definitions for each defect type:

Matrix fractures are mostly perpendicular to the force direction. They are often clustered and their shapes do not correlate with any shapes of fibers in the proximity.

Fiber/matrix debondings mostly affect fibers that are perpendicular to the force direction. They occur above or below fibers. Debondings have directions and shapes similar to neighboring fibers.

Fiber pull-outs mostly affect fibers that are aligned parallel to the force direction. They only occur at the ends of fibers and are cylindrical in shape. Diameter and direction of a pull-out are the same as those of the corresponding fiber.

Fiber fractures are located between two fiber pieces. They have a cylindrical shape. The fracture diameter is approximately equal to the fiber diameter.

If a force is applied, which is larger than a test specimen can sustain, individual defects of different types accumulate and *the final fracture* is formed as a result. The final fracture runs through the specimen and separates it into two parts (see Figure 5.1e).

In this work, we investigate the Fibremod GD301 material produced by Borealis AG. This material is a glass fiber reinforced polypropylene with a fiber content of 30% in weight. Due to its strong mechanical qualities this material is used in air bag housings, pedal carriers, pump housings, etc. During interrupted *in situ* tensile testing, a force inducing stress, strain, and failure was applied to the material. The material's failure is caused due to numerous individual defects that can be classified into the four previously mentioned types.

In Section 5.2 we proceed by deriving seven tasks that are important for the 4DCT analysis and present a workflow for solving these tasks. In Section 5.3 we provide a literature overview and discuss the related work. In Section 5.4 we give a detailed description of the datasets used in this work. Section 5.5 and Section 5.6 describe the workflow in greater details, that is, defect classification and tools for visualization and analysis are presented. In Section 5.7 and Section 5.8 we apply the proposed system and present use cases. We discuss the design choices made when implementing the presented visualization and analysis methods in Section 5.9. In Section 5.10 we provide conclusions.

5.2 Workflow and Task Analysis

Interrupted *in situ* tensile testing in combination with 3DCT is a relatively novel topic. Currently commercially available software such as VGStudio Max [16] or Avizo 3D [2] provide a wide range of functionalities such as segmentation, surface determination, feature extraction, porosity/inclusion analysis, and fiber analysis. However, the functionalities are not sufficient for the 4DCT data analysis and do not cover the requirements of our domain specialists. To the authors' knowledge, there is no tool currently available that can extract and classify defects from tensile tests of GFRPs and perform a visual analysis of these defects on the 4DCT data.

In this project, we collaborated with Borealis AG, a major supplier of plastic materials for engineering applications in the automotive industry. They aim to use the 4DCT tensile test data analysis to acquire new knowledge and insights on the correlation between fiber characteristics and resulting material properties. This knowledge is then used to improve the in-house fiber simulation software. In particular, the dynamics of defect development under load is of high interest. After several meetings and discussions, we identified the following list of tasks, which are of highest relevance for analyzing 4DCT data, and cannot be automatically implemented with existing software:
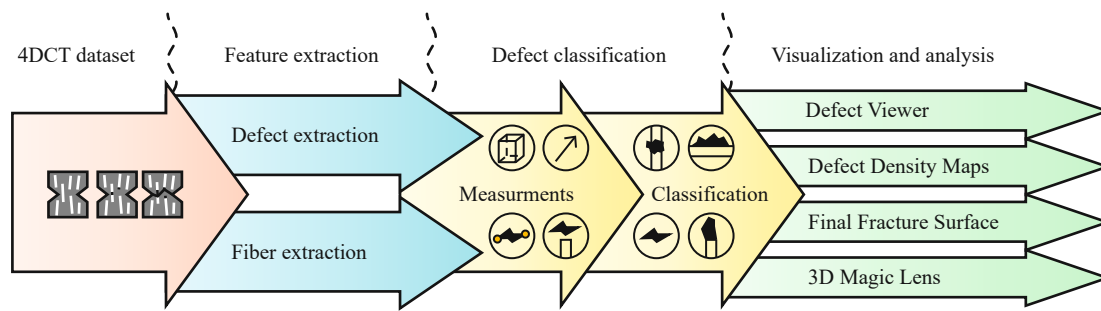
Figure 5.2: An overview of the workflow for 4DCT interrupted *in situ* tensile-test data-analysis.

T1: Finding individual defects and calculating defect characteristics, such as size, the center of mass, and direction.

T2: Classification of each individual defect by type.

T3: Visualization of individual defects.

T4: Finding and visualizing regions with the highest defect densities.

T5: Finding and visualizing the final fracture that appears when the specimen is completely fractured.

T6: Visualizing the history of the material destruction under the influence of increasing load.

T7: Presenting quantitative information on the final fracture and defects.

The purpose of the work presented here is to overcome the limitations of existing analysis techniques and to present a workflow and a system that address the tasks listed above. An overview of the workflow is shown in Figure 5.2.

Defect and fiber extraction: The first stage of the workflow is to extract individual defects and fibers (T1). For the defect segmentation, we use binary thresholding on the data that is pre-filtered with gradient anisotropic diffusion to reduce noise. For the fiber extraction, we employ the algorithm proposed by Salaberger et al. [241].

Defect classification: To classify defects (T1, T2), we introduce an automatic defect classifier based on the assumptions we made together with material engineers. Using these assumptions, a type is assigned to each defect in a 4DCT dataset. The complete list of assumptions for each type concerning defect characteristics as well as the details of the classification process are provided in Section 5.5.

Visualization and analysis: As the last stage of the workflow, we present a visual analysis system that provides a set of visualization techniques to address tasks T3–T7. A detailed description is given in Section 5.6. The system comprises four different visualization methods that can be combined or used sequentially to achieve the analysis goals:

- The *Defect Viewer* (Section 5.6.1) highlights defects (T3) in 2D slices of the CT data and provides visual encodings for the various defect types.

- The *Defect Density Maps* (Section 5.6.2) encode the defect distributions by type and provide visualizations of these distributions in 2D and in 3D (T4). In addition, Defect Density Maps provide quantitative information on the frequency of defect occurrences in a region (T7).

- The *Final Fracture Surface* (Section 5.6.3) estimates the shape and location of the final fracture as a 3D surface and displays it on top of the contextual information in a 3D view (T5).

- The *3D Magic Lens* (Section 5.6.4) is an interactive tool facilitating the analysis by combining detailed visualizations in the region of interest with overview visualizations as context in a single 3D scene. In addition, displaying data from different time-steps in the same scene is possible (T6).

5.3   Related Work

Visualization of 3DCT data: One of the first works utilizing visualization techniques for non-destructive testing using industrial 3DCT data was provided by Huang et al. [150]. The authors focused on feature extraction and visualized the data using direct volume rendering with two-dimensional transfer functions. Heinzl et al. [142] introduced a method for surface extraction from multi-material components using dual energy CT. Hadwiger et al. [132] presented a method for analyzing 3DCT data with the goal of interactively detecting, classifying, and quantifying features using a visualization-driven approach. The work by Fritz et al. [114] focused on characterizing steel fiber-reinforced concrete by segmenting fibers and quantifying their properties. A direction-sphere histogram provided an overview of the fiber orientations. Direction transfer functions defined to encode orientations are used to color-code fibers. More recent publications addressed the visualization of carbon- and glass-fiber reinforced composites using 3DCT scans [143]. Weissenböck et al. [280] presented the FiberScout tool for exploring and analyzing carbon fiber reinforced polymers. The authors used 2D information visualization methods, such as parallel coordinates, coupled with a scatter plot matrix, as well as 3D visualization techniques, such as blob surfaces and metadata visualizations. Within the same framework Reh et al. [229] implemented the mean object (MObject) technique for analyzing pores in carbon fiber reinforced polymers. The MObject technique was used for the exploration of hierarchically clustered pores and the identification of common patterns in the pore shapes.

Visualization of time-varying data: Havre et al. [137] presented the ThemeRiver approach for visualizing thematic variations over time within a large collection of documents. The work by Silver et al. [247] focused on tracking and visualizing features in computational fluid dynamics datasets. The authors extract features from each individual step and use spatial overlap to determine matching. Feature properties and their evolution history are used to enhance isosurface visualization and volume rendering. Van Pelt et al. [266] presented a framework for analyzing blood-flow dynamics and visualizing essential characteristics.

Segmentation and extraction of features and defects: Zhou et al. [296] used images from a camera that is fixed on a vehicle to analyze road surfaces for cracks. The authors presented an illumination-invariant image-enhancement and segmentation algorithm for the extraction and classification of cracks. Several approaches have been presented for an automatic crack detection in industrial 3DCT data. The work by Landstrom et al. [180] focused on crack detection in steel slabs. The authors presented an automated on-line system, based on 3D profile data of steel slab surfaces, which utilizes morphological image processing and statistical classification by logistic regression. Yamaguchi et al. [291] proposed an efficient and high-speed method for automatic crack detection in concrete surfaces employing percolation-based image processing. In civil engineering, the corrosion of steel reinforcements in structural elements of concrete bares the risk of reduced stability. The work by Paetsch et al. [213] explored various image processing and visualization methods for 3D corrosion detection in time-dependent 3DCT images of concrete. Specimens are exposed to several cycles of electrochemical pre-damaging to generate a large artificial crack. Bhowmik et al. [49] introduced a method for the detection and segmentation of cracks from 2D cross-sectional images of rock under compressive force. A two-dimensional matched filtering technique is followed by local entropy-based thresholding, morphological operators, and length filtering.

All the works as mentioned above focused on a single type of defect and usage scenario. In these works, 3D or even only 2D datasets are processed. 4DCT data is considered in the work of Paetsch et al. [213] and is visualized using side-by-side views. Reh et al. [228] presented a tool for evaluating 4DCT data. A tracking graph [283] was employed to relate pores in a specimen between scans. The system we are proposing provides means for the extraction and classification of four different defect types (matrix fractures, fiber pull-outs, debondings, and fractures) and methods for the detailed integrated visual analysis of these defects.

Various image segmentation methods are employed in medical and industrial 3DCT for feature detection and classification. These methods range from simple automatic thresholding methods, e.g., Otsu thresholding [212], to more complex probabilistic methods, such as random walks [125]. The visual analysis in our proposed system is not limited to a particular image segmentation algorithm. Our flexible system allows users to apply any segmentation algorithm for defect detection. The system we are proposing determines fibers using an algorithm presented by Salaberger et al. [241], which is designed for individual fiber extraction in fiber-reinforced composite materials.

5.4   Data Description

We performed series of measurements under load of three test specimens on a GE phoenix|xray nanotom® device. The corresponding 3DCT scans contain three components with different gray-value ranges (from high to low): fibers, matrix, and air.

Dataset 1: The specimen is notched in order to ensure failure in a predefined region. This is important, as high resolution 3DCT scanning with high magnification is only possible if the scan volume is reduced to a particular region of the test object. The force direction and the main fiber orientation are parallel, which leads to characteristic failure modes. It is expected that fiber pull-outs occur more frequently than fiber/matrix debondings. Four scans of a GD301-MiniNotched-OH1 object were performed with 0, 208, 224, 244, and 260 newton of force applied. The voxel size is 2 μm.

Dataset 2: This specimen is notched as well, the main fiber orientation and the force direction are perpendicular. Therefore, it is expected that fiber/matrix debondings will occur more frequently than fiber pull-outs. The scans were performed with 0, 187, 212, 225, 230, and 235 newton. The voxel size is 2 μm.

Dataset 3: The specimen is narrowed in the middle and extended at the ends. Similarly as for the notched specimens, this narrowed geometry guarantees that the failure happens in a predefined region. However, in this case the expected defect region is rather large in comparison to the notched specimens. In our dataset, the final fracture occurs at the top of the scanned region. The scans were performed with 0, 374, 396, 405, 414, 422, 431, 440, and 449 newton. The voxel size is 3 μm.

5.5   Defect Classifier

The Defect Classifier takes labeled images of defects and extracted fibers as inputs (see Figure 5.2). An automatic defect classification is then performed in two steps. In the first step, we measure geometrical features of the defects. In the second step, we assign the most suitable type to each defect based on the measured features (see Figure 5.3).

Measuring geometrical features of defects: For each defect the following characteristics are measured (see Figure 5.4): (i) length, width, depth; (ii) direction; (iii) volume (as the sum of a defect's voxel volumes); (iv) position; (v) and existence of neighboring fibers.

We use the LabelGeometryImageFilter as provided by the Image Segmentation and Registration Toolkit (ITK) [9] to derive these geometrical characteristics. In addition, for each defect, an object-oriented bounding box (OOBB) is computed. In order to determine the OOBB, the voxels of a defect are analyzed. The 3D positions of the voxel centers are used as data points for a principal component analysis (PCA). The PCA produces an orthogonal basis of three eigenvectors from the covariance-matrix, which are taken as the OOBB axes. The eigenvector with the largest eigenvalue indicates length, the
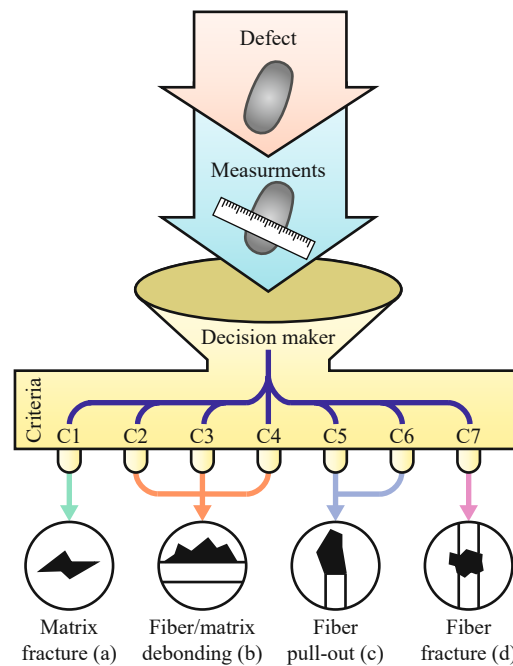
Figure 5.3: The pipeline for measurement and classification of individual defects. Defect types are assigned based on the combination of criteria C1-C7 that are evaluated: C1 - matrix fracture (a), C2, C3, C4 - fiber/matrix debonding (b), C5, C6 - fiber pull-out (c), and C7 - fiber fracture (d).

second largest eigenvalue indicates width, and the smallest eigenvalue indicates depth (see Figure 5.4). To calculate the exact extent of the OOBB, the voxel centers are projected onto the eigenvectors and min-max ranges are determined for each vector.

Assigning the defect type: To assign the most appropriate type to each defect we introduce a detailed list of criteria for each defect type. The derivation and formalization of these criteria were performed by material engineers from Borealis AG, who provided us with criteria for defect classification commonly accepted in industry. Recently, work by Rolland et al. [238] used a similar system of criteria for classifying defects in glass-fiber-reinforced thermoplastics. The resulting criteria (C1-C7) for the defect types are as follows:

Matrix fracture:

C1: The angle between the matrix fracture direction and the force direction is more than 25°.

Fiber/matrix debonding:

C2: The angle between the debonding direction and the force is more than 45°.
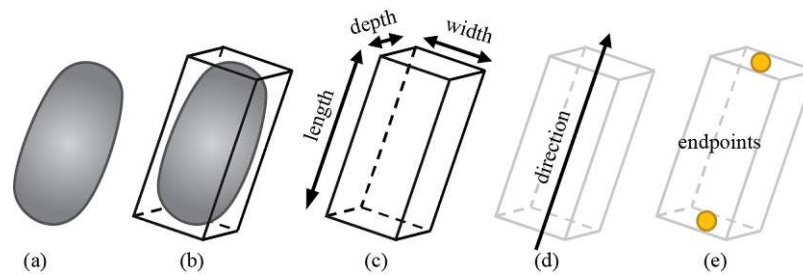
Figure 5.4: Characteristics important for the defect classification: the original defect shape (a); object oriented bounding box (b); length, width, and depth (c); direction (d); endpoints (e).

C3: The ratio between the width of a debonding and its depth should be in the range [0.5, 2].

C4: The ratio between the length of a debonding and its width should be larger than 4.

Fiber pull-out:

C5: The angle between the fiber pull-out direction and the force direction is less than 25°.

C6: In close proximity of a *fiber pull-out* there has to be at least one fiber end. The angle between the defect direction and the fiber direction has to be less than 10°.

Fiber fracture:

C7: In close proximity of a *fiber fracture* there have to be at least two fiber ends. The defect has to be in between these fibers.

A classification is assigned to a potential defect if the respective conditions are satisfied. The connected component ID of the defect is then added to the defect-type list. There are four lists in total. It is possible that a defect is assigned to more than one classification, in which case the defect ID is stored in multiple lists.

5.6 Visualization and Analysis

The visual analysis system integrates a set of techniques for investigating the 4DCT data of interrupted *in situ* tests. The techniques can be used separately or combined in a single 3D scene. To initiate a visualization method, the user specifies the required files, which are stored in a resource manager. The resource manager keeps links to all files involved in the analysis and loads them automatically on demand. Each visualization method can then be added or removed from the 3D scene setup, which allows the user to build visualizations tailored to the current analysis goal. A goal could be, for example,
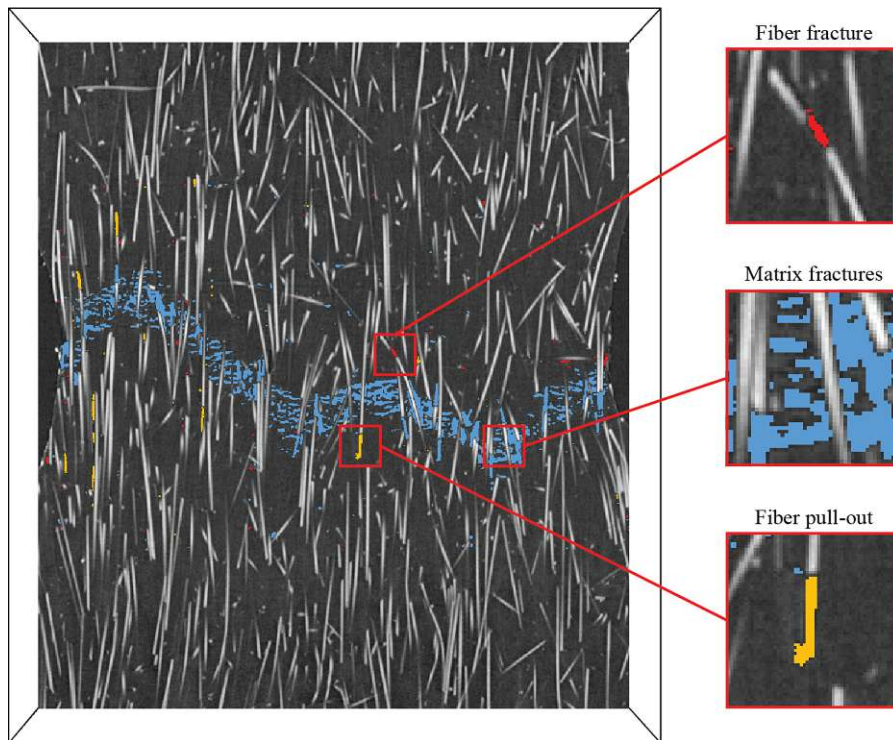
Figure 5.5: The Defect Viewer shows the original image of Dataset 1 as context with highlighted fiber pull-outs, matrix fractures, and fiber fractures.

comparing different stages of the tensile test, comparing distributions of defects by type or by region, getting an overview or detailed visualization of defects, and interactive exploration.

5.6.1 Defect Viewer

The Defect Viewer displays a 3DCT dataset in 2D slices. Defects in these slices are highlighted according to their type (see Figure 5.5). The user can change the slice position and orientation. To perform defect highlighting, a labeled image and a list of defects are required. User-specified colors are assigned to defects of the different types. The image containing the 2D slice with defect highlighting is computed on-the-fly. The Defect Viewer can provide the context in case it is combined with other visualizations.

5.6.2 Defect Density Maps

One of the features of the visual analysis system is calculating and visualizing Defect Density Maps. Defect Density Maps can be calculated for particular defect types or for all defects. They provide an overview on the distribution of defects by type and help to visually determine regions with the highest number of defect occurrences. A general

(a) Fiber pull-outs

(b) Matrix fractures

(c) Fiber fractures

(d) Fiber/matrix debondings
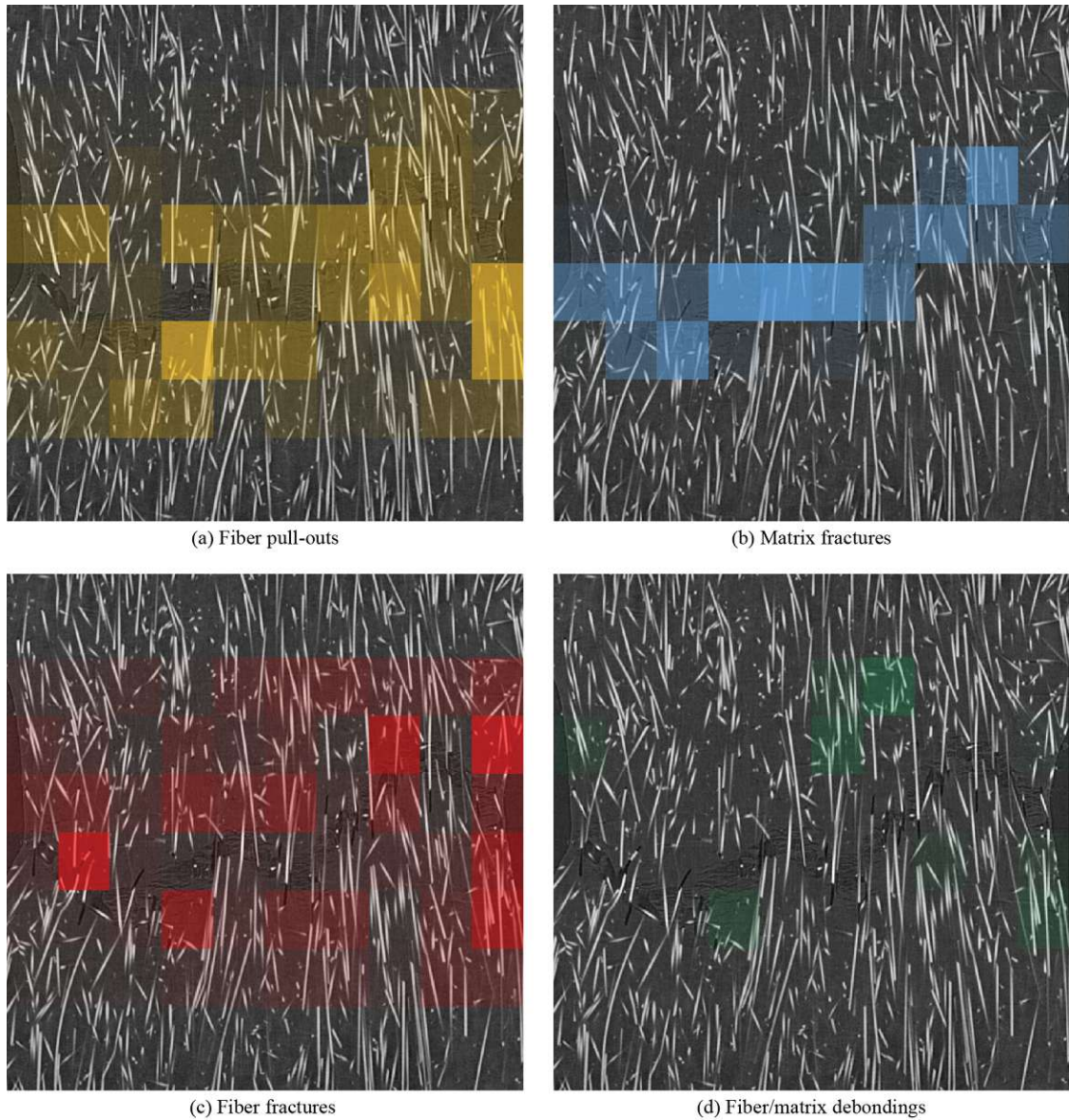
Figure 5.6: Four Defect Density Maps displaying distributions of fiber pull-outs (a), matrix fractures (b), fiber fractures (c), and fiber/matrix debondings (d) in Dataset 1.

overview of the material is provided, which enables decision making based on defect-density and fiber-characteristics distributions in the areas or interest. Figure 5.6 illustrates Defect Density Maps of fiber pull-outs (a), matrix fractures (b), fiber fractures (c), and fiber/matrix debondings (d). The distribution of defects depends on the defect type. Matrix fractures mostly occur in the region of the final fracture, while fiber pull-outs and fiber fractures occur close to the final fracture region. Material engineers use density maps
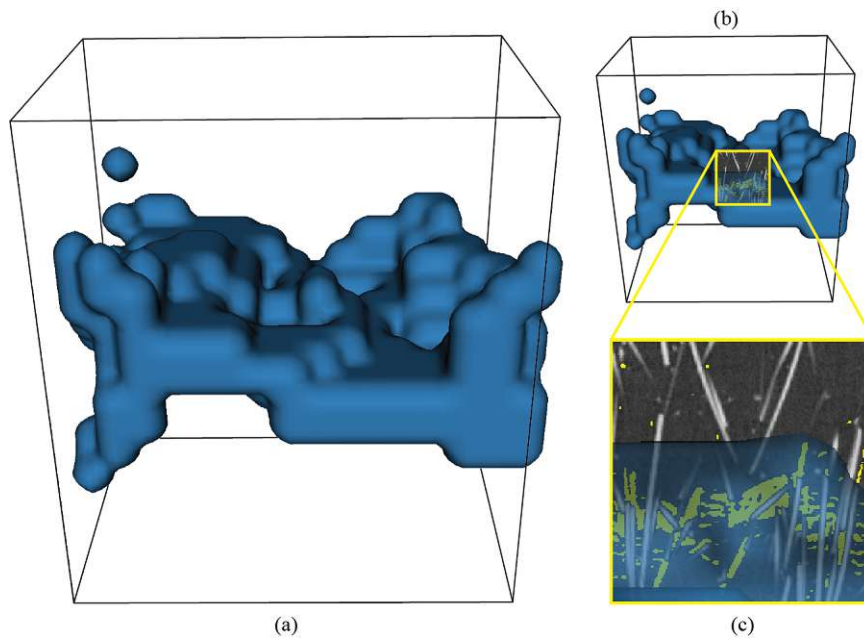
Figure 5.7: The Defect Density Map isosurface outlines the high-defect density region in 3D for Dataset 1 (a). A slice with defects highlighted in yellow is shown using the 3D Magic Lens (b). A zoom-in into the region of interest demonstrates that the isosurface efficiently separates regions containing a high number of defects (c).

to correlate fiber characteristics with defect occurrences, e.g., correlate fiber pull-outs to fiber orientations. Defect Density Maps can be visualized in 2D slices or directly in the 3D view using smooth surfaces.

Defect Density Maps calculation: A Defect Density Map is generated from the labeled image and the list of defects. From the labeled image, a mask of defects is extracted. The mask is then divided into subregions. For each subregion, the number of voxels belonging to defects is calculated and stored in a Defect Density Map.

2D visualization: A color transfer function assigned to the density map allows users to identify areas with a specific rate of defect occurrence. Finally, the density map is superimposed on the original CT data in the Defect Viewer, using an opacity transfer function, where opacity increases proportionally to density.

3D visualization: Defect Density Maps in 3D can be represented by smooth semi-transparent isosurfaces with contours around areas of interest (areas with high defect densities) (see Figure 5.7). Alternatively, direct volume rendering using color and opacity transfer functions can be employed (for example, see Figure 5.8). The isosurfaces are calculated from the Defect Density Map by a marching-cubes surface-extraction applying a user-defined threshold. As a result, the surfaces are calculated for regions where the defect density is higher than the threshold. Gaussian blurring can be applied to smooth

Figure 5.8: A series of Defect Density Maps showing the defect development process from the first to the last stage for Dataset 3. Direct volume rendering is used. The color transfer function is shown at the top.

the region surfaces. Multiple surfaces can be extracted for any combination of tensile-test stage and defect type. Colors of the su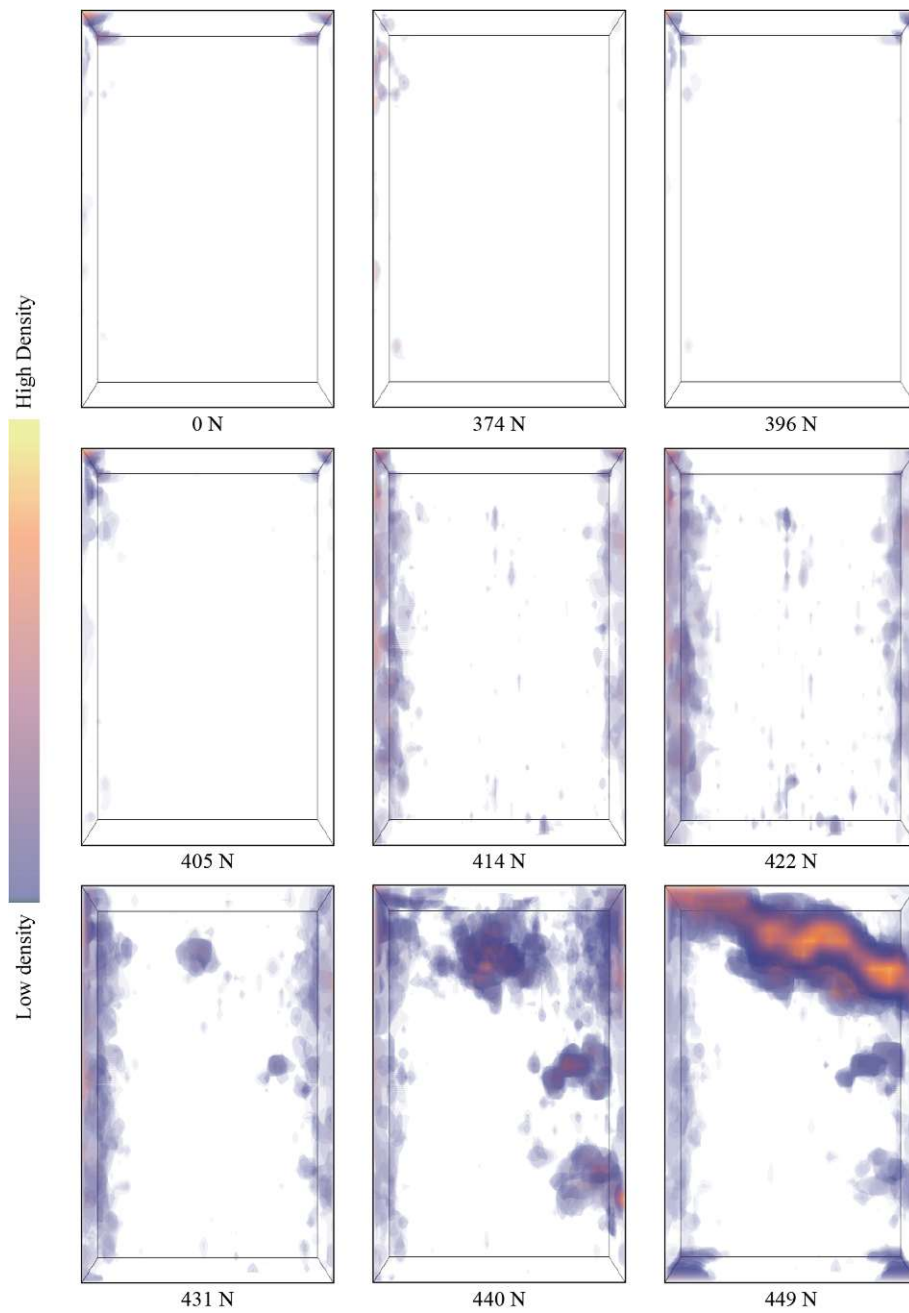rfaces are configurable. The 3D Defect Density Maps can be used for comparing defect distributions of different types in one stage or for comparing distributions of a given defect type for several or all tensile-test stages (see Figure 5.7 and Figure 5.9). This visualization facilitates the exploration of the defect-development process in the material as the load increases.

5.6.3   Final Fracture Surface

The shape of the final fracture provides relevant information about a specimen's mechanical properties. The final fracture always occurs in the weakest region of the specimen. The fracture shape, therefore, serves as an important indicator of the material properties. The main challenge in the analysis of the final fracture is to properly define its shape. Figure 5.10 shows the final fractures for Datasets 1, 2, and 3. The final fracture first occurs at a force of 244 newton. At a force of 260 newton, the final fracture widens while preserving shape.

Calculation: The fracture surface is constructed such that it runs through the middle of the final fracture in the direction of the applied force (see Figure 5.11). The fracture-surface extraction is performed as follows: Rays are cast along the force direction (usually from top to bottom) through the segmentation mask of a fracture. One ray is cast for each column of voxels. The ray starts at the center of a top voxel and ends at the center of the corresponding bottom voxel. For each ray the average position of all fracture voxels along the ray is then determined (see Figure 5.11). Finally, for all rays these positions correspond to points on the fracture surface. A Gaussian filter can then be used to reduce noise and smooth the surface, which can be then stored as a height map.

5.6.4   3D Magic Lens

To combine different visualization techniques, we implemented a 3D Magic Lens. The 3D Magic Lens is an interactive tool that changes the object appearance in a selected region (see Figure 5.9). Possible combinations of available visualization methods and test stages can be chosen for the selected region and for the surrounding area. This is achieved by rendering 3D scenes twice from the same camera position and then displaying a part of one image on top of the other one. The 3D Magic Lens has a rectangular shape and the user can change its position and size. We assume the following usage scenarios of the 3D Magic Lens:

- Analyzing and comparing differences between tensile test stages. For instance, it is possible to use an early stage of the test for providing the context, while studying defects at a later stage within the region of interest. The usage example is given in Figure 5.9a.
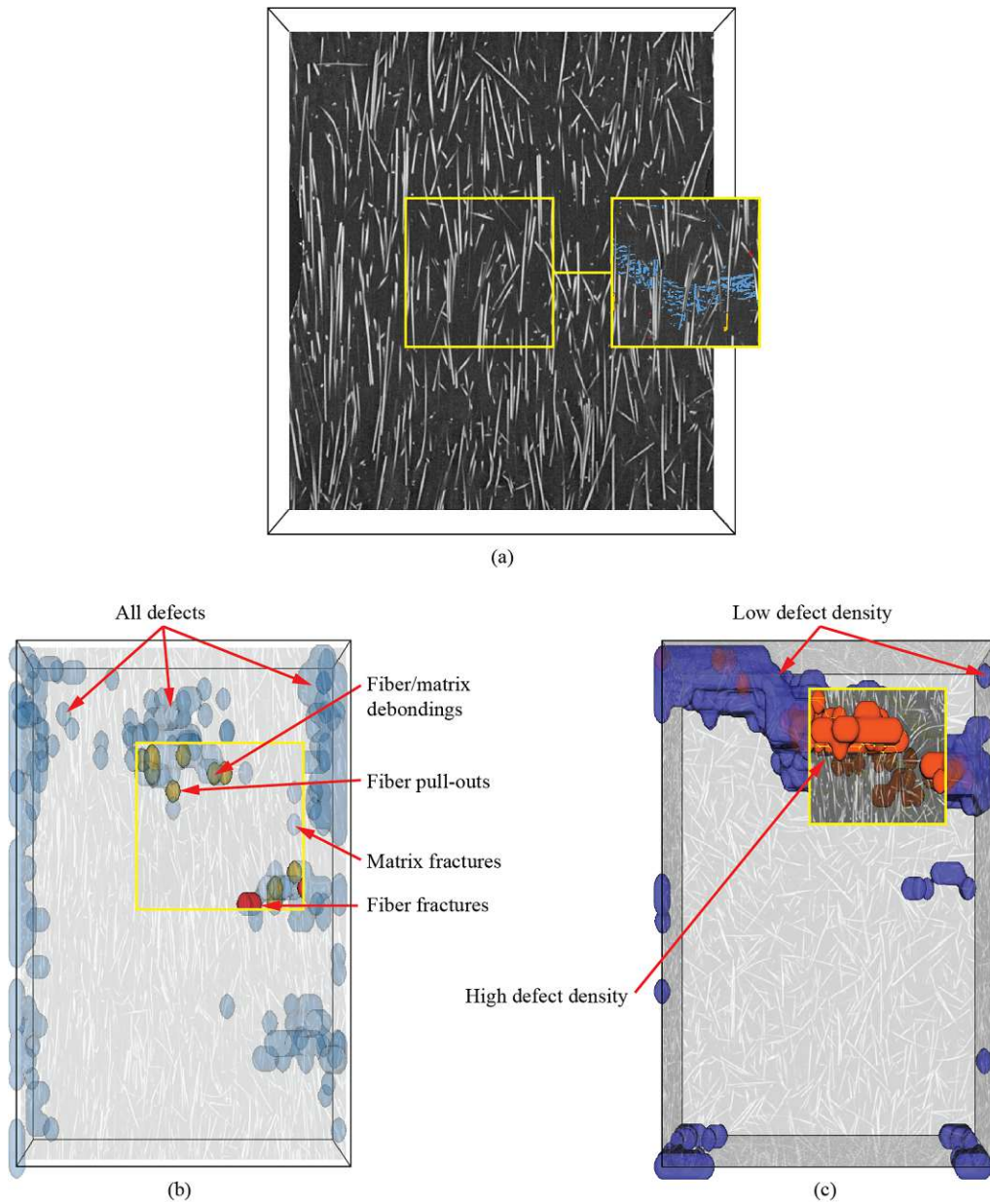
Figure 5.9: The usage scenarios of the 3D Magic Lens: (a) Comparing two stages (244 newton and 260 newton) of Dataset 2 in the side-by-side mode. Defects are highlighted for the 260 newton stage. (b) Comparing a Defect Density Map of all defects to the Density Maps for matrix fractures (blue), fiber/matrix debondings (green), fiber pull-outs (yellow), and fiber fractures (red). The 440 newton stage of Dataset 3 is shown. (c) Exploring the Density Map isosurface with high density (red) together with the 2D slice of CT data in the context of an isosurface with low density (blue).
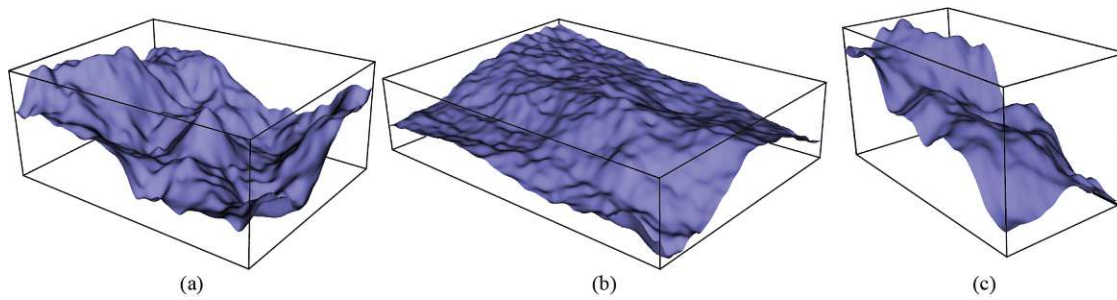
Figure 5.10: The Final Fracture Surfaces show differences between the final fractures of Datasets 1 (a), 2 (b), and 3 (c). Dataset 2 has a fracture surface that is flatter than the ones for Datasets 2 and 3.

- Applying a specific combination of visualization techniques in the region of interest. For example, Figure 5.9a demonstrates that with the 3D Magic Lens the user can interactively enable defect highlighting with the Defect Viewer in a small region. In Figure 5.9b the Defect Density Map of all defects provides the context, while the Density Maps of individual defects are shown in the region of interest.

- Looking through the occluding visualizations such as Density Maps by disabling them in the region of interest. This enables an interactive visual analysis, where the overview is provided with one set of techniques, and details are shown with other set of visualizations. The usage example is given in Figure 5.9c.

The 3D Magic Lens visualization can be used in two modes: a side-by-side mode (Figure 5.9a), and a conventional magic lens mode (Figure 5.9b,c).

5.7   Evaluation and Results

From the beginning of this work, we closely collaborated with material engineers from the industrial company Borealis AG. The company is focused on the production of polyolefins. During our collaboration, Borealis AG provided us with test specimens, which are made of the Fibremod GD301 material. Our goal was to create approaches for analyzing 4DCT data and to develop a tool, which implements these approaches. The tool was developed for and is mainly used by material engineers of FHOOE (University of Applied Sciences Upper Austria). At the beginning of the project, we had several meetings with material engineers. During these meetings we identified and specified the analysis tasks given in Section 5.2. We then aimed to support the tasks with our approaches. We regularly had internal meetings with the material engineers of FHOOE. During these meetings, we discussed new ideas and refined existing ones. We started from initial ideas and iteratively improved and developed them while staying in a close feedback loop with the users. If we required additional input, we contacted Borealis AG and discussed the open topics. For example, we learned that all the four defect types are interesting to
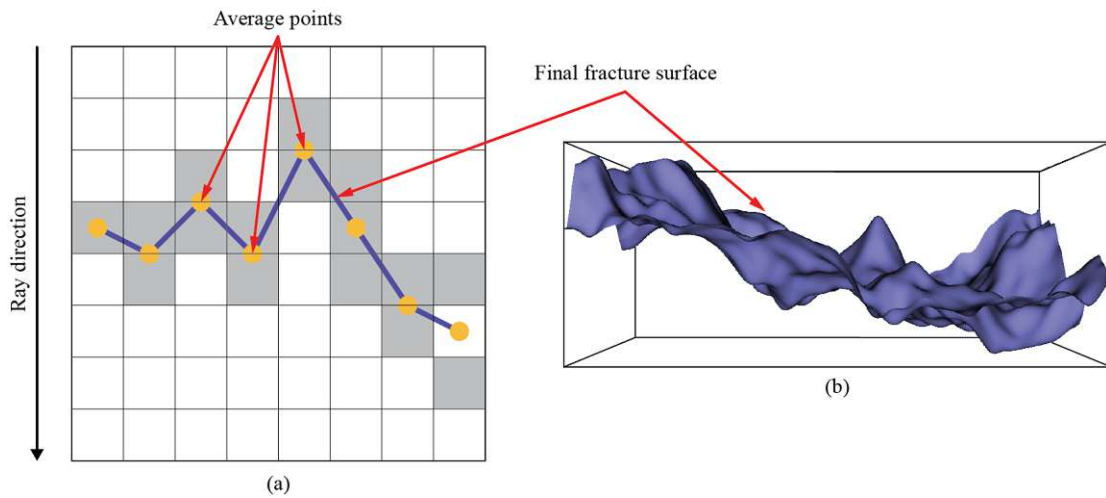
Figure 5.11: To calculate the Final Fracture Surface rays are cast from top to bottom. An average position of defects is calculated for each ray (a). The resulting height-map is then smoothed and reconstructed into a 3D mesh (b).

the material engineers and there are no types, which we could ignore. In addition, we discussed where the transition limits between defects types are (e.g., when a fiber pull-out becomes a matrix fracture). Together with partners from Borealis AG, we frequently assessed the correctness of our assumptions. In addition, we regularly demonstrated our approaches and results. We got a positive feedback concerning results of the Defect Classifier tool. Material engineers at Borealis AG were very interested in the automatic defect classification and found the tool very useful concerning the description of the fracture mechanism. The utility of the Density Viewer tool was highly estimated as well. In the Defect Density Maps visualization, they appreciated that it highlights regions of high density and enables finding correlations between them and internal material characteristics. In the 3D version of the Density Maps they liked that it can provide a description of the material destruction process with a visualization technique that is easy to interpret. Material engineers also appreciated the ability to quickly compare different 4DCT datasets as well as different stages of a single tensile test with each other.

Additionally, we interviewed a material engineer at FHOOE who is one of the main users of our software. We asked about the software usability and the usefulness of the developed tools, and whether they met the defined goals. In general, the user appreciates that the approach with a single 3D scene is interesting and agile. Regarding the individual visualization tools, we have got the following feedback:

Defect Classifier: Similar as with the material engineers from Borealis AG, we got a positive feedback about the tool. The interviewee thought that the provided defect extraction and classification provides important information for the 4DCT analysis.

112

Defect Viewer: The material engineer's opinion is that the Defect Viewer's concept fits well into the used visualization toolbox. The tool integrates well into the 3D scene and is well suited for combining and interacting with other components of the toolbox.

Defect Density Maps: The value of this tool was evaluated as being high. The material engineer felt that Defect Density Maps help in gaining important information about the defects' distributions as well as finding a correspondence between internal characteristics and the damage (final fracture shape and the number of defects close to it).

Final Fracture Surface: The material engineer appreciated "a nice overview of the fracture" that this tool provides. He also mentioned that the possibility of measuring the distance between the highest and the lowest points would be of use to him.

3D Magic Lens: The tool was highly commended due to its flexibility and ability to supplement many other tools in solving a number of tasks. The interviewee appreciated that the interactive comparative visualizations provided by the 3D Magic Lens allowed him to observe and analyze the fracture evolution.

5.8   Use Cases

Use case 1: Using the Final Fracture Surface a material engineer compared fractures of specimens with different major fiber orientations, i.e., 0° orientation (Datasets 1 and 3) and 90° orientation (Dataset 2). Before the fracture extraction, material engineers assumed that the specimen with the 90° fiber orientation will have the flattest fracture surface of all three. The visualization results provided by the Final Fracture Surface confirmed this assumption (see Figure 5.11). The results show that the 0° fiber orientation fracture has higher peaks, more variance, and a larger distance between the maximum and minimum points.

Use case 2: To evaluate defect distributions the users have built density maps for defects of each type (see Figure 5.6). The visualizations clearly show that matrix fractures are located mostly in the region of the final fracture while fiber pull-outs and fiber fractures are distributed in a wider region. However, a more detailed analysis using the Defect Viewer and the 3D Magic Lens revealed that all defects contributed to the development of the final fracture. The fracture developed in the areas with the highest densities of matrix fractures, while several fibers adjacent to the final fracture had fiber fractures and fiber pull-outs on both ends. As fiber pull-outs often occur at the fiber ends that point away from the final fracture, fiber pull-outs are characterized by a wider distribution.

Use case 3: First, to get an overview, the user compares density maps of all defect types for all the stages of the tensile test (Figure 5.8). It can be seen that the 440 newton stage is of high importance: defects accumulate critical mass before the final crack is formed. The user then chooses to investigate this stage in more detail employing the 3D Magic Lens by visualizing distributions of individual defects in the context of the total defect density (Figure 5.9b). It can be seen that there are many fiber/matrix debondings and matrix fractures in the center of the top area. Therefore, this is the weakest region,

where the final crack will appear. Finally, the user applies the 3D Magic Lens showing areas with the highest defect density in the context of the CT data to study individual defects in the weakest region after the final crack has formed (Figure 5.9c).

5.9   Discussion of Design Choices

The aim of this work is to provide material engineers with methods for 4DCT visualization and analysis. We organized the design and development process in the following way. First, together with material engineers, we formalized their requirements and needs into a set of tasks (provided in Section 5.2). Then, in a series of discussions, we defined the methods that would be prototyped. Each idea for a new method was ranked based on two criteria:

- the ability of the suggested method to solve the corresponding analysis tasks,

- and the cost of implementing the method.

Ideas with the highest rankings were then prototyped. Finally, the prototypes were presented to the material engineers. Using their feedback, we picked a set of methods for the final implementation.

We chose to implement the Defect Viewer and the Defect Density Maps because they have relatively low implementation complexities and were considered sufficient for solving tasks of the material engineers. For the Final Fracture Surface and the 3D Magic Lens, a few alternatives were discussed.

Final Fracture Surface: At early prototyping stages, we considered the alternative of representing the final fracture as a 3D volume. When implementing this approach we encountered two main issues:

- After the fracture appears, its thickness keeps changing in the following steps but the fracture surface shape remains the same. This hinders the comparative analysis between late tensile-test stages.

- Since defects could occur in the whole volume of the specimen, it is not clear where the border of the final fracture should be defined. To resolve this issue, the density parameter could be used, which leads to an additional complexity of the final fracture extraction algorithm. In the end, it would make the comparison of two fractures a non-trivial task.
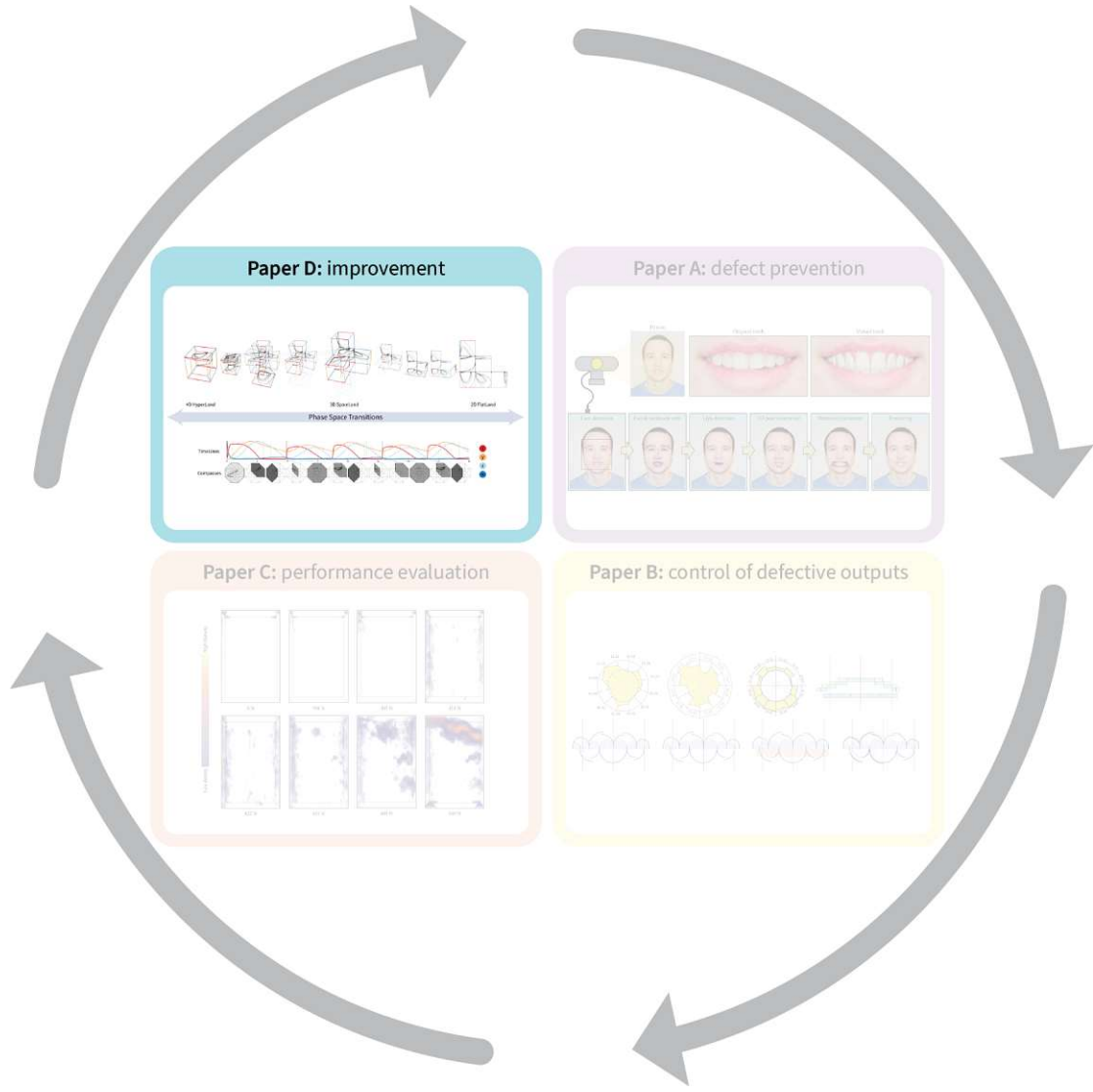
Based on the above considerations, the approach of representing the final fracture as a surface in 3D proved to be more suitable for our analysis tasks (see Figure 5.10).

3D Magic Lens: As alternative comparative visualization methods, we considered side-by-side and checkerboard image displays. For the latter, the checkerboard pattern is used where black squares show one image and white squares show the other image. We chose the 3D Magic Lens because of the following reasons:

- The interactive functions of the 3D Magic Lens were rated as highly useful by the material engineers.

- The resulting visualization was considered easier to interpret.

- Due to its local nature, the 3D Magic Lens enabled the user to focus only on important changes and only on the regions where they are relevant.

- When the checkerboard visualization was used, borders between two squares could go through an important feature, like defect or fiber. The 3D Magic Lens visualization avoids such cases.

5.10   Conclusions and Future Work

We have presented a workflow for the visual analysis of 4DCT data. The workflow comprises two parts: 1) defect extraction and classification, and 2) visual analysis. A novel defect classification method based on commonly accepted domain-specific criteria is introduced. Four approaches for the interactive visual analysis of 4DCT data are presented and evaluated, i.e., Defect Viewer, Defect Density Maps, Final Fracture Surface, and 3D Magic Lens. The methods complement each other and can be combined for more flexible visualizations tailored to specific analysis tasks. The presented methods were continuously evaluated in close feedback loops with industry and university material engineers, all of whom work with GFRPs on a daily basis. Further evaluation interviews resulted in positive feedback and confirm the importance of these methods in the area of 4DCT tensile tests. This work opens several exciting directions for future work. The first direction encompasses applying presented visualization tools to other materials. Carbon fiber reinforced polymers and natural fiber reinforced polymers are very promising and relevant for the industry. The second direction includes finding the correlation between fractures and material properties, as is of particular interest to the industry. An extensive base of defects should be compiled and used to predict fractures in new materials. The third direction for future work involves investigating methods for a non-rigid registration or morphing of the interrupted *in situ* tensile test stages. This would allow material engineers to navigate between the stages and visualize the continuous development of defects and fractures using animations or highlight the regions where the most significant deformation occurs.

Paper D: improvement

Paper A: defect prevention

Paper C: performance evaluation

Paper B: control of defective outputs

CHAPTER 6

Improvement

This chapter addresses the improvement (i.e., the *act* phase of the PDCA cycle) and is based on the publication:

Mathematical models of ordinary differential equations are used to describe and understand biological phenomena. These models are dynamical systems that often describe the time evolution of more than three variables, i.e., their dynamics take place in a multi-dimensional space, called the *phase space*. Currently, mathematical domain scientists use plots of typical trajectories in the phase space to analyze the qualitative behavior of dynamical systems. These plots are called phase portraits and they perform well for 2D and 3D dynamical systems. However, for 4D, the visual exploration of trajectories becomes challenging, as simple subspace juxtaposition is not sufficient. We propose *ManyLands* to support mathematical domain scientists in analyzing 4D models of biological systems. By describing the subspaces as *Lands*, we accompany domain scientists along a continuous journey through 4D *HyperLand*, 3D *SpaceLand*, and 2D *FlatLand*, using seamless transitions. The *Lands* are also linked to 1D *TimeLines*. We offer an additional dissected view of trajectories that relies on small-multiple compass-alike pictograms for easy navigation across subspaces and trajectory segments of interest. We show three use cases of 4D dynamical systems from cell biology and biochemistry. An informal evaluation with mathematical experts confirmed that *ManyLands* helps them to visualize and analyze complex 4D dynamics, while facilitating mathematical experiments and simulations.

117
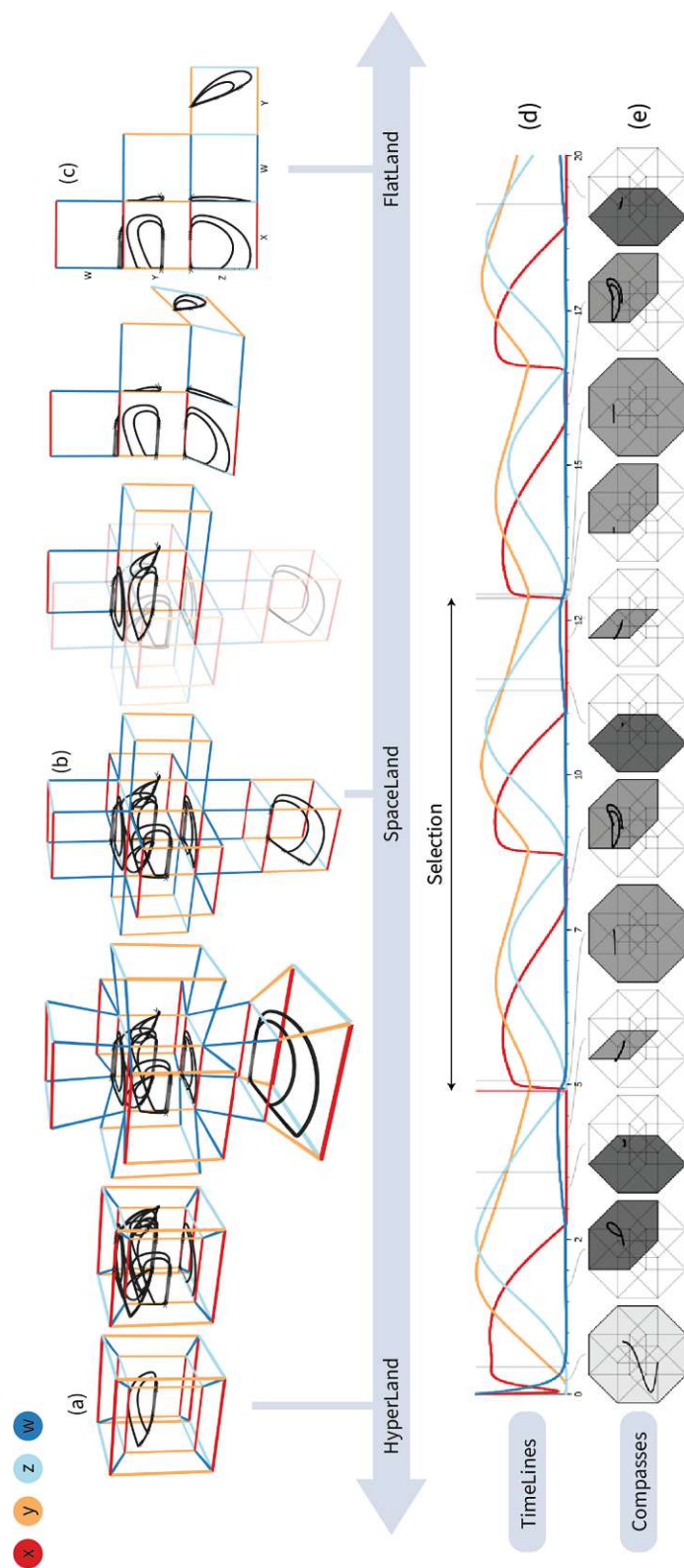
Figure 6.1: *ManyLands*: a visualization application to support the exploration and analysis of 4D dynamical systems trajectories. The capabilities of *ManyLands* are based on interaction and smooth, animated navigation through phase space. We use the metaphor of traveling across (a) 4D *HyperLand*, (b) 3D *SpaceLand*, and (c) 2D *FlatLand*. Additionally, we employ (d) *TimeLines* to represent each variable of the dynamical system across time, and (e) *compasses* to facilitate the analysis of segments of the trajectories.
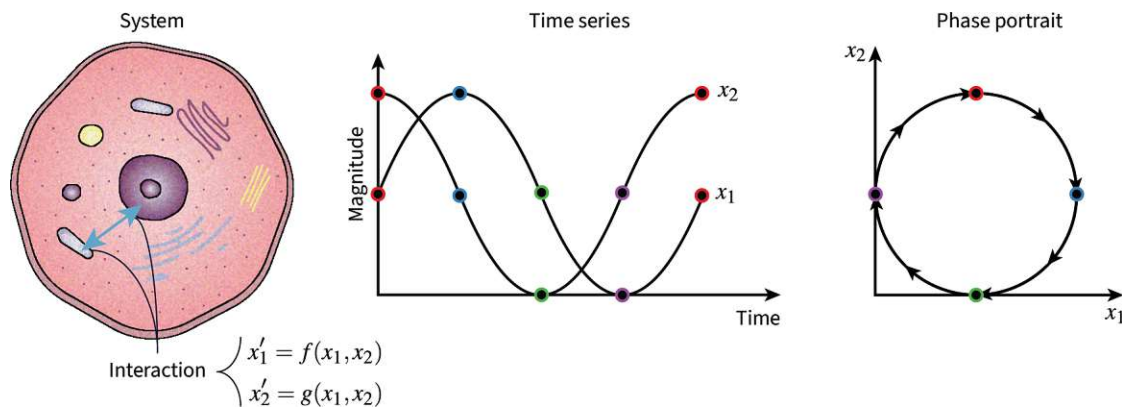
Figure 6.2: Illustration of a 2D ODE system of a cellular interaction, the time series of its variables (x_1, x_2) and its phase portrait.

6.1 Introduction

During the last decades, mathematical modeling and simulations have emerged as important resources to gain a better understanding of the function and dynamics of complex biological and biochemical processes. Guided by experiments, mathematical models of biological processes are formalized as large systems of *ordinary differential equations (ODEs)* describing the time evolution of biochemical species [168]. Important examples include dynamical systems describing glycolytic oscillations [170], i.e., a repetitive biochemical fluctuation in the concentration of metabolites, mathematical models for the cell division cycle [172], and models for the explanation of mechanisms behind bipolar disorders [124].

The exploration and analysis of the behavior of these systems by mathematical domain scientists is anticipated to generate new knowledge on the underlying biological phenomena. This is especially significant for the development of new models that describe biological processes more accurately. Typical and pertinent mathematical questions of biological interest are the existence and stability of *equilibria, periodic oscillations, and switching phenomena* in the behavior of the investigated systems.

Temporal patterns and mathematical behaviors are analyzed using *dynamical systems theory*. Approaches within this area focus on a qualitative analysis of the behavior of dynamical systems, instead of seeking explicit computations of solutions. Geometric techniques like *phase space analysis* study the flow of dynamical systems and the geometry of the phase space, which depicts biologically meaningful solutions, or *trajectories* [71,128]. Within the current workflow, *phase portraits*, i.e., representations of typical trajectories in phase space, are employed for 2D and 3D systems. A 2D phase portrait is shown in Figure 6.2.
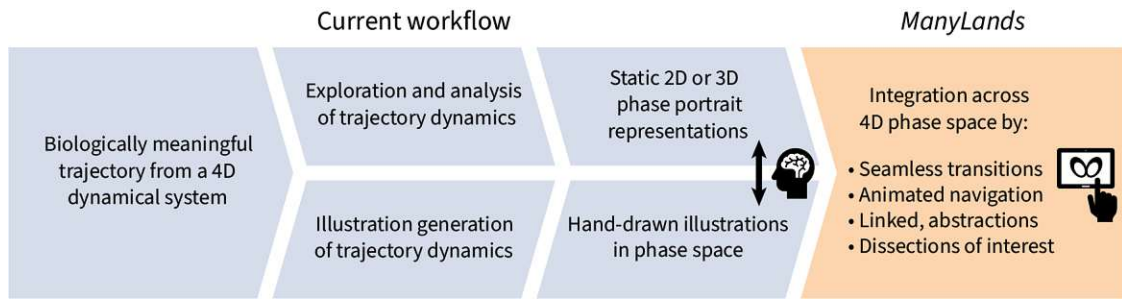
Figure 6.3: Comparison of the current analysis workflow of 4D dynamical system trajectories vs. *ManyLands*. The current workflow (blue) requires mental linking between its components, whereas *ManyLands* (orange) provides mathematical domain scientists with an integrated solution, based on seamless transitions and animated navigation.

Many biological processes are modeled by 4D dynamical systems [176]. For 4D systems, phase portrait analysis becomes challenging. The current technology adopted by mathematical domain experts involves generating and analyzing static 2D or 3D phase portraits—subspace projections of 4D dynamical system trajectories. This is done, e.g., in *MATLAB* [189], or *DYNAMICA* within *Wolfram Mathematica* [177]. The juxtaposition of the 2D or 3D phase portraits provides limited insight, as certain complex patterns can be revealed only in the 4D space. Besides, mathematicians sometimes generate their own hand-drawn illustrations to compensate for the limitations of the available technology, to navigate across subspaces, to demonstrate interesting phenomena in their systems, and to externalize their findings and proofs [93, 171]. To link their hand-drawn illustrations to computer-generated 2D or 3D phase portraits and to reconstruct the 4D space, the experts rely on spatial imagination. The current workflow is depicted in Figure 6.3.

The *contribution* of this chapter is the design and implementation of *ManyLands*. It is a web-based application that supports mathematical domain scientists in understanding and analyzing the mathematical behavior of biologically meaningful 4D dynamical system trajectories. It allows them to discover new knowledge within their systems and to illustrate the findings for externalization and education purposes. *ManyLands* offers an interactive and integrated workflow for exploring and analyzing 4D dynamical systems trajectories, as shown in Figure 5.2 in the rightmost column.

6.2   Background of 4D Biological Dynamical Systems

A *dynamical system* is characterized by the relationship between state variables and their time derivatives, i.e., it can be represented as a system of ODEs given by:

$$x' = f(x),\ x(0) = x_0,\ x \in \mathbb{R}^n. \tag{6.1}$$

This allows from a given initial state x_0 of the system to compute the future (and past) states according to the dynamical law f [71, 128]. Since it is usually impossible to derive an explicit formula for the solution of a nonlinear equation, *dynamical systems theory*

has proven to be a powerful resource for understanding *qualitative features* of trajectories through *phase space analysis*. Examples of dynamical systems are shown in Section 6.6.1, and their mathematical descriptions are provided in our repository [34].

Each of the n independent variables of the system defines a coordinate axis in the corresponding n-dimensional *phase space*. In the context of biological systems, we focus on 4D dynamical systems and their 4D phase space, while the variables x in Equation 6.1 represent molecular concentrations (or numbers) within a cell, i.e., concentrations of enzymes or proteins. The *dynamics* of the system represent changes in these concentrations. If $x(t)$ is a solution of the system, $x(t_0)$ defines at each time t_0 a point in the phase space. As the point changes with time, the entire solution $x(t)$ traces out a curve, or *trajectory* in the phase space, as depicted in Figure 6.2. By this, solutions correspond to geometric objects in phase space. For instance, resting states (when variables do not change) correspond to equilibria points, and oscillations to so-called limit cycle curves. Nowadays, numerical computations of solutions are commonly used to analyze how the system state evolves over time. In particular, numerical simulations of solutions are used as a tool to guide a theoretical phase space analysis.

Although all solutions could be mathematically interesting, there are several biological restrictions. For instance, there are no negative concentrations. Therefore, *only few, biologically meaningful trajectories* are explored and analyzed at the end. Observing the geometry of these trajectories in phase space using *phase portraits* allows mathematical domain scientists to understand important dynamics of their systems, and potentially, of the underlying modeled processes [71, 128]. Examples of dynamics arising in biology, i.e., changes in concentrations, include *switch-like behaviors* and *periodic oscillations*. For instance, the activity of a protein is "off" (inactive), i.e., the corresponding variable has a value that equals zero (or close to zero)—otherwise, it is "on" (active).

6.3 Domain Requirements and Tasks

In Section 6.1, we discussed the current workflow of mathematical domain scientists and its limitations regarding the analysis of 4D dynamical systems. Mathematical domain scientists need strategies for completing the following tasks:

(T1) Holistic Analysis of 4D Trajectory Dynamics: As several phenomena can be only observed in the entire 4D phase space, domain scientists need to: *(i)* have an overview of the entire space; *(ii)* detect alternations in trajectory behaviors across subspaces; *(iii)* explore and analyze where, how, and why these occur; *(iv)* drill down to subspaces. Instead of using static 2D or 3D phase portraits, they require an integrated approach to *navigate and transit across the entire dimensionality of the phase space* **(T1.a)**. Additional *features*, e.g., slow vs. fast behavior, need to be discovered and highlighted for further exploration **(T1.b)**.

121

(T2) Dissected Analysis of 4D Trajectory Dynamics: Several phenomena can only be observed in distinct subspaces of the trajectories or in smaller segments thereof. As the behavior of the trajectories is not always known a priori, domain scientists require a mechanism to drill down interactively to a *dissected analysis view* of the phase space that localizes segments of interest **(T2.a)**. This needs to be summarized in a *comprehensive representation* **(T2.b)**, where *navigation* is possible **(T2.c)**.

(T3) Integration of Knowledge Discovery with the Generation of Phase Space Illustrations and Animations: In the current workflow, mathematical domain experts often use manual illustrations for investigation, education, or externalization purposes. To reduce the manual and mental effort of domain scientists, functionality for *meaningful illustrations and animations* needs to be integrated with the interactive trajectory computation and representation.

6.4 Related Work

Several previous works have addressed the exploration of dynamical systems [20, 126, 277] or objects with a dimensionality higher than three [136]. All these approaches propose to employ multiple subspace projections or additional encodings in the 3D space to incorporate the fourth dimension. For more than three dimensions, parallel coordinates [152] have been employed by Wegenkittl et al. [277] and Grottel [127]. None of these approaches can fully provide the required functionality of a combined holistic and dissected exploration and analysis of the 4D phase space, as well as the ability to integrate knowledge discovery with the generation of illustrations and animations.

To tackle a dimensionality higher than 3D, previous work has been conducted also concerning discrete data—*rolling the dice* [99] being the most relevant one. It is an interactive method for the exploration of multidimensional data through queries. It is based on a scatterplot matrix that provides an overview, while supporting interactive navigation in the multidimensional space by animated transitions. Animated transitions have been investigated in several other works for the exploration of discrete data [111, 237, 269]. Still, our focus is on strategies for transiting between continuous data—in particular, approaches that employ *slicing* through high-dimensional data, approaches that employ *projections*, and *hybrids* thereof.

HyperSlice [267] is the first method to depict multiple two-dimensional slices in a trellis plot for the visualization of high-dimensional functions. This is done around a point of interest, common to all 2D slices. The approach is local in nature—similar to other multidimensional strategies, such as the *grand tour* [38]. It requires to repeatedly probe the function, in order to simulate a global view on the data. *HyperMoVal* [219] is another slicing approach, which is designed with a strong focus on modeling validation. This is done with 2D or 3D projections of the involved high-dimensional scalar functions around a point of interest. In all cases, only a localized view is provided.

As opposed to slicing approaches, several methods provide a global view on the functions, such as *continuous scatterplots* [41], *continuous parallel coordinates* [140], or *profile contour plots* [134]. Here, density fields are mapped to either 2D or 3D representations. However, in our case, we are interested in the function itself—not in density fields. Other global views include focus+context approaches like *PolarEyez* [156]. Reduction techniques, such as the *HyperCell* [92], take lower-dimensional representations as basis for the visualization of high-dimensional spaces. In *HyperCell*, the user determines 1D to 3D cells, i.e., subspaces, where operations like switching, rotating, and brushing are possible. In the *Worlds Within Worlds* approach [105], nested heterogeneous coordinate systems of lower dimensions enable to view and manipulate high-dimensional functions in arbitrary, nested, interactive boxes. Another projection-based method is proposed by Nouanesengsy et al. [210], where the high-dimensional space is projected onto 2D plots, showing changes of the different parameters. In all these cases, selected projections with reduced dimensionality are offered as static views to the user. An overview on the entire dimensionality, and an easy transition between the represented subspaces is not possible.

To balance between local and global approaches, Torsney-Weir et al. [261] propose the *Sliceplorer*, where 1D line plots constitute the central view to show changes in a single dimension. By probing through the function and showing all slices in a projection, they obtain a hybrid that offers both global (through projections) and local (through slicing) views on the underlying data. Later, they propose *Hypersliceplorer* [259], which works with 2D slices of multi-dimensional shapes. Yet, transitions between subspaces and the generation of illustrations and animations are still not feasible. These, though, are very important in the application of biological dynamical systems for understanding the mathematical behavior and specific phenomena in the trajectories. Other hybrid categories include *exploded view diagrams* [164], while pure topological approaches [44, 66, 77, 119] are also available, but out of the scope of this work.

Concerning interaction, our work revolves around seamless transitions [139, 262] and multiple coordinated views [47, 60, 286]. Our domain experts intend to interact with 4D trajectories in phase space and visually inspect them through different views. However, our domain scientists require to perform the visual analysis in many different subspaces and their intermediate transition stages. Having multiple coordinated views only would not be sufficient, given the limited screen space. Therefore, we provide a different interaction concept based on animated navigation and seamless, continuous transitions—related to the work of Jianu et al. [157], Miao et al. [201] ,and Sorger et al. [249].

6.5 The Design of *ManyLands*

We propose *ManyLands* for the exploration and analysis of pre-computed 4D trajectories from continuous-time dynamical systems of biological interest. The concepts in *ManyLands* have been inspired by the book *Flatland: a Romance of Many Dimensions* [19]. The mathematical domain scientist—as another *Square*—visits and navigates across all *Lands*, i.e., subspaces, and their respective dimensions. We introduce three different

x y z w

HyperLand   SpaceLand   FlatLand

(a) unfolding through the Dalí cross

HyperLand   SpaceLand   FlatLand

(b) simplified unfolding
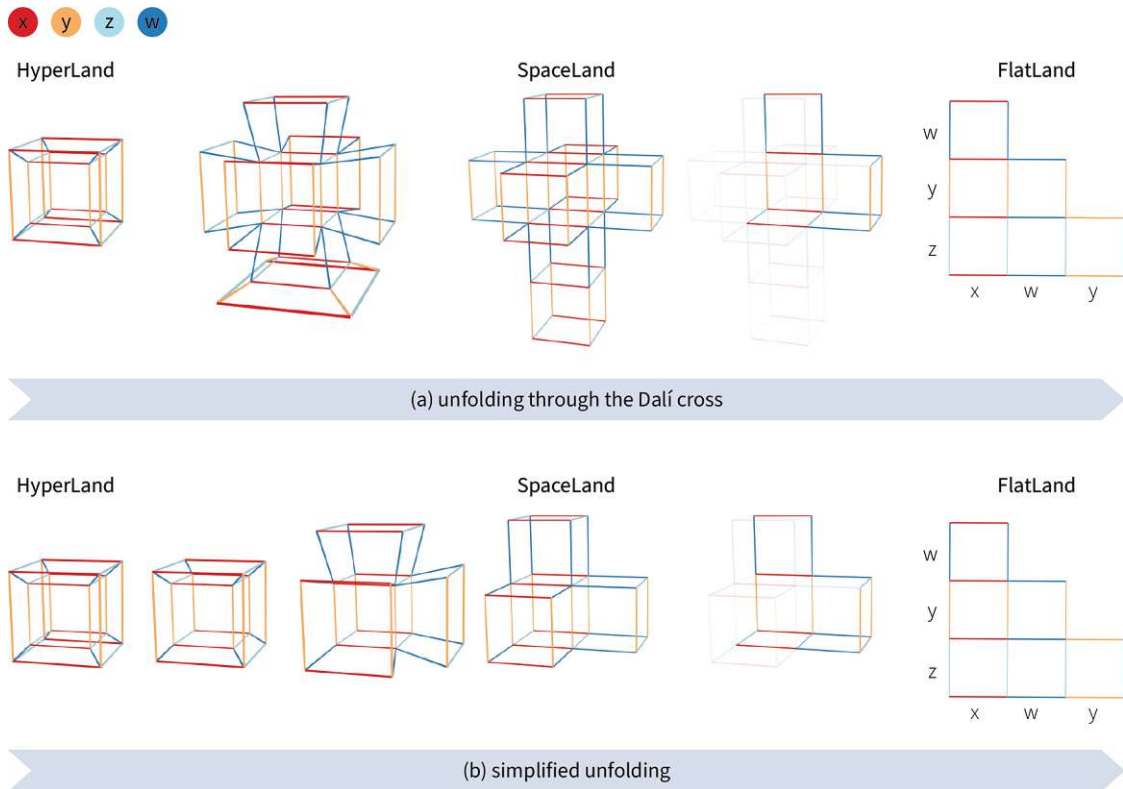
Figure 6.4: Two strategies for smooth, animated transitions between *HyperLand*, *Space-Land* and *FlatLand*. In (a) 3D symmetries are preserved, but 2D symmetries are omitted, while in (b) 3D and 2D symmetries and rotations are disregarded.

Lands: 4D *HyperLand*, 3D *SpaceLand*, and 2D *FlatLand*. We also include an additional linked diagram, called *TimeLines*. The design of *ManyLands* takes into consideration the framework proposed by Gleicher [121] regarding comprehensibility in modeling.

The initial view in *ManyLands* shows the entire 4D dimensionality of the system in *HyperLand* (Figure 6.1a). Represented as a tesseract, it shows the morphology of the 4D mathematical system in a Schlegel diagram [248]. From the 4D space, the domain expert can smoothly travel to all lower-dimensional *Lands*, which are built upon a polytope concept. A space with n dimensions ($n > 0$) is composed of a number $E_{m,n}$ of m-dimensional ($n > m$) subspaces [79], given by:

$$E_{m,n} = 2^{n-m} \binom{n}{m},\tag{6.2}$$

where $\binom{n}{m} = \frac{n!}{m!(n-m)!}$ and $n!$ denotes the factorial of n. A tesseract ($n = 4$) based on Equation 6.2 contains eight cubes ($m = 3$) and twenty-four squares ($m = 2$). When considering symmetries and rotations in space to be redundant, the decomposition of a

tesseract will result in four unique cubes and six unique squares. This decomposition of subspaces is shown in Figure 6.4, where redundancies have been omitted only for the squares in (a), and for cubes and squares in (b). From the *HyperLand* (Section 6.5.1), the domain scientists can smoothly travel to *SpaceLand*, by unfolding the tesseract into its corresponding 3D cubes (Figure 6.1b, Section 6.5.2). They can further unfold the cubes into a representation consisting of a matrix of 2D plots, called *FlatLand* (Figure 6.1c, Section 6.5.3). *TimeLines* is an additional linked diagram that shows the graphs of all the variables of the dynamical system over the time axis t (Figure 6.1d, Section 6.5.4). The navigation across *Lands* is achieved with smooth, animated transitions, motivated by the nature of the data and the tasks. *TimeLines* are linked to the *Lands* through brushing and linking [47, 60, 286] (Section 6.5.5). We show how *ManyLands* support the domain expert in their workflow (Sections 6.5.6–6.5.8).

6.5.1 *HyperLand*

Given the mathematical problems investigated by our domain scientists, we need a basis for a single-view analysis of 4D trajectories, which as a whole explores all phase space dimensions. To this end, we created *HyperLand*. In *HyperLand*, the trajectory of a 4D dynamical system is displayed inside a tesseract (Figure 6.1a). To display a 4D object on a 2D screen, this object must be first projected into a 3D space—from 4D to 3D space, and then to 2D space.

4D to 3D projection: The projection from 4D to 3D space is similar to the one from 3D to 2D space. To project a 4D object to 3D space, each vertex of the object is multiplied by the world matrix, the view matrix, and the projection matrix. Both the world matrix and the view matrix can be defined by multiplications of rotation and translation matrices. In 4D space, objects can be rotated around six planes—namely, the XY-, YZ-, XZ-, XW-, YW-, ZW-planes. For instance, the rotation in homogeneous coordinates around the XY-plane is given by the matrix:

$$R_{xy} = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) & 0 & 0 & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{6.3}$$

where α denotes the rotation angle. The rotation matrices around the other planes are defined analogously. The translation in 4D space with homogeneous coordinates is given by the matrix:

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 & dx \\ 0 & 1 & 0 & 0 & dy \\ 0 & 0 & 1 & 0 & dz \\ 0 & 0 & 0 & 1 & dw \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{6.4}$$

where dx, dy, dz, dw denote the translation distances along the x, y, z, and w axes, correspondingly. By applying a 4D projection matrix to the object, we obtain the result in the 3D space. The 4D projection is given by the matrix:

$$P_{4D} = \begin{pmatrix} \frac{n}{r} & 0 & 0 & 0 & 0 \\ 0 & \frac{n}{t} & 0 & 0 & 0 \\ 0 & 0 & \frac{n}{d} & 0 & 0 \\ 0 & 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & 0 & -1 & 0 \end{pmatrix}, \tag{6.5}$$

where n denotes the near clipping plane; f is the far clipping plane; and r, t, d are the extents of the truncated pyramid frustum (width, height, and depth).

3D to 2D projection: The object is subsequently projected into 2D, by a conventional 3D-to-2D perspective projection matrix. The 3D projection parameters do not necessarily repeat corresponding parameters in the 4D projection. For example, the 4D and 3D cameras can have different viewing angles, positions, and clipping planes.

Encoding: To encode the perspective distortion of a 4D object in the thickness of the employed primitives, we use 3D objects, such as cylinders and spheres, for rendering the scene (Figure 6.1). We assign a unique Deuteranopia-safe color to each of the axis-aligned edges of the tesseract based on their orientation in the 4D space to denote the four variables of the system (Figure 6.1). As requested by the mathematical domain scientists, the axes are labeled according to the domain convention, namely x, y, z, and w for the four variables, but the labels can be changed by the user if required. Axes scaling can be incorporated in the scene, although in most of the cases all variables have already undergone *nondimensionalization* [265] and they are dimensionless. By standardizing the proportions of the tesseract, we provide the user with a familiar depiction of the trajectories. The user can inspect the object in 4D and 3D space with decoupled rotation operations. The 3D rotations are performed in a conventional way through mouse interaction, while 4D rotations are performed using sliders in the interface.

6.5.2   *SpaceLand*

For 3D trajectory analysis, mathematical domain scientists widely use 2D and 3D phase portraits of their models. *SpaceLand* consists of three dimensions and contains 3D subspace projections from *HyperLand* (Figure 6.1b). It is formed by unfolding the bounding tesseract of *HyperLand* and the trajectory into a 3D space by a smooth transition between spaces, as it will be described in Section 6.5.5. There are two alternative representations of *SpaceLand*, between which the user can switch. In one representation, the mathematical model is shown with eight 3D cubes, containing all eight projections, i.e., subspaces, of the 4D dynamical system (Figure 6.4a), as discussed in the description of Equation 6.2. These plots are connected at their faces. All together, they form a so-called *Dalí Cross*. The name follows from the resemblance of the plot to the cross in the artwork *Corpus Hypercubus* [84] of Salvador Dalí. When unfolding *HyperLand*

to *SpaceLand*, the resulting *Dalí Cross* contains symmetric cases, which redundantly represent the same 3D spaces—analogously to unfolding a 3D cube into its 2D faces. A second representation excludes symmetric plots from the *Dalí Cross* and consists of only four unique 3D cubes (Figure 6.4b). Transiting directly from 4D to the *reduced SpaceLand* might be intuitively challenging [139], as we discuss in Section 6.5.5. We, therefore, leave it up to the users to select between a *Dalí Cross* or a *reduced SpaceLand* visualization. We further enable a focus+context approach [65]. To make the transition between *HyperLand* and *SpaceLand* smooth and intuitive, all encodings, e.g., color encodings and labels, are preserved [139].

6.5.3 *FlatLand*

Static 2D phase portraits are conventionally used by mathematical domain scientists, as shown in Figure 6.2. The tesseract representation of *HyperLand* consists of twenty-four 2D faces. As discussed in the description of Equation 6.2, omitting redundancies results in six unique 2D faces. A contiguous arrangement of the unique faces of all 3D cubes produces *FlatLand*, as the result of a smooth unfolding transition from *SpaceLand* (Figure 6.1c), following the principles of Section 6.5.5. This layout represents all combinations of 2D subspaces in a *trellis plot* configuration [73]. To preserve the mental model between the spaces [139], the *FlatLand* representation inherits all visual encodings from the previous *Lands*.

6.5.4 *TimeLines*

Apart from phase portraits, mathematical domain scientists also use representations that illustrate the temporal evolution of biological system variables (Figure 6.2c). The view showing the values of the trajectory variables vs. time is called *TimeLines* (Figure 6.1d). Obtaining *TimeLines* with a smooth transition from *FlatLand* cannot be achieved with simple unfolding operations that do not interfere with our perception [139]. A possibility would be to adopt mechanisms, such as unfolding and transiting the trajectories in *FlatLand* to one-dimensional subspaces, similar to the 1D slices of *Sliceplorer* [261]. This, however, would violate significant principles of animation [139, 262], as it will be discussed in Section 6.5.5. Therefore, we decided to have *TimeLines* as a separate static, but linked view and to employ transitions only between the 4D, 3D, and 2D subspaces. *TimeLines* are linked to the other *Lands* through brushing and linking [47, 60] and inherit also all visual encodings from them. An additional advantage of the linked *TimeLines* view will become obvious in Section 6.5.7. There, *TimeLines* are employed to dissect, explore, and analyze local system features, supporting the completion of task **(T2)**.

6.5.5 A Journey Across *Lands*

A straightforward approach to explore all available subspaces would be to use multiple coordinated views [276]. In our case, this would require a significant amount of screen space. For example, for the simple case of a 4D system, four different views would be
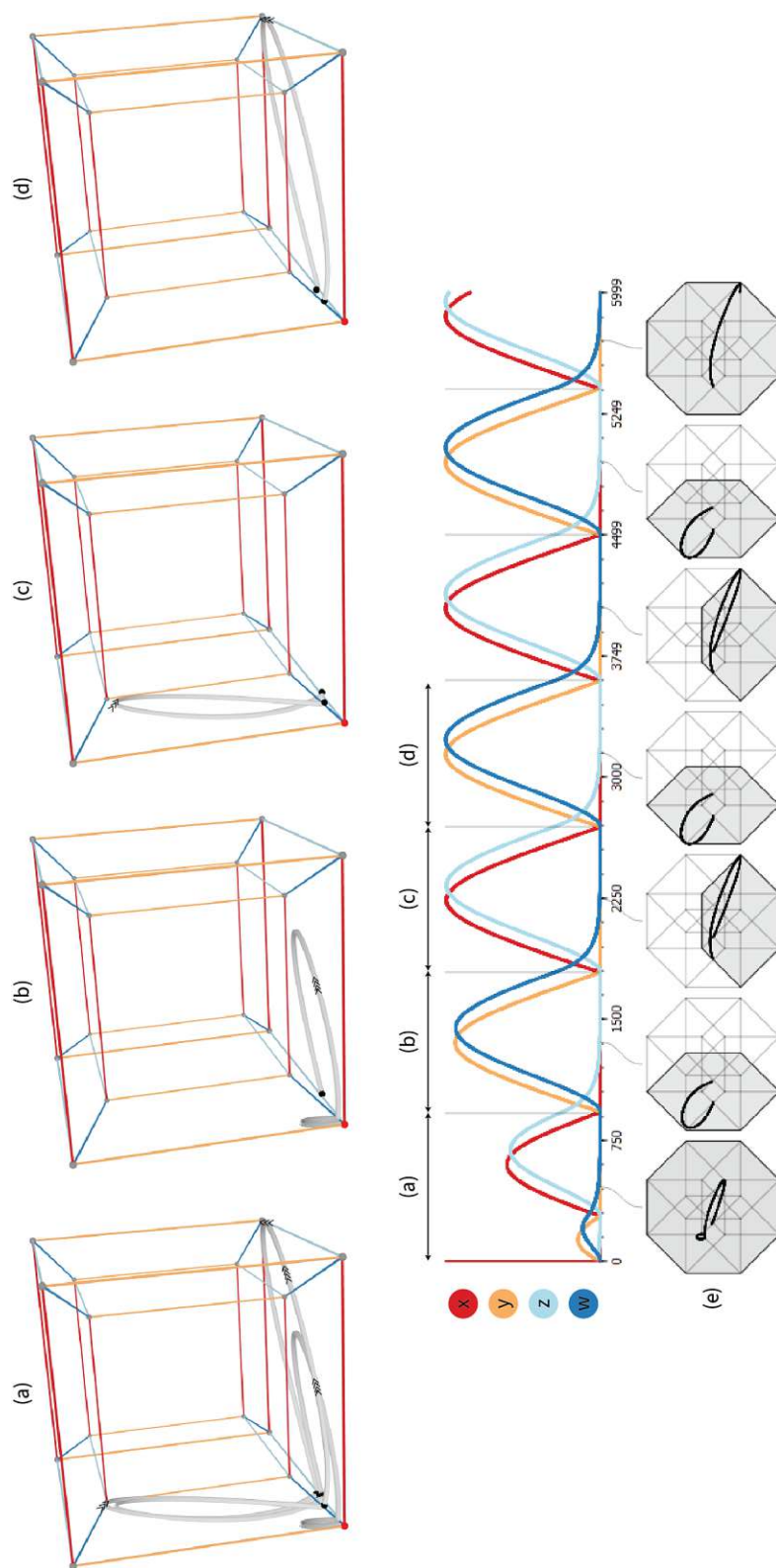
Figure 6.5: Depiction of the selection mechanism using brushing and linking between *TimeLines* and the *Lands* of *ManyLands*. Here, we use a *HyperLand* example. Each selection in *TimeLines*, indicated with (a), (b), (c), (d) is highlighted in *HyperLand* to denote the respective segment in 4D. Underneath the *TimeLines*, we depict *phase space compasses* (e). These are pictograms, representing respective segments (and their trajectory characteristics) from the dissection of *TimeLines*.

required. We would need one view for the 4D subspace, one for the four 3D subspaces, one for the six 2D subspaces—excluding symmetries and rotations—and one for the *TimeLines*. This would also require a significant amount of mental effort, e.g., when tracking the temporal development of a 4D trajectory. Given the temporal evolution of the systems, the varying dimensionality along segments of the trajectories, and the interest in localizing where the behavior of the trajectories alternates, we employ smooth transitions between the different subspaces. Smooth, animated transitions are of particular importance for tracking and understanding the current state of the system, and at the same time constitute the interactions between the system variables.

Although animation has been often considered to be a controversial topic [48, 139, 236], animated transitions can be beneficial for providing insights, especially for tracking—as long as the principles of congruence and apprehension are respected [139, 262]. According to the *congruence principle* [139], the structure and content of an animation must correspond to the desired structure and content of the mental representation. To this end, intermediate interpolation stages, e.g., between *Lands*, retain valid data graphics. Also, we use consistent mappings and visual encodings by having standardized transitions across graphic types. We respect correspondence and avoid ambiguity across transitions, e.g., for the axes labeling and coloring, and the trajectory attributes. According to the *apprehension principle* [139], the structure and content of the animation must be readily and accurately perceived and comprehended. To this end, we group similar transitions together and minimize occlusion, e.g., of redundant spaces, during transitions. Our transitions remain predictable and simple, but when more complex transitions are needed, such as in the *SpaceLand* to *FlatLand* transition, we use staging, i.e., grouping. The speed of transitions in *ManyLands* is at any case user-controlled. For a smooth transition between the 4D, 3D, and 2D subspaces, the user interacts with a simple slider that can be moved across different subspace configurations and their intermediate stages. The slider is annotated to facilitate traveling from one *Land* to another one, and slider snapping to *Land* positions is implemented. Other standard interactions, i.e., rotating, zooming, and panning, are possible in all *Lands*, but always independently from transitions. Hereby, we discuss all *Land* transitions.

***HyperLand* to *SpaceLand*:** *HyperLand* initially morphs into a *Dalí Cross* representation, which morphs, subsequently, into a *reduced SpaceLand* (Figure 6.4a). Without the intermediate *Dalí Cross* (Figure 6.4b), a sudden transition would be less intuitive and less easy to follow, as suggested by the apprehension principle [139, 262]. Still, the users can hide the symmetric subspaces within the *Dalí Cross*, and can travel directly to the *reduced SpaceLand* (Figure 6.4b). The transition employs staging, in accordance with the congruence and apprehension principles. First, the trajectory is replicated several times according to the number of the involved subspaces (eight cubes for the *Dalí Cross* and four for the *reduced* case). We need this to move each instance of the trajectory into the corresponding 3D cube. Then, each trajectory instance is projected into the respective cube. This projection is animated, so that the user can track the movement of the trajectory. Afterward, each 3D cube is rotated around one of its faces in the direction

that will bring the w-coordinate of each cube to a constant value. During this unfolding, the 4D camera must be moved into a position where the camera points towards the w-axis. So far, all operations are performed only in 4D space. Thus, during the animated unfolding, the dynamical system data are shown unaltered. After the animated unfolding, all vertices of the tesseract are lying on the same 3D space, i.e., their w coordinates are equal. Therefore, the w coordinate is discarded and the object is projected into 3D space.

SpaceLand to FlatLand: The transition between *SpaceLand* and *FlatLand* is performed in three stages (Figure 6.4). First, symmetrical 3D cubes are faded out in the animation, to remove redundant information. Second, the 3D trajectories are replicated into several instances and projected onto the faces of the 3D cubes (Figure 6.1). Third, the faces are rotated into the image plane. During this rotation, the 3D camera is also orthogonally aligned, providing a distortion-free view of *FlatLand* (Figure 6.1c).

Animations showing all transition stages in detail are available in our repository [34].

6.5.6 Holistic Analysis of 4D Trajectory Dynamics (T1)

For the holistic exploration and analysis of the trajectory behavior, domain scientists need an overview of the entire 4D phase space and its subspaces to be able to better observe certain phenomena. This is also important for the detection of alternations in trajectory behaviors across subspaces, and exploring where, how, and why these occur. A thorough exploration of all subspaces and the ability to drill down to each one of them is also required. Our approach for solving these subtasks involves primarily the ability to transition smoothly between the *Lands* **(T1.a)**. This is done as described in Sections 6.5.1–6.5.5. Apart from the design of the *Lands* and the ability to transition between them, we need to convey further information about the dynamical system **(T1.b)**. For example, we need to convey the presence of *slow-fast dynamics*, i.e., a pronounced phenomenon during which some variables are changing faster than others, the *local dimensionality*, or *switching phenomena*. This provides mathematical domain scientists with important knowledge on how and where to start their analysis. We use an intuitive strategy, where we dissect the trajectory based on one of these three manifestations. The slow-fast behavior, i.e., the velocity vector at each point along the trajectory is given by the right-hand side of the ODE system in Equation 6.1. The dimensionality is the number of active (non-zero) variables at each trajectory segment. The derivation of switching behaviors is not straightforward, as it is often computed for each model individually. We, therefore, give an initial suggestion to the user based on changes in the slope along the trajectory.

Subsequently, we encode this information directly on the trajectory with colors and glyphs. For the *color encoding of the switches* or *the slow-fast behavior*, we use a logarithmic, perceptually uniform, greyscale colormap [59]. The colormap is truncated on the white side to not interfere with the background color of the scene. The greyscale choice is due to the presence of other colors in the scene, e.g., to encode the axes. We chose this encoding as an "inking" metaphor: the slower a point is moving on the trajectory,

the darker is its ink footprint. The logarithmic scaling is used to compensate for the—sometimes—extremely fast or slow motion along the trajectory. For the *glyph encoding of the dimensionality*, we use a corresponding number of arrows, e.g., for a 2D segment, we show two arrows. These two visual encodings are used across all *Lands* to preserve semantics, and demonstrated in the *HyperLands* of Figure 6.5. A fly-through probe that travels along the trajectory indicates the temporal evolution of a trajectory, similar to the colored dots employed in Figure 6.2. In the *Lands*, this is a sphere, and in *TimeLines*, a line sliding across time.

The combination of smooth transitions across *HyperLand*, *SpaceLand*, and *FlatLand* with a bi-directionally interactively linked view on the *TimeLines* enables mathematical domain scientists to perform task **(T1)**. They are now able to effectively interact and transit between different visual representations of the dynamical system. The smooth transition between the various *Lands* and the link to the *TimeLines* provides consistent information across the different subspaces that conveys to the user a holistic view on the entire space of the 4D trajectory.

6.5.7 Dissected Analysis of 4D Trajectory Dynamics (T2)

The localized exploration and understanding of dynamical systems trajectories and their underlying biological processes is also important. Several phenomena may be observed only in smaller or lower-dimensional regions, i.e., trajectory segments. To provide a strategy for the easy localization, exploration, and analysis of the dynamical system representation across subspaces and trajectory segments, we extend the *TimeLines* to be used as a selection mechanism for subspaces/segments of interest.

As mentioned also for the first task, we can dissect the trajectories represented in *TimeLines* into segments, depending on their dimensionality, slow-fast, or switching behavior **(T2.a)**. With trajectory dissections available, we propose an additional representation—the *phase space compasses*, or simply, *compasses*. These are abstracted, small-multiple tesseract pictograms [90] that depict the active variables (and relevant subspaces) of the corresponding trajectory segment [73]. Each individual *compass* is, in essence, an orthogonal projection of the tesseract to the screen space, given by the isometric projection matrix:

$$P = \begin{pmatrix} 1 & 0 & \cos(\alpha) & -\cos(\beta) & 0 \\ 0 & -1 & \sin(\alpha) & \sin(\beta) & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{6.6}$$

where α and β are the orientations of the z and w axes in screen space.

The *compasses* **(T2.b)** are configured in a trellis representation, under the respective segments of the *TimeLines* (Figure 6.5e). If a segment of a trajectory is inactive in one of its four dimensions, then it "lives" in a 3D domain, and a *compass* denoting the respective 3D subspace is created (Figure 6.5b–d). A trajectory segment can be mapped

together with its characteristics on such a *compass*, using the same visual encodings as within the *Lands*. This is shown in Figures 6.1e and 6.5e in greyscale. Hovering over the *compasses* provides a magnified view, to enhance visibility, e.g., when there are too many segments or when the display size is small. Selecting a *compass* provides a focus+context view on the respective segment of the trajectory and a smooth, continuous transition to the respective subspace **(T2.c)**. Afterward, through the interaction described in Section 6.5.6, the domain scientist can navigate across other subspaces of the segment. Arbitrary selections of segments on the *TimeLines* are possible within and across all *Lands*, using brushing and linking [47, 60, 286] (Figure 6.5). The abstracted representation of dissections with *compasses* offers easier navigation across subspaces and segments. It also facilitates interaction and the selection of trajectory segments.

6.5.8   Integration of Knowledge Discovery with Generation of Phase Space Illustrations and Animations (T3)

The final task for mathematical domain scientists is to generate meaningful, scientific illustrations for further analysis, for education and externalization of their findings, or as a support to their mathematical proofs. These illustrations are usually based on their knowledge of the dynamical systems. Now, they can integrate the newly formed knowledge from the visual analysis, as performed through tasks **T1**–**2**, with the generation of illustrations in the web-based platform of *ManyLands*. We employ a flexible web-based design space, where users can specify the appearance of their representations. Rotations, alterations of the rendering scene, and aesthetic changes to the representations are feasible, with the use of simple, interactive sliders. The results of our illustration generation functionality can be seen throughout all the figures of this chapter. Analogously, animations can be generated.

6.6   Results and Evaluation

After the design of *ManyLands*, we conducted an informal evaluation with four domain scientists, following the guidelines of Isenberg et al. [153]. The domain scientists are researchers in applied mathematics, working on the exploration of 4D dynamical systems of biological interest. Their field experience varies from *low* (student with few years in the field) to *very high* (professor with > 30 years in the field). Two of them have normal and two normal-to-corrected vision. One of the participants had been actively involved in the design of *ManyLands*. At the beginning of the evaluation session, we introduced *ManyLands* to the domain scientists, and we demonstrated the basic functionality and main components of the framework. Afterward, the domain experts proceeded with the investigation of three use cases. A visual environment for the exploration and analysis of the respective 4D dynamical systems was simulated. We used the think-aloud method, allowing the experts to comment and discuss visualizations and potential insights. Discussions among themselves were allowed, like in a real-life collaborative scenario. By providing *ManyLands* as a web-based application, domain scientists could interact with

the systems in real-time and generate custom illustrations. All three tasks of Section 6.3 were executed, allowing domain experts to reason about the employed visual encodings, interaction mechanisms, and potential findings (Section 6.6.1). Subsequently, our domain experts completed individually a questionnaire to give us feedback on their experience with *ManyLands* (Section 6.6.2).

6.6.1 Use Cases

In this section, we present three use cases for the exploration and analysis of 4D trajectories with *ManyLands*. The dynamical systems and initial conditions of these cases were provided by the evaluation participants, based on what phenomena they are currently interested in. These are documented in detail in our repository [34]. For the investigation of the trajectories, the domain scientists employed *ManyLands*, as they would do in a real-life analysis scenario. The findings presented in this section have been documented by the domain scientists. Given the focus of a visualization chapter, these are not meant for direct inferences on the dynamical systems (even more, on the underlying biological systems)—they are, rather, exemplifications of whether (and how) *ManyLands* enables them to conduct their analysis.

Use Case A—Analysis of Bipolar Disorder: The Goldbeter model of bipolar disorder [124] is explored in the first use case. It is a 4D system of ODEs describing the dynamics of a bipolar disorder, i.e., the alternations between manic and depressive state. For certain initial conditions, the model exhibits oscillatory behavior, i.e., mania and depression alternate periodically. Moreover, the four variables of the system change over different time scales, as shown in Figure 6.6f. The first straightforward observation is that the system exhibits switch-like oscillations, as seen in the *TimeLines* (Figure 6.6f). By initializing the *HyperLand* and *SpaceLand* view, the corresponding trajectory is presented in the phase space (Figure 6.6a–b). The color of the trajectory remains grey primarily and does not show significant differences in speed. Starting with a certain initial condition, the orbit converges to the so-called limit cycle, which can be seen easily when moving to *FlatLand* with the "loop-like" geometry (Figure 6.6c). This phenomenon is already visible in *HyperLand* and *SpaceLand* (Figure 6.6a–b). Extracting the segments of the trajectories indicated in the *TimeLines* (Figure 6.6f) and inspecting the respective *compasses* shows also the repetitive pattern of the closed trajectory (Figure 6.6d–e). This confirms that the observed oscillations are of limit cycle type. The analysis of the bipolar model trajectories in *ManyLands* revealed the switch-like nature of the model dynamics and confirms results obtained from a prior geometric analysis.

Use Case B—Analysis of NF-κB Pathway: A trajectory from a mathematical model describing the NF-κB signaling pathway [175] is explored in the second use case. The NF-κB system plays a key role in regulating the immune response to infection. When the NF-κB system is activated, the model exhibits damped oscillations for certain initial conditions. This means that the transcription factor NF-κB is located in and out of the nucleus in a periodic fashion, and finally stays in the cytoplasm, i.e., the system settles at a stable equilibrium point. In *ManyLands*, two numerically computed solutions for

two different initial conditions have been analyzed simultaneously. The solutions exhibit a transition towards the stable equilibrium point, as shown in *HyperLand*, *SpaceLand*, and *Flatland* (Figure 6.7a–c, marked with the red dots). This transition happens in a switch-like manner, i.e., the variable x changes its values from non-zero to zero in a repeated fashion and eventually remains at a nonzero constant value, as shown in *TimeLines* (Figure 6.7e). Also, the *compasses* (Figure 6.7e) show that some parts of the oscillatory phenomena can be observed in specific subspaces, e.g., the fast (light grey) behavior at the beginning of one trajectory "lives" in 4D (Figure 6.7d). The application of *ManyLands* to the NF-κB model enabled the geometric analysis of its complicated 4D dynamics—also, for two solutions. In particular, the dissections in *TimeLand* support and complement the mathematical model dissection to segments of interest in phase space.

Use Case C—Analysis of Peroxidase−Oxidase Reaction: The Olsen model for the peroxidase−oxidase reaction [176] is explored in the third use case. The Olsen model is a chaotic attractor, for which domain scientists often look only at specific combinations of the ZW, XZW and XYW spaces. The 4D view of *HyperLand* (Figure 6.8a) can confirm at a glance the chaotic attractor behavior, as denoted by the multitude of loops. By transiting to *FlatLand*—in particular, to the ZW space—the trajectory comes repeatedly close to the origin and spends longer intervals of time there (Figure 6.8b). With focus+context on *SpaceLand*, the XZW (Figure 6.8c) and XYW spaces (Figure 6.8d) are illustrated. In the XZW space, the system is a chaotic attractor, while in the XYW space, it exhibits a transient. Practically, the system exhibits a slow motion and makes a long excursion in the $XYZW$ space before returning to $(Z, W) = (0, 0)$. Additionally, the initial part of *TimeLines* shows changes in speed (Figure 6.8e). The second part (Figure 6.8f) manifests periodic behavior, while in-between a switching behavior can be also noticed (Figure 6.8g). *ManyLands* facilitated the analysis of the complex 4D dynamics of the Olsen model, supporting a geometric analysis of the chaotic attractor and revealing its oscillatory behavior.

In the three explored cases, the transitions from one space to another **(T1)** and the interactive dissection with the compasses **(T2)** have proven to be very helpful for the analysis and understanding of the mathematical behavior of the systems, as well as for the creation of the aesthetic, scientific illustrations of Figures 6.6–6.8 **(T3)**.

6.6.2 User Experience

For the evaluation of the user experience, we designed a questionnaire. The *first part* was related to tasks **(T1-3)**. Each question required an open answer, and a grading in a Likert scale $(++$ to $--)$ for the perceived effectiveness, efficiency, and satisfaction. In Figure 6.9, we summarize the results of the evaluation. One of the participants was involved in the design of *ManyLands*—and the corresponding results are presented separately. The visualizations received positive grades for all tasks. Tasks **(T1.b)** and **(T3)** received neutral grades $(=)$ by the most experienced domain scientist. For task **(T1.b)**, the reason is that they would like to map additional features onto the trajectory.
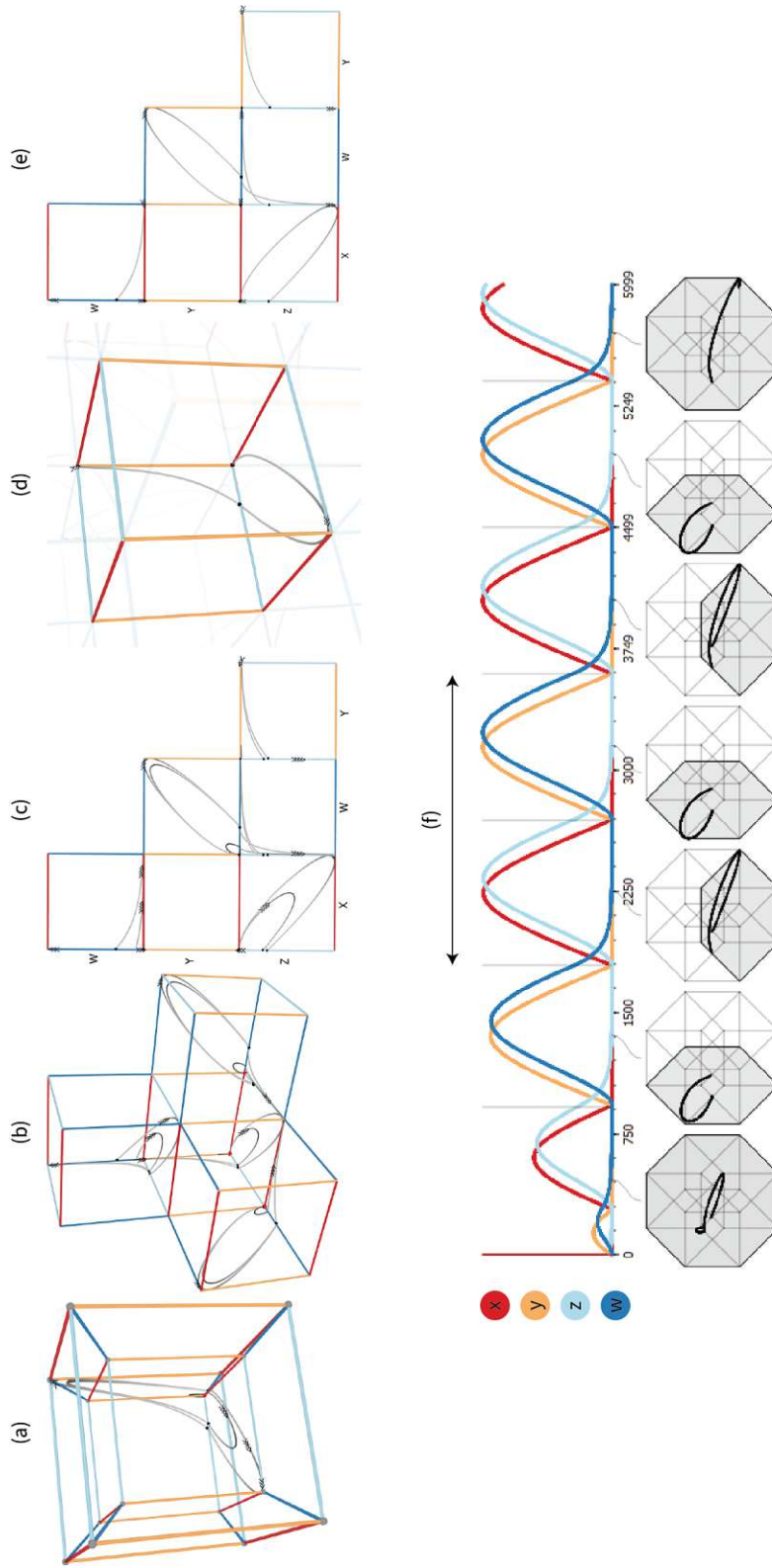
Figure 6.6: Analysis conducted by the domain scientists for a trajectory, describing a bipolar disorder [124]. With (a)–(f), we denote the visual analysis steps, as described in Section 6.6.1, Use Case A.

Figure 6.7: Analysis conducted by the domain scientists for a trajectory, describing the NF-κB signaling pathway [175]. With (a)–(e), we denote the visual analysis steps of Section 6.6.1, Use Case B.
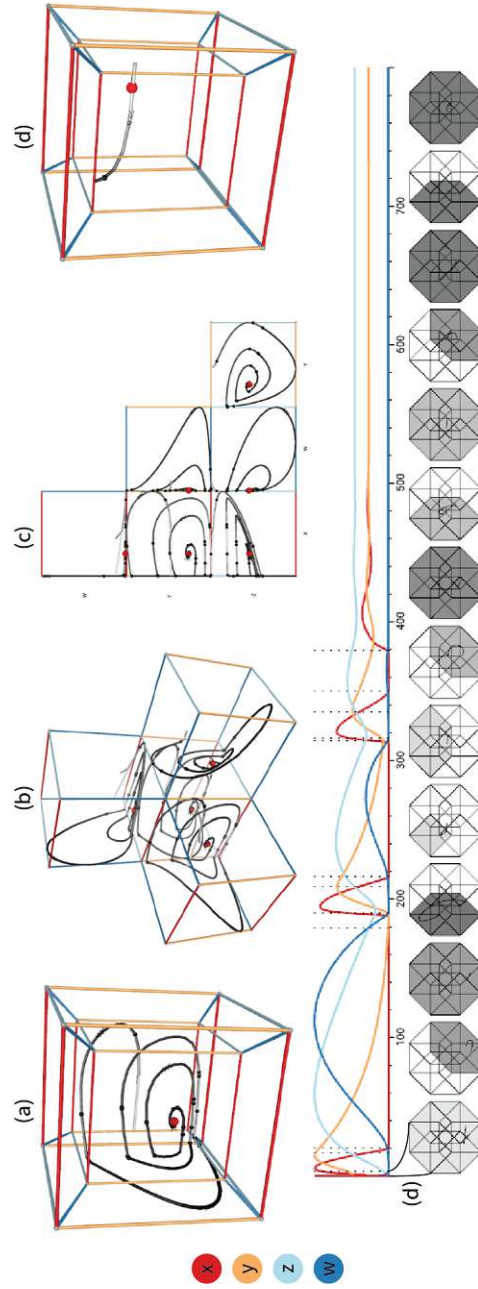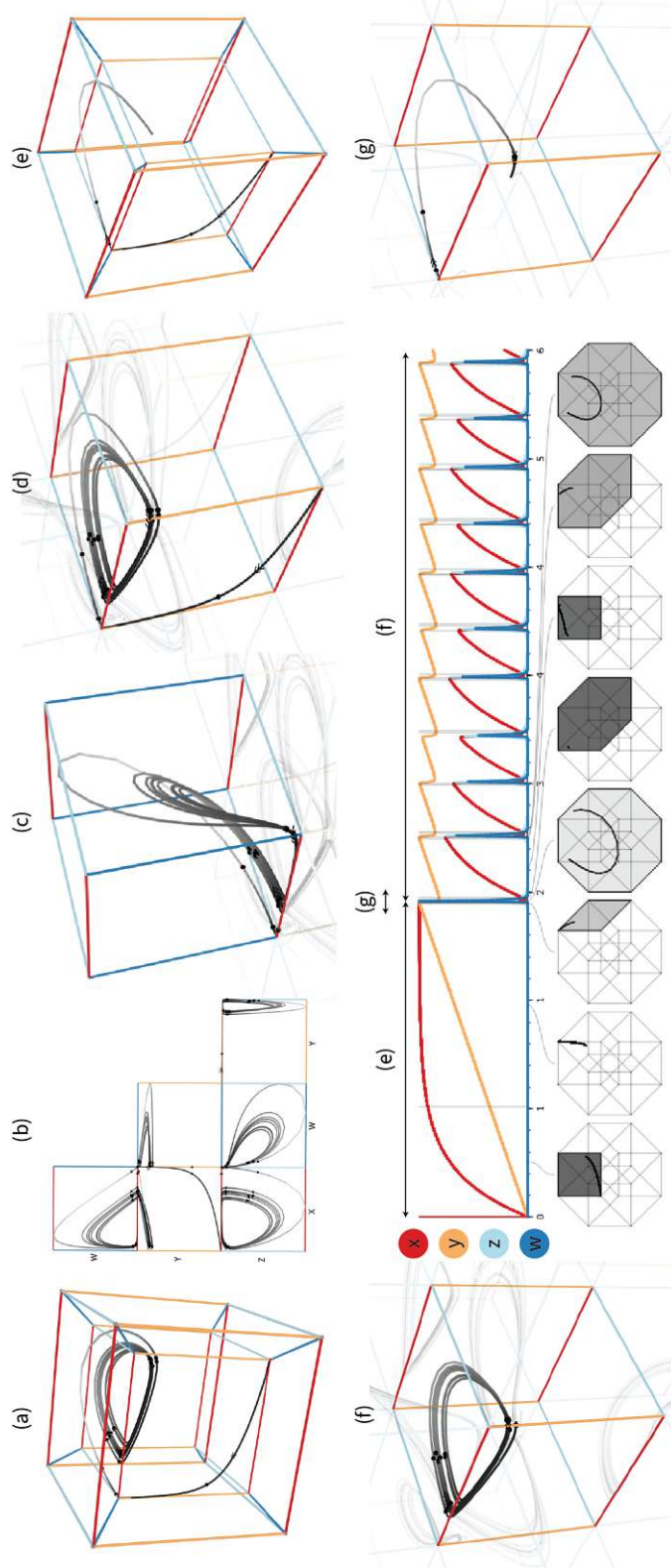
Figure 6.8: Analysis conducted by the domain scientists for a trajectory, describing the peroxidase–oxidase reaction [176]. With (a)–(g), we denote the visual analysis steps, as described in Section 6.6.1, Use Case C.
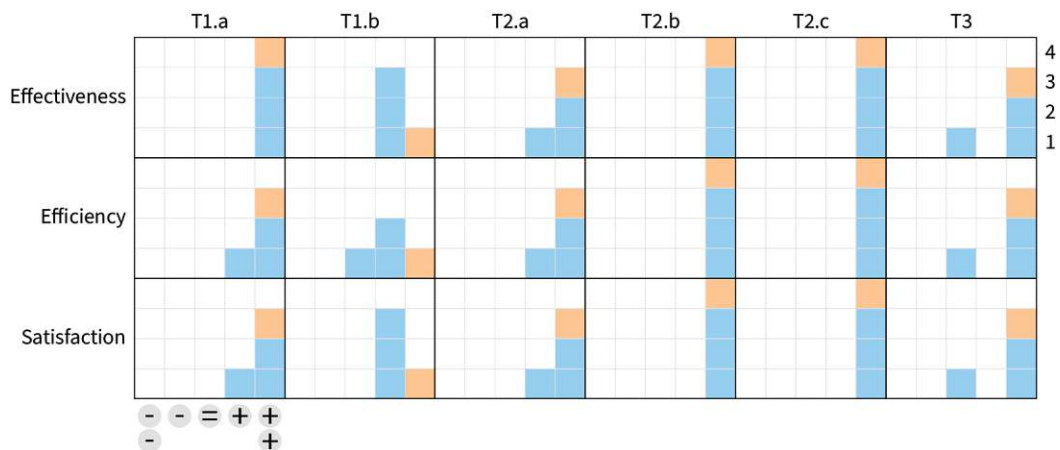
Figure 6.9: Evaluation outcomes for user experience. Each task **(T1–3)** has been graded according to its perceived effectiveness, efficiency, and satisfaction on a 5-points Likert scale (−− to ++). We denote the domain expert that was involved in the design of *ManyLands* with orange squares, and the others with blue squares.

For task **(T3)**, the scientist commented that the functionality should be tested also with other systems, where there is no prior understanding of the dynamics—as opposed to now, where the dynamics were already known. The most appreciated functionalities were the animated, smooth transitions and the facilitated interaction for trajectory exploration.

In the *second part*, the participants were asked to compare *ManyLands* to what they are currently using and to evaluate the overall usefulness of our tool—including strengths, weaknesses, limitations, and future improvements. They commented that *"DYNAMICA allows visualizations only in 3D and 2D. ManyLands gives new possibilities and access into 4D"*. All participants agreed that the visual tool is overall understandable and useful, and that they would be able to use it after training: *"ManyLands already helps. It will help even more after reasonable training, due to the ease of switching between representations"*. One participant stated that *"it helps to understand the underlying geometry of the phase space better, it contributes to the understanding of the model dynamics and supports a geometric mathematical analysis of a given system"*. One experienced participant summarized their overall opinion as: *"ManyLands has a user-friendly interface for rapid re-analysis and visualization. By experimenting with different initial conditions and parameter values, the trajectory plot will update accordingly; this process will assist the development of better understanding of model dynamics. In contrast to other dynamical systems software, it is a straightforward tool for phase space visualization that allows not only to show numerically computed trajectories on a tesseract, but shows also the time evolution of variables. When needed, all possible variants of 2D and 3D phase portraits are shown at once. While dynamical systems knowledge is essential, ManyLands can make a proficient domain scientist more productive, while empowering students with limited visual mathematical/geometrical skills."*. The potential of *ManyLands* is,

according to the evaluators, the integrated ability to transition across *Lands*, while interactively inspecting *TimeLines*. Improvement suggestions were related to redefining approximations to reflect better the mathematical descriptions, while one participant suggested that saving viewpoints and snapping back to them would be valuable.

6.6.3 Discussion

Our evaluation and the use cases conducted by the mathematical domain experts showed that *ManyLands* introduces new capabilities into the workflow of phase space exploration and analysis of 4D trajectories. This positive outcome is partially due to our application targets explicitly the requirements of Section 6.3. The introduction of the 4D representation and the use of smooth, animated transitions were proposed by our domain scientists as a suitable solution to their current issues with the use of 2D and 3D phase portraits. For this reason, and as discussed in Section 6.5.5, we did not investigate further or compare to other obvious alternatives, such as multiple coordinated views [47]. Additionally, the visualization literacy of our intended users has also been considered. If *ManyLands* were to be extended to other application domains, an additional evaluation considering all potential alternatives should be conducted. In this case, scalability beyond 4D systems and beyond a small number of trajectories should also be regarded. In the present context, this was considered out of scope by our domain scientists. Scalability to higher-dimensional systems is currently not relevant given their particular focus on 4D systems of biological interest. Considering a vast number of trajectories is also neither biologically justified nor mathematically relevant for them. The trajectories relate to specific initial conditions reflecting biological setups. Therefore, looking into all possible solutions is not necessary, and only in few cases, a small number of them (2-3) might be investigated simultaneously. Parameter sensitivity analysis has also not been investigated, for the same reasons. We propose these topics for future work in Section 6.7.

6.7 Conclusions and Future Work

Observing the geometry of dynamical systems is anticipated to provide mathematical domain scientists with a significant understanding of the dynamics, and potentially, of underlying modeled biological processes. Mathematical experts are now restricted to the detached use of computer-generated 2D or 3D phase portraits and of hand-drawn illustrations, hampering an integrated analysis of system trajectories. *ManyLands* is a new visualization tool to support domain scientists to understand, analyze, and illustrate trajectories of 4D dynamical systems that describe biological processes.

There are several promising avenues for future research. Most importantly, the scalability of our proposed approach must be investigated with strategies to accommodate dynamical systems beyond 4D, and the comparison of multiple trajectories. These two topics were currently not relevant for our collaborating domain scientists as they are not encountered in their applications. However, they are ubiquitous in many other application domains, e.g., in the physical and environmental sciences. To support systems higher than 4D, combining

ManyLands with *rolling the dice* [99] would facilitate querying and navigating the entire n-dimensional ($n > 4$) phase space. Alternatively, the use of non-linear embeddings [190] could also be investigated. To compare multiple trajectories emerging from different initial conditions, techniques from ensemble visualization could be incorporated, e.g., contour boxplots [282] and curve boxplots [203], or approaches from weather forecast ensemble visualizations [109, 110]. Other directions include the smart incorporation of additional topological features in the representations and improvements regarding visual perception, e.g., by depth-dependent halos [102], or by illumination and shadowing [96]. *ManyLands* has opened interesting directions for the exploration of phase spaces, enabling the analysis of 4D trajectories of systems that model biological processes.

CHAPTER 7

Conclusion

7.1 Summary

In this dissertation, we introduced the visual analysis of defects to integrate visual analytics and defect analysis. We proposed a four-step workflow for the visual analysis of defects based on the PDCA cycle of continual improvement. In the *defect prevention* step, domain experts plan the design and development process, foresee possible defects and assure that intended results can be achieved. In the *control of defective outputs* step, domain experts implement the design and development process and immediately resolve arising defects. In the *performance evaluation* step, domain experts inspect object characteristics before delivering it to customers to ensure it conforms to all the requirements. In the *improvement* step, domain experts reflect on the entire design and development process, detect bottlenecks and identify strategies to improve the process and make it more efficient.

We investigated strategies supporting the visual analysis of defects within these aforementioned steps. In Chapter 2, we identified, structured and discussed relevant visualization and visual analytics methods to the visual analysis of defects. This fulfilled our first research objective [**Obj1**]. In Chapters 3–6, we presented the design of four solutions addressing the four distinct steps of the visual analysis of defects. These solutions fulfilled our second research objective [**Obj2**] in the following ways:

- *Defect prevention:* in Chapter 3, we presented a novel technique that assists dental technicians and patients in dental treatment planning. The proposed method predicts and visualizes a new patient's appearance with a selected denture. This preview facilitates an early feedback to build a patient's trust and confidence in the treatment outcome.

141

- *Control of defective outputs:* in Chapter 4, we presented a system to support dental technicians in designing aesthetic and functional dental prosthetic devices or dentures. The system is based on simulations of the patient's face after dental treatment, and it analyzes facial and dental proportions to provide the dental technician with information on the aesthetics. The dental technicians rely on this information during the design process to ensure that the denture fits the patient's face. The system is integrated into state-of-the-art CAD software and allows following the dental reconstruction workflow. For example, the dental technician at any stage can switch to a functional analysis module to review how changes introduced to improve aesthetics affect functional properties of the denture.

- *Performance evaluation:* in Chapter 5, we presented an approach to quantify defects in CT images of composite materials. Together with material engineers, we designed a system to segment voids and classify them into four defect types: matrix fractures, fiber/matrix debondings, fiber pull-outs, and fiber fractures. To analyze these defects and their characteristics, the system provides several visualization techniques. Overall, the proposed system allows material engineers to study the correlation between fractures and material properties (e.g., fiber orientation or fiber volume ratio).

- *Improvement:* in Chapter 6, we introduced a new system that enables smooth transitions between dimensions (2D↔4D) to support domain scientists in exploring and understanding the data.

Our proposed solutions addressed defects of different types, which fulfilled our third research objective [**Obj3**]. Solutions presented in Chapters 3 and 4 handle *design space defects*. These are defects that prevent dentures from functioning correctly, due to a combination of factors stemming from the design space (Figure 7.1 a). For example, dentures causing the patient physical and/or physiological discomfort due to their shape and/or aesthetics. The solution presented in Chapter 5 handles *spatial* and *spatiotemporal defects*. The spatiotemporal defects are matrix fractures or fibers fractures that increase their size and quantity in time (Figure 7.1 b). A spatial defect is the final fracture of the test specimen that typically happens in the last stage of an *in situ* tensile test (Figure 7.1 c). The solution presented in Chapter 6 handles *temporal defects* within the context of biological processes, i.e., object characteristics that change in time (Figure 7.1 d).

The presented solutions also demonstrated a strong *impact* on the visualization, visual analytics and domain communities. Chapter 3 is based on a paper [28] that was cited in a patent "Dental Design Transfer" [179] as a document relevant to the whole patent in category A (documents defining state of the art and not prejudicing novelty or inventive step). This demonstrates the usefulness and industrial interest in the approach of dental design preview. Chapter 5 is based on a paper [29] that was cited in the visualization and the material science community. Domain experts used the tool's results to characterize
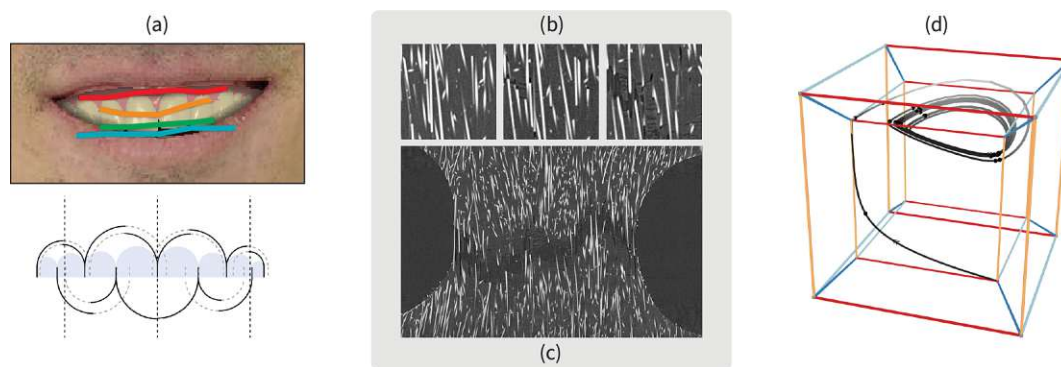
Figure 7.1: In this dissertation we considered defects of different types: (a) design space defects, (b) spatiotemporal defects, (c) spatial defects, and (d) temporal defects.

short fiber reinforced polymers and published them in a related domain conference [227]. Chapter 6 is based on a paper [35] that was awarded as the best picture (people's choice) at Eurographics Workshop on Visual Computing for Biology and Medicine 2019. Additionally, the paper has featured in the TU Wien Magazine [14].

Our research aim was to "investigate strategies supporting the visual analysis of defects within the four steps of defect prevention, control of defective outputs, performance evaluation, and improvement" (Chapter 1, page 6). In this section, we discussed how we addressed the research objectives and the impact of our work. Thus we conclude that:

> We have met our research aim by completing all three objectives.

7.2 Outlook

Visual analysis of defects opens many exiting directions for future work. Being a highly relevant topic to the industry, it evolves with technology development and continuous manufacturing process optimization and improvement. Many defect analysis applications benefit from the digital workflow, allowing to automatize previously tedious and time-consuming procedures (e.g., medical treatment planning, mechanical engineering, architectural design). This section discusses the most prominent ones and provides our perspective for interesting future directions.

An promising research perspective is the integration of systems for the visual analysis of defects and modern CAD software. Both systems would mutually benefit from integration, as this would allow designing objects in an assisted virtual environment. Such integrated visual analysis of defects and CAD systems use existing domain knowledge and support the user in decision-making and provide additional guidelines to avoid or resolve defects. Highly efficient and interactive techniques bring the user experience to a new level, allowing instant hypotheses generation and verification. Shortening workflow cycles for

visual analysis of defects saves the user time and provides a better experience. These systems often show a predicted outcome, which helps users to evaluate the impact of a design change. The user can create several design prototypes and identify the most appropriate one based on the system's suggestions in an engaging and interactive way.

Additionally, the current technology development enables capturing or creating more data, which brings several challenges to the visual analysis of defects. (i) *Progressiveness:* continuous data growth slows the performance down of visual analytics systems as large data visualizations are computationally expensive and require powerful hardware. Progressive visual analytics methods optimize calculations and provide the user with an enhanced system's performance. (ii) *Trustworthiness and explainability:* large data visualizations often employ automated data analysis approaches, which are not always understandable or easily explainable to the user. Visual analysis of defects methods can support the users in understanding the large corpus of data and the employed automated data analysis methods for an increased trust into the analysis process and the achieved results. (iii) *Visual clutter:* big data is complex, which affects visualizations and makes them cluttered and hard to understand. Domain experts can benefit from visual analytics techniques that can handle complex and cluttered views on the data. (iv) *Efficient data formats:* big data is challenging to store as it requires significant storage space. The situation is getting more serious when new data is created during the analysis procedure or a simulation. Efficient ways of storing data or protocol-achieved results is crucial as it increases the reputability of analyses results.

Another future challenge rises from the fact that mathematical models describing various processes are quite complex and often go far beyond 3D. The analysis of such models requires advanced techniques to harvest the model complexity and to support an analysis within *high-dimensional spaces.* This challenge opens the following promising research directions: (i) *Dimensionality reduction:* applying a dimensionality reduction technique to a model and visualizing it using conventional 2D/3D plots helps study phenomena efficiently. However, in this case, some information is lost during the data transformation. To avoid this, one can keep a mapping between the original and reduced data, which enhances reproducibility and clarity of the analysis. (ii) *Multidimensional visualization:* employing sophisticated multi-dimensional visualizations to visualize models is a prominent direction. However, such visualizations may bring additional challenges with result interpretation and interaction. The challenges can be approached by employing advanced data exploration techniques or novel display devices such as head-mounted or volumetric displays. For example, virtual reality allows the user to immerse into the data space and focus on its spatial properties. (iii) *Parameter space analysis:* techniques that support the exploration and analysis of a broad spectrum of model parameters help understand the model behavior. These techniques perform multiple simulations of the model with different parameters and study how parameter changes affect the result.

Finally, as discussed in the previous chapters of this dissertation, visual analysis of defects is relevant for many different applications. An important direction of future research would be to investigate generalizable visualization and visual analytics strategies suitable to various datasets and domains. Studying these strategies would allow the development of reusable frameworks with significant impact.

This dissertation is an important first step towards bridging the gap between defect analysis and visual analytics by introducing the visual analysis of defects. We proposed a formal workflow for the visual analysis of defects and thus established a solid ground for future developments. Our presented strategies targeted current problems in the visual analysis of defects and provided concrete and reasonable answers. We believe that the theoretical and practical foundations of the presented research will further promote the introduction and integration of visual analysis of defects within industrial processes.

Glossary

control of defective outputs the second step of the visual analysis of defects, where domain experts implement the processes and control defects arising throughout these processes. ix, 5, 6, 9, 13, 18, 24, 67, 141–143

defect analysis a part of quality management that is focused on controlling and resolving defects. ix, 5, 6, 19, 141, 143, 145

defect prevention the first step of the visual analysis of defects, where domain experts plan the design and development processes to ensure that intended results can be achieved while forecasting risks and opportunities. ix, 5, 6, 8, 13, 18, 20, 47, 141, 143

design space defects defects that are described as (combinations of) object characteristics in the design space that prevent the object from functioning correctly (e.g., bridge collapse during bad weather). 3, 6, 28, 33, 142, 143

improvement the fourth step of the visual analysis of defects, where domain experts explore possible actions to improve the object quality. ix, x, 6, 11, 13, 19, 34, 117, 141–143

performance evaluation the third step of the visual analysis of defects, where domain experts measure the object's characteristics to ensure that defective outputs are identified. ix, x, 5, 6, 10, 13, 18, 28, 95, 141–143

spatial defects defects that are described as geometrical objects in Euclidean space or on a Euclidean plane (e.g., material fracture). 3, 28, 29, 143

spatiotemporal defects defects that are described as geometrical objects in Euclidean space or on a Euclidean plane measured in time (e.g., corrosion development in time). 3, 31, 142, 143

temporal defects defects that are described as object characteristics measured in time (e.g., the brake system temperature measured multiple times during stopping). 3, 30, 142, 143

147

visual analysis of defects visually assisted defect analysis. ix, 6, 7, 13, 15, 18, 19, 30, 34, 141, 143–145

visual analytics the science of analytical reasoning facilitated by interactive visual interfaces. ix, 6, 13, 16–19, 34, 35, 42, 141, 142, 144, 145

Acronyms

3DCT 3D X-ray computed tomography. 96–98, 100–102, 105

AR augmented reality. 50, 51, 53

CAD computer-aided design. 31, 48–50, 67, 142, 143

CAM computer-aided manufacturing. 48

CRediT contributor roles taxonomy. 7–11

CT computed tomography. 3, 10, 22, 25, 31, 32, 37–42, 95, 142

GFRPs glass fiber reinforced polymers. 37, 95–98, 115

NF-κB nuclear factor kappa-light-chain-enhancer of activated B cells. 33, 34, 133, 134, 136

ODE ordinary differential equation. 119, 130

ODEs ordinary differential equations. 33, 34, 117, 119, 120, 133

PCA principal component analysis. 42, 43

PDCA plan-do-check-act. ix, 4, 5, 8–11, 13, 34, 47, 67, 95, 117, 141

t-SNE t-distributed stochastic neighbor embedding. 43

Bibliography

[1] 3Shape. `https://www.3shape.com/de-de`. Accessed: 18.06.2021.

[2] Amira-Avizo 3D Software. `https://www.fei.com/software/avizo3d/`. Accessed: 18.06.2021.

[3] Apple Vision. `https://developer.apple.com/documentation/vision`. Accessed: 26.06.2021.

[4] ARCore. `https://developers.google.com/ar`. Accessed: 26.06.2021.

[5] ARToolKit. `https://artoolkit.org/`. Accessed: 26.06.2021.

[6] Casrai, contributor roles taxonomy. `https://casrai.org/credit/`. Accessed: 05.07.2021.

[7] Chaoscope: a 3D strange attractors rendering software. `http://www.chaoscope.org/`. Accessed: 2021-01-18.

[8] exocad. `https://exocad.com/`. Accessed: 18.06.2021.

[9] The insight segmentation and registration toolkit. `https://www.itk.org`. Accessed: 18.06.2021.

[10] Kapanu AG. `https://www.kapanu.com/`. Accessed: 18.06.2021.

[11] Merz Dental GmbH, baltic denture system. `https://www.baltic-denture-system.de`. Accessed: 12.06.2020.

[12] Subsurface profile shading model. `https://docs.unrealengine.com/en-us/Engine/Rendering/Materials/LightingModels/SubSurfaceProfile`. Accessed: 18.06.2021.

[13] Théodore Olivier, intersection of an oblique cylinder and an oblique conoid (yellow and blue), ca. 1830–1845, fruitwood, brass, thread, lead weights, union college permanent collection, 1868.28 UCPC. `https://muse.union.edu/mandeville/project/olivier-models/`. Accessed: 2021-06-09.

151

[14] TU Wien Magazine—Mathematics (Issue 4, November 2019). `https://www.tuwien.at/en/tu-wien/about-tu-wien/reports-and-documents/tu-wien-magazine`. Accessed: 27.06.2021.

[15] Unreal Engine. `https://www.unrealengine.com/`. Accessed: 18.06.2021.

[16] VGSTUDIO MAX. `https://www.volumegraphics.com/en/products/vgsm.html`. Accessed: 18.06.2021.

[17] World health organization, oral health: Key facts. `https://www.who.int/news-room/fact-sheets/detail/oral-health`. Accessed: 19.05.2020.

[18] Directorate General Health and Consumers, Eurobarometer 72.3: Oral Health, 2010. `https://ec.europa.eu/commfrontoffice/publicopinion/archives/ebs/ebs_330_en.pdf`. Accessed 19.05.2020.

[19] E. A. Abbott. *Flatland: a Romance of Many Dimensions.* Seeley & Co, London, 1884.

[20] R. H. Abraham and C. D. Shaw. *Dynamics-The geometry of behavior.* Aerial Press, 1983.

[21] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on pattern analysis and machine intelligence*, 16(6):641–647, 1994.

[22] R. Aguayo. *Dr. Deming: The American Who Taught the Japanese About Quality.* A Fireside book. Touchstone, 1991.

[23] A. Aichert, W. Wein, A. Ladikos, T. Reichl, and N. Navab. Image-based tracking of the teeth for orthodontic augmented reality. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, pages 601–608, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[24] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of time-oriented data.* Springer Science & Business Media, 2011.

[25] M. F. Akay. Support vector machines combined with feature selection for breast cancer diagnosis. *Expert Systems with Applications*, 36(2):3240–3247, 2009.

[26] D. G. Aliaga, P. A. Rosen, and D. R. Bekins. Style grammars for interactive visualization of architecture. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):786–797, 2007.

[27] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. Awwal, and V. K. Asari. A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 8(3):292, 2019.

152

[28] A. Amirkhanov, A. Amirkhanov, M. Bernhard, Z. Toth, S. Stiller, A. Geier, E. Gröller, and G. Mistelbauer. WithTeeth: denture preview in augmented reality. *Vision, Modeling, and Visualization*, 2018.

[29] A. Amirkhanov, A. Amirkhanov, D. Salaberger, J. Kastner, M. E. Gröller, and C. Heinzl. Visual analysis of defects in glass fiber reinforced polymers for 4DCT interrupted in situ tests. *Computer Graphics Forum*, 35(3):201–210, 2016.

[30] A. Amirkhanov, M. Bernhard, A. Karimov, S. Stiller, A. Geier, M. E. Gröller, and G. Mistelbauer. Visual analytics in dental aesthetics. *Computer Graphics Forum*, 39(7):635–646, 2020.

[31] A. Amirkhanov, B. Fröhler, J. Kastner, E. Gröller, and C. Heinzl. InSpectr: Multi-modal exploration, visualization, and analysis of spectral data. *Computer Graphics Forum*, 33(3):91–100, 2014.

[32] A. Amirkhanov, C. Heinzl, C. Kuhn, J. Kastner, and M. E. Gröller. Fuzzy CT metrology. In *Spring Conference on Computer Graphics - SCCG '13*. ACM Press, 2013.

[33] A. Amirkhanov, C. Heinzl, M. Reiter, and E. Gröller. Visual optimality and stability analysis of 3DCT scan positions. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1477–1486, 2010.

[34] A. Amirkhanov, I. Kosiuk, P. Szmolyan, A. Amirkhanov, G. Mistelbauer, E. Gröller, and R. Raidou. Repository of ManyLands. `https://amirkhanov.net/manylands/`. Accessed: 18.06.2021.

[35] A. Amirkhanov, I. Kosiuk, P. Szmolyan, A. Amirkhanov, G. Mistelbauer, M. E. Gröller, and R. G. Raidou. ManyLands: A journey across 4D phase space of trajectories. *Computer Graphics Forum*, 38(7):191–202, 2019.

[36] G. Andrienko, N. Andrienko, P. Bak, D. Keim, and S. Wrobel. *Visual analytics of movement*. Springer Science & Business Media, 2013.

[37] M. Angelini, G. Santucci, H. Schumann, and H.-J. Schulz. A review and characterization of progressive visual analytics. In *Informatics*, volume 5. Multidisciplinary Digital Publishing Institute, 2018.

[38] D. Asimov. The Grand Tour: a tool for viewing multidimensional data. *SIAM journal on Scientific and Statistical Computing*, 6(1):128–143, 1985.

[39] B. M. Asl, S. K. Setarehdan, and M. Mohebbi. Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal. *Artificial Intelligence in Medicine*, 44(1):51–64, 2008.

[40] U. Ayachit, A. Bauer, B. Geveci, P. O'Leary, K. Moreland, N. Fabian, and J. Mauldin. Paraview catalyst: Enabling in situ data analysis and visualization. In *Proceedings of the First Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization*, pages 25–29, 2015.

[41] S. Bachthaler and D. Weiskopf. Continuous scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1428–1435, 2008.

[42] C. Bahlmann, B. Haasdonk, and H. Burkhardt. Online handwriting recognition with support vector machines - a kernel approach. In *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*. IEEE Comput. Soc, 2002.

[43] J.-P. Balabanian. Multi-aspect visualization: Going from linked views to integrated views. 2010.

[44] R. Ballester-Ripoll and R. Pajarola. Visualization of high-dimensional scalar functions using principal parameterizations. *ArXiv e-prints*, page arXiv:1809.03618, 2018.

[45] D. Baricevic, C. Lee, M. Turk, T. Höllerer, and D. A. Bowman. A hand-held AR magic lens with user-perspective rendering. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2012.

[46] A. C. Bauer, H. Abbasi, J. Ahrens, H. Childs, B. Geveci, S. Klasky, K. Moreland, P. O'Leary, V. Vishwanath, B. Whitlock, et al. In situ methods, infrastructures, and applications on high performance computing platforms. In *Computer Graphics Forum*, volume 35, pages 577–597. Wiley Online Library, 2016.

[47] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[48] B. B. Bederson and A. Boltman. Does animation help users build mental maps of spatial information? In *Proceedings of the Symposium on Information Visualization*, pages 28–35. IEEE, 1999.

[49] U. K. Bhowmik, D. Mandala, N. W. Hudyma, O. P. Kreidl, and A. Harris. Segmentation of cracks in X-ray CT images of tested macroporous plaster specimens. In *SOUTHEASTCON 2014, IEEE*, pages 1–8, 2014.

[50] C. Bichlmeier. *Immersive, interactive and contextual in-situ visualization for medical applications.* PhD thesis, Technische Universität München, 2010.

[51] C. Bichlmeier, S. M. Heining, M. Feuerstein, and N. Navab. The virtual mirror: A new interaction paradigm for augmented reality environments. *IEEE Transactions on Medical Imaging*, 28(9):1498–1510, 2009.

154

[52] C. Bichlmeier, T. Sielhorst, S. M. Heining, and N. Navab. Improving depth perception in medical AR. In *Proceedings of Bildverarbeitung für die Medizin*, pages 217–221. Springer Berlin Heidelberg, 2007.

[53] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 73–80, 1993.

[54] A. Bisler, U. Bockholt, and G. Voss. The virtual articulator—applying VR technologies to dentistry. In *Proceedings Sixth International Conference on Information Visualisation*, pages 600–602, 2002.

[55] T. Blum, V. Kleeberger, C. Bichlmeier, and N. Navab. mirracle: An augmented reality magic mirror system for anatomy education. In *Proceedings of IEEE Virtual Reality Workshops*, pages 115–116, 2012.

[56] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory - COLT '92*, pages 144–152. ACM Press, 1992.

[57] M. Bostock, V. Ogievetsky, and J. Heer. D³ data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[58] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[59] C. A. Brewer, G. W. Hatchard, and M. A. Harrower. Colorbrewer in print: a catalog of color schemes for maps. *Cartography and Geographic Information Science*, 30(1):5–32, 2003.

[60] A. Buja, J. A. McDonald, J. Michalak, and W. Stuetzle. Interactive data visualization using focusing and linking. In *Visualization*, pages 156–163. IEEE, 1991.

[61] C. A. Câmara. Aesthetics in orthodontics: Six horizontal smile lines. *Dental Press Journal of Orthodontics*, 15(1):118–131, 2010.

[62] C. Cao, D. Bradley, K. Zhou, and T. Beeler. Real-time high-fidelity facial performance capture. *ACM Trans. Graph.*, 34(4):46:1–46:9, 2015.

[63] C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Trans. Graph.*, 33(4):43:1–43:10, 2014.

[64] G. Cao, S. Ruan, Y. Peng, S. Huang, and N. Kwok. Large-complex-surface defect detection by hybrid gradient threshold segmentation and image registration. *IEEE Access*, 6:36235–36246, 2018.

[65] S. K. Card, J. D. Mackinlay, and B. Shneiderman. Using vision to think. In *Readings in information visualization*, pages 579–581. Morgan Kaufmann Publishers Inc., 1999.

[66] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75–94, 2003.

[67] J. R. Cebral and R. Lohner. Efficient simulation of blood flow past complex endovascular devices using an adaptive embedding technique. *IEEE transactions on medical imaging*, 24(4):468–476, 2005.

[68] M. Chabanas, V. Luboz, and Y. Payan. Patient specific finite element model of the face soft tissues for computer-assisted maxillofacial surgery. *Medical Image Analysis*, 7(2):131–151, 2003.

[69] G. Chandrashekar and F. Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[70] J. X. Chen, H. Wechsler, J. M. Pullen, Y. Zhu, and E. B. MacMahon. Knee surgery assistance: patient model construction, motion simulation, and biomechanical visualization. *IEEE Transactions on Biomedical Engineering*, 48(9):1042–1052, 2001.

[71] C. Chicone. *Ordinary differential equations with applications*, volume 34. Springer Science & Business Media, 2006.

[72] M. Chu, B. Dalal, A. Walendowski, and J. Begole. Countertop responsive mirror: supporting physical retail shopping for sellers, buyers and companions. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, pages 2533–2542, 2010.

[73] W. S. Cleveland. *Visualizing data.* Hobart Press, 1993.

[74] K. A. Cook and J. J. Thomas. Illuminating the path: The research and development agenda for visual analytics. 2005.

[75] F. L. Corbusier. *Le Corbusier Archive: Le Modulor and other buildings and projects, 1944–1945.* Le Corbusier Archive. Garland Pub., 1982.

[76] D. Cornel, A. Buttinger-Kreuzhuber, A. Konev, Z. Horvath, M. Wimmer, R. Heidrich, and J. Waser. Interactive visualization of flood and heavy rain simulations. *Computer Graphics Forum*, 38(3):25–39, 2019.

[77] C. Correa, P. Lindstrom, and P.-T. Bremer. Topological spines: A structure-preserving visual representation of scalar fields. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1842–1851, 2011.

[78] G. Courtois and T. D. Gilmore. Mutations in the NF-κB signaling pathway: implications for human disease. *Oncogene*, 25(51):6831–6843, 2006.

[79] H. S. M. Coxeter. *Regular polytopes.* Courier Corporation, 1973.

[80] Z. Cui, J. Kancherla, H. C. Bravo, and N. Elmqvist. Sherpa: Leveraging user attention for computational steering in visual analytics. In *2019 IEEE Visualization in Data Science (VDS)*, pages 48–57, 2019.

[81] C. Cullen. *Astronomy and Mathematics in Ancient China: The'Zhou Bi Suan Jing'*. Cambridge University Press, 2007.

[82] C. M. Daikuzono, F. M. Shimizu, A. Manzoli, A. Riul, M. H. O. Piazzetta, A. L. Gobbi, D. S. Correa, F. V. Paulovich, and O. N. Oliveira. Information visualization and feature selection methods applied to detect gliadin in gluten-containing foodstuff with a microfluidic electronic tongue. *ACS Applied Materials & Interfaces*, 9(23):19646–19652, 2017.

[83] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[84] S. Dalí. Crucifixion (corpus hypercubus), 1954. Oil on canvas. Metropolitan Museum of Art, New York.

[85] S. K. Datta, P. Morrow, and B. Scotney. *Facial Feature Extraction Using a 4D Stereo Camera System*, pages 209–218. Springer Berlin Heidelberg, 2012.

[86] E. de Llano, H. Miao, Y. Ahmadi, A. J. Wilson, M. Beeby, I. Viola, and I. Barisic. Adenita: interactive 3D modelling and visualization of dna nanostructures. *Nucleic acids research*, 48(15):8269–8275, 2020.

[87] W. E. Deming. *Out of the crisis*. Massachusetts Institute of Technology, Center for Advanced Engineering Study, 1986.

[88] P. V. der Geld, P. Oosterveld, G. V. Heck, and A. M. Kuijpers-Jagtman. Smile attractiveness. *The Angle Orthodontist*, 77(5):759–765, 2007.

[89] P. V. der Geld, P. Oosterveld, J. Schols, and A. M. Kuijpers-Jagtman. Smile line assessment comparing quantitative measurement and visual estimation. *American Journal of Orthodontics and Dentofacial Orthopedics*, 139(2):174–180, 2011.

[90] G. Di Battista and R. Tamassia. Algorithms for plane representations of acyclic digraphs. *Theoretical Computer Science*, 61(2-3):175–198, 1988.

[91] C. Dick, J. Georgii, R. Burgkart, and R. Westermann. Stress tensor field visualization for implant planning in orthopedics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1399–1406, 2009.

[92] S. R. dos Santos and K. W. Brodlie. Visualizing and investigating multidimensional functions. In *Proceedings of the Symposium on Data Visualisation*, pages 173–182. Eurographics Association, 2002.

[93] F. Dumortier and R. Roussarie. *Canard cycles and center manifolds*, volume 577. American Mathematical Society, 1996.

[94] S. Dutta, C.-M. Chen, G. Heinlein, H.-W. Shen, and J.-P. Chen. In situ distribution guided analysis and visualization of transonic jet engine simulations. *IEEE transactions on visualization and computer graphics*, 23(1):811–820, 2016.

[95] O. Ecabert, J. Peters, H. Schramm, C. Lorenz, J. von Berg, M. J. Walker, M. Vembar, M. E. Olszewski, K. Subramanyan, G. Lavi, and J. Weese. Automatic model-based segmentation of the heart in CT images. *IEEE Transactions on Medical Imaging*, 27(9):1189–1201, 2008.

[96] S. Eichelbaum, M. Hlawitschka, and G. Scheuermann. Lineao–improved three-dimensional line rendering. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):433–445, 2013.

[97] Y. Eisenthal, G. Dror, and E. Ruppin. Facial attractiveness: Beauty and the machine. *Neural Computation*, 18(1):119–142, 2006.

[98] K. El Emam, W. Melo, and J. C. Machado. The prediction of faulty classes using object-oriented design metrics. *Journal of systems and software*, 56(1):63–75, 2001.

[99] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1539–1148, 2008.

[100] K. Engel, M. Hadwiger, J. M. Kniss, A. E. Lefohn, C. R. Salama, and D. Weiskopf. Real-time volume graphics. In *Proceedings of the conference on SIGGRAPH 2004 course notes - GRAPH '04*, pages 29–es. ACM Press, 2004.

[101] N. Eveno, A. Caplier, and P.-Y. Coulon. Accurate and quasi-automatic lip tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5):706–715, 2004.

[102] M. H. Everts, H. Bekker, J. B. T. M. Roerdink, and T. Isenberg. Depth-dependent halos: Illustrative rendering of dense line data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1299–1306, 2009.

[103] Y.-Y. Fan, S. Liu, B. Li, Z. Guo, A. Samal, J. Wan, and S. Z. Li. Label distribution-based facial attractiveness computation by deep residual learning. *IEEE Transactions on Multimedia*, 20(8):2196–2208, 2017.

[104] S. Fang, T. Biddlecome, and M. Tuceryan. Image-based transfer function design for data exploration in volume visualization. In *Proceedings Visualization '98*. IEEE, 1998.

[105] S. K. Feiner and C. Beshers. Worlds within worlds: Metaphors for exploring n-dimensional virtual worlds. In *Proceedings of the 3rd annual ACM SIGGRAPH Symposium on User Interface Software and Technology*, pages 76–83. ACM, 1990.

[106] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[107] N. E. Fenton and M. Neil. A critique of software defect prediction models. *IEEE Transactions on software engineering*, 25(5):675–689, 1999.

[108] N. F. Fernandez, G. W. Gundersen, A. Rahman, M. L. Grimes, K. Rikova, P. Hornbeck, and A. Ma'ayan. Clustergrammer, a web-based heatmap visualization and analysis tool for high-dimensional biological data. *Scientific Data*, 4(1):1–12, 2017.

[109] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann. Visual analysis of spatial variability and global correlations in ensembles of iso-contours. In *Computer Graphics Forum*, volume 35, pages 221–230, 2016.

[110] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann. Time-hierarchical clustering and visualization of weather forecast ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):831–840, 2017.

[111] M. A. Fisherkeller, J. H. Friedman, and J. W. Tukey. Prim-9: An interactive multidimensional data display and analysis system. *Dynamic Graphics for Statistics*, pages 91–109, 1988.

[112] D. Freedman, R. Radke, T. Zhang, Y. Jeong, D. Lovelock, and G. Chen. Model-based segmentation of medical imagery by matching distributions. *IEEE Transactions on Medical Imaging*, 24(3):281–292, 2005.

[113] K. Friedrich. Fractographic analysis of polymer composites. In *Composite Materials Series*, volume 6 of *Composite Materials Series*, pages 425–487. Elsevier, 1989.

[114] L. Fritz, M. Hadwiger, G. Geier, G. Pittino, and M. E. Gröller. A visual approach to efficient analysis and quantification of ductile iron and reinforced sprayed concrete. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1343–1350, 2009.

[115] B. Fröhler, T. Möller, and C. Heinzl. GEMSe: Visualization-guided exploration of multi-channel segmentation algorithms. *Computer Graphics Forum*, 35(3):191–200, 2016.

[116] B. Fröhler, J. Weissenböck, M. Schiwarth, J. Kastner, and C. Heinzl. open_ia: A tool for processing and visual analysis of industrial computed tomography datasets. *Journal of Open Source Software*, 4(35):1185, 2019.

[117] K. Furmanová, L. P. Muren, O. Casares-Magaz, V. Moiseenko, J. P. Einck, S. Pil-skog, and R. G. Raidou. PREVIS: predictive visual analytics of anatomical variability for radiotherapy decision support. *Computers & Graphics*, 97:126–138, 2021.

[118] P. Garrido, M. Zollhöfer, C. Wu, D. Bradley, P. Pérez, T. Beeler, and C. Theobalt. Corrective 3D reconstruction of lips from monocular video. *ACM Trans. Graph.*, 35(6):219:1–219:11, 2016.

[119] S. Gerber, P.-T. Bremer, V. Pascucci, and R. Whitaker. Visual exploration of high dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1271–1280, 2010.

[120] E. Gladilin, S. Zachow, P. Deuflhard, and H.-C. Hege. Anatomy-and physics-based facial animation for craniofacial surgery simulations. *Medical and Biological Engineering and Computing*, 42(2):167–170, 2004.

[121] M. Gleicher. A framework for considering comprehensibility in modeling. *Big Data*, 4(2):75–88, 2016.

[122] M. Gleicher. Considerations for visualizing comparison. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):413–423, 2018.

[123] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011.

[124] A. Goldbeter. A model for the dynamics of bipolar disorders. *Progress in biophysics and molecular biology*, 105(1-2):119–127, 2011.

[125] L. Grady. Random walks for image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11):1768–1783, 2006.

[126] E. Gröller, R. Wegenkittl, A. Milik, A. Prskawetz, G. Feichtinger, and W. C. Sanderson. The geometry of wonderland. *Chaos, Solitons & Fractals*, 7(12):1989–2006, 1996.

[127] S. Grottel, J. Heinrich, D. Weiskopf, and S. Gumhold. Visual analysis of trajectories in multi-dimensional state spaces. In *Computer Graphics Forum*, volume 33, pages 310–321, 2014.

[128] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields.* Springer, New York, 1983.

[129] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making faces. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pages 55–66, 1998.

160

[130] G. Guo, S. Z. Li, and K. Chan. Face recognition by support vector machines. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition.* IEEE Comput. Soc, 2000.

[131] H. Guo, Z. Wang, B. Yu, H. Zhao, and X. Yuan. Tripvista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection. In *2011 IEEE Pacific Visualization Symposium*, pages 163–170. IEEE, 2011.

[132] M. Hadwiger, F. Laura, C. Rezk-Salama, T. Höllt, G. Geier, and T. Pabel. Interactive volume exploration for feature detection and quantification in industrial CT data. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1507–1514, 2008.

[133] M. Haidacher, D. Patel, S. Bruckner, A. Kanitsar, and M. E. Groller. Volume visualization based on statistical transfer-function spaces. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 2010.

[134] K. W. Hall, P. G. Kusalik, and S. Carpendale. Profile contour plots: Alternative projections of 3D free energy surfaces. *In IEEE Conference on Scientific Visualization (SciVis) – Poster*, 2014.

[135] C. Hansen, J. Wieferich, F. Ritter, C. Rieder, and H.-O. Peitgen. Illustrative visualization of 3D planning models for augmented reality in liver surgery. *International Journal of Computer Assisted Radiology and Surgery*, 5(2):133–141, 2010.

[136] A. J. Hanson, K. I. Ishkov, and J. H. Ma. Meshview: Visualizing the fourth dimension. *Technical report. Indiana University.*, 1999.

[137] S. Havre, B. Hetzler, and L. Nowell. Themeriver: visualizing theme changes over time. In *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*, pages 115–123, 2000.

[138] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. Themeriver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[139] J. Heer and G. G. Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247, 2007.

[140] J. Heinrich and D. Weiskopf. Continuous parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, pages 1531–1538, 2009.

[141] J. Heinrich and D. Weiskopf. State of the art of parallel coordinates. In *Eurographics (STARs)*, pages 95–116. The Eurographics Association, 2012.

[142] C. Heinzl, J. Kastner, and M. E. Gröller. Surface Extraction from Multi-Material Components for Metrology using Dual Energy CT. Technical Report TR-186-2-07-06, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2007.

[143] C. Heinzl and S. Stappen. STAR: Visual computing in materials science. *Computer Graphics Forum*, 36(3):647–666, 2017.

[144] G. E. Hinton and S. Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15:857–864, 2002.

[145] L. Höfel, M. Lange, and T. Jacobsen. Beauty and the teeth: perception of tooth color and its influence on the overall judgment of facial attractiveness. *International Journal of Periodontics & Restorative Dentistry*, 27(4), 2007.

[146] R. Hooke and T. A. Jeeves. "direct search" solution of numerical and statistical problems. *Journal of the ACM (JACM)*, 8(2):212–229, 1961.

[147] Z. Horvath, R. A. P. Perdigao, J. Waser, D. Cornel, A. Konev, and G. Blöschl. Kepler shuffle for real-world flood simulations on GPUs. *The International Journal of High Performance Computing Applications*, 30(4):379–395, 2016.

[148] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[149] H. Huang, J. Chai, X. Tong, and H.-T. Wu. Leveraging motion capture and 3D scanning for high-fidelity facial performance acquisition. *ACM Trans. Graph.*, 30(4):74:1–74:10, 2011.

[150] R. Huang, K.-L. Ma, P. McCormick, and W. Ward. Visualizing industrial CT volume data for nondestructive testing applications. In *IEEE Visualization 2003 Conference*, pages 547–554, 2003.

[151] F. Hutmacher. Why is there so much more research on vision than on any other sensory modality? *Frontiers in psychology*, 10:2246, 2019.

[152] A. Inselberg. The plane with parallel coordinates. *The visual computer*, 1(2):69–91, 1985.

[153] T. Isenberg, P. Isenberg, J. Chen, M. Sedlmair, and T. Möller. A systematic review on the practice of evaluating visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2818–2827, 2013.

[154] Quality management systems — fundamentals and vocabulary. Standard, International Organization for Standardization, Geneva, CH, 2015.

[155] Quality management systems — requirements. Standard, International Organization for Standardization, Geneva, CH, 2015.

[156] S. Jayaraman and C. North. A radial focus+context visualization for multi-dimensional functions. In *Visualization*, pages 443–450. IEEE, 2002.

[157] R. Jianu, C. Demiralp, and D. Laidlaw. Exploring 3D DTI fiber tracts with linked 2D representations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1449–1456, 2009.

[158] J. Jo, S. Lyi, B. Lee, and J. Seo. ProReveal: Progressive visual analytics with safeguards. *IEEE Transactions on Visualization and Computer Graphics*, 27(7):3109–3122, 2021.

[159] C. D. Johnston, D. J. Burden, and M. R. Stevenson. The influence of dental to facial midline discrepancies on dental attractiveness ratings. *European Journal of Orthodontics*, 21(5):517–522, 1999.

[160] K. Jung, S. Jung, I. Hwang, T. Kim, and M. Chang. Registration of dental tomographic volume data and scan surface data using dynamic segmentation. *Applied Sciences*, 8(10):1762, 2018.

[161] A. Jurcik, K. Furmanova, J. Byska, V. Vonasek, O. Vavra, P. Ulbrich, H. Hauser, and B. Kozlikova. Visual analysis of ligand trajectories in molecular dynamics. In *2019 IEEE Pacific Visualization Symposium (PacificVis)*, pages 212–221. IEEE, 2019.

[162] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[163] N. Kanopoulos, N. Vasanthavada, and R. L. Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2):358–367, 1988.

[164] O. Karpenko, W. Li, N. Mitra, and M. Agrawala. Exploded view diagrams of mathematical surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1311–1318, 2010.

[165] E. Keeve, S. Girod, P. Pfeifle, and B. Girod. Anatomy-based facial tissue modeling using the finite element method. In *Proceedings of Seventh Annual IEEE Visualization*, pages 21–28, 1996.

[166] Y. Kim, K. Lee, and W. Kim. 3D virtual simulator for breast plastic surgery. *Computer Animation and Virtual Worlds*, 19(3-4):515–526, 2008.

[167] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[168] E. Klipp, W. Liebermeister, C. Wierling, A. Kowald, and R. Herwig. *Systems biology: a textbook*. John Wiley & Sons, 2016.

[169] P. R. Koralakunte and M. Aljanakh. The role of virtual articulator in prosthetic and restorative dentistry. *Journal of clinical and diagnostic research: JCDR*, 8(7):ZE25, 2014.

[170] I. Kosiuk and P. Szmolyan. Geometric desingularization in slow-fast systems with application to the glycolytic oscillations model. *AIP Conference Proceedings*, 1281(1):235–238, 2010.

[171] I. Kosiuk and P. Szmolyan. Scaling in singular perturbation problems: blowing up a relaxation oscillator. *SIAM Journal on Applied Dynamical Systems*, 10(4):1307–1343, 2011.

[172] I. Kosiuk and P. Szmolyan. Geometric analysis of the goldbeter minimal model for the embryonic cell cycle. *Journal of Mathematical Biology*, 72(5):1337–1368, 2016.

[173] A. Kratz, M. Schoeneich, V. Zobel, B. Burgeth, G. Scheuermann, I. Hotz, and M. Stommel. Tensor visualization driven mechanical component design. In *2014 IEEE Pacific Visualization Symposium*, pages 145–152, 2014.

[174] P. R. Krekel, C. P. Botha, E. R. Valstar, P. W. de Bruin, P. M. Rozing, and F. H. Post. Interactive simulation and comparative visualisation of the bone-determined range of motion of the human shoulder. In *SimVis*, pages 275–288, 2006.

[175] S. Krishna, M. H. Jensen, and K. Sneppen. Minimal model of spiky oscillations in NF-κB signaling. *Proceedings of the National Academy of Sciences*, 103(29):10840–10845, 2006.

[176] C. Kühn and P. Szmolyan. Multiscale geometry of the olsen model and non-classical relaxation oscillations. *Journal of Nonlinear Science*, 25(3):583–629, 2015.

[177] M. R. S. Kulenovic and O. Merino. *Discrete dynamical systems and difference equations with Mathematica.* Chapman and Hall/CRC, 2002.

[178] O. Kutter, A. Aichert, C. Bichlmeier, J. Traub, E. Euler, and N. Navab. Real-time volume rendering for high quality visualization in augmented reality. In *Proceedings of Workshop on Augmented environments for Medical Imaging including Augmented Reality in Computer-aided Surgery*, pages 104–113, 2008.

[179] M. Lancelle, R. Mörzinger, N. Degen, G. Sörös, and B. Nemanja. Dental design transfer, 2019. US Patent App. 16/279,166.

[180] A. Landstrom and M. J. Thurley. Morphology-based crack detection for steel slabs. *IEEE Journal of Selected Topics in Signal Processing*, 6(7):866–875, 2012.

[181] T.-Y. Lee, C.-H. Lin, and H.-Y. Lin. Computer-aided prototype system for nose surgery. *IEEE Transactions on Information Technology in Biomedicine*, 5(4):271–278, 2001.

164

[182] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.

[183] E. I. Levin. Dental esthetics and the golden proportion. *The Journal of Prosthetic Dentistry*, 40(3):244–252, 1978.

[184] X. Liu, R. Wang, C.-F. Chen, M. Yin, H. Peng, S. Ng, and X. Li. Face beautification: Beyond makeup transfer. *arXiv preprint arXiv:1912.03630*, 2019.

[185] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li. Deepcrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338:139–153, 2019.

[186] P. Ljung, J. Krüger, E. Groller, M. Hadwiger, C. D. Hansen, and A. Ynnerman. State of the art in transfer functions for direct volume rendering. In *Computer Graphics Forum*, volume 35, pages 669–691. Wiley Online Library, 2016.

[187] R. E. Lombardi. The principles of visual perception and their clinical application to denture esthetics. *The Journal of Prosthetic Dentistry*, 29(4):358–382, 1973.

[188] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.

[189] S. Lynch. *Dynamical systems with applications using Matlab®*. Springer, 2014.

[190] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[191] A. Majid, S. G. Roberts, L. Cilissen, K. Emmorey, B. Nicodemus, L. O'Grady, B. Woll, B. LeLan, H. De Sousa, B. L. Cansler, et al. Differential coding of perception in the world's languages. *Proceedings of the National Academy of Sciences*, 115(45):11369–11376, 2018.

[192] M. M. Malik, C. Heinzl, and M. E. Gröller. Comparative visualization for parameter studies of dataset series. *IEEE Transactions on Visualization and Computer Graphics*, 16(5):829–840, 2010.

[193] F. Mangano, A. Gandolfi, G. Luongo, and S. Logozzo. Intraoral scanners in dentistry: a review of the current literature. *BMC Oral Health*, 17(1), 2017.

[194] K. Matkovic, H. Hauser, R. Sainitzer, and M. E. Gröller. Process visualization with levels of detail. In *IEEE Symposium on Information Visualization.*, pages 67–70. IEEE, 2002.

[195] B. H. McCormick. Visualization in scientific computing. *ACM SIGBIO Newsletter*, 10(1):15–21, 1988.

[196] M. Meehan, M. Teschner, and S. Girod. Three-dimensional simulation and prediction of craniofacial surgery. *Orthodontics & craniofacial research*, 6:102–107, 2003.

[197] S. Mei, Y. Wang, and G. Wen. Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model. *Sensors*, 18(4):1064, 2018.

[198] M. Meng, P. Fallavollita, T. Blum, U. Eck, C. Sandor, S. Weidert, J. Waschke, and N. Navab. Kinect for interactive AR anatomy learning. In *Proceedings of IEEE Symposium on Mixed and Augmented Reality*, pages 277–278, 2013.

[199] J. Mercier-Ganady, F. Lotte, E. Loup-Escande, M. Marchal, and A. Lécuyer. The mind-mirror: See your brain in action in your head using EEG and augmented reality. In *Proceedings of IEEE Virtual Reality*, pages 33–38, 2014.

[200] G. P. Meyer, S. Gupta, I. Frosio, D. Reddy, and J. Kautz. Robust model-based 3D head pose estimation. In *Proceedings of IEEE Conference on Computer Vision*, pages 3649–3657, 2015.

[201] H. Miao, E. De Llano, T. Isenberg, M. E. Gröller, I. Barišić, and I. Viola. Dimsum: Dimension and scale unifying map for visual abstraction of dna origami structures. In *Computer Graphics Forum*, volume 37, pages 403–413, 2018.

[202] D. Mintz, T. Fitz-Simons, and M. Wayland. Tracking air quality trends with sas/graph. In *Proceedings of the 22nd Annual SAS User Group International Conference (SUGI'97)*, pages 807–812, 1997.

[203] M. Mirzargar, R. T. Whitaker, and R. M. Kirby. Curve boxplot: Generalization of boxplot for ensembles of curves. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2654–2663, 2014.

[204] N. Mitsuhashi, K. Fujieda, T. Tamura, S. Kawamoto, T. Takagi, and K. Okubo. BodyParts3D: 3D structure database for anatomical concepts. *Nucleic Acids Research*, 37(suppl 1):D782–D785, 2009.

[205] T. Munzner. *Visualization analysis and design*. CRC press, 2014.

[206] N. Nadjmi, W. Mollemans, A. Daelemans, G. V. Hemelen, F. Schutyser, and S. Bergé. Virtual occlusion in planning orthognathic surgical procedures. *International Journal of Oral and Maxillofacial Surgery*, 39(5):457–462, 2010.

[207] N. Navab, M. Feuerstein, and C. Bichlmeier. Laparoscopic virtual mirror new interaction paradigm for monitor based augmented reality. In *Proceedings of IEEE Virtual Reality Conference*, pages 43–50, 2007.

[208] M. Neugebauer, G. Janiga, O. Beuing, M. Skalej, and B. Preim. Anatomy-guided multi-level exploration of blood flow in cerebral aneurysms. In *Computer Graphics Forum*, volume 30, pages 1041–1050. Wiley Online Library, 2011.

[209] T. Nocke, M. Flechsig, and U. Bohm. Visual exploration and evaluation of climate-related simulation data. In *2007 Winter Simulation Conference*, pages 703–711. IEEE, 2007.

166

[210] B. Nouanesengsy, S.-C. Seok, H.-W. Shen, and V. J. Vieland. Using projection and 2D plots to visually reveal genetic mechanisms of complex human disorders. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 171–178. IEEE, 2009.

[211] S. Oeltze, H. Doleisch, H. Hauser, P. Muigg, and B. Preim. Interactive visual analysis of perfusion data. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1392–1399, 2007.

[212] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9:62–66, 1979.

[213] O. Paetsch, D. Baum, S. Prohaska, K. Ehrig, D. Meinel, and G. Ebell. 3D corrosion detection in time-dependent CT images of concrete. In *Digital Industrial Radiology and Computed Tomography*, 2015.

[214] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[215] G. Pedersen. Stanislav Petrov World Hero. *Fellowship*, 71(7-8):9, 2005.

[216] D. I. Perrett, D. M. Burt, I. S. Penton-Voak, K. J. Lee, D. A. Rowland, and R. Edwards. Symmetry and human facial attractiveness. *Evolution and human behavior*, 20(5):295–307, 1999.

[217] D. I. Perrett, K. J. Lee, I. Penton-Voak, D. Rowland, S. Yoshikawa, D. M. Burt, S. P. Henzi, D. L. Castles, and S. Akamatsu. Effects of sexual dimorphism on facial attractiveness. *Nature*, 394(6696):884–887, 1998.

[218] M. Piotraschke and V. Blanz. Automated 3D face reconstruction from multiple images using quality measures. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3418–3427, 2016.

[219] H. Piringer, W. Berger, and J. Krasser. Hypermoval: Interactive visual validation of regression models for real-time simulation. In *Computer Graphics Forum*, volume 29, pages 983–992, 2010.

[220] S. Plokhy. *Chernobyl: History of a tragedy*. Penguin UK, 2018.

[221] B. Preim and C. P. Botha. *Visual computing for medicine: theory, algorithms, and applications*. Newnes, 2013.

[222] A. J. Pretorius, M.-A. Bray, A. E. Carpenter, and R. A. Ruddle. Visualization of parameter space for image analysis. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2402–2411, 2011.

167

[223] A. S. M. M. Rahman, T. T. Tran, S. A. Hossain, and A. E. Saddik. Augmented rendering of makeup features in a smart interactive mirror system for decision support in cosmetic products selection. In *Proceedings of IEEE/ACM Symposium on Distributed Simulation and Real Time Applications*, pages 203–206, 2010.

[224] R. G. Raidou, O. Casares-Magaz, L. P. Muren, U. A. Van der Heide, J. Rørvik, M. Breeuwer, and A. Vilanova. Visual analysis of tumor control models for prediction of radiotherapy response. In *Computer Graphics Forum*, volume 35, pages 231–240. Wiley Online Library, 2016.

[225] R. G. Raidou, M. Eisemann, M. Breeuwer, E. Eisemann, and A. Vilanova. Orientation-enhanced parallel coordinate plots. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):589–598, 2016.

[226] R. G. Raidou, U. A. van der Heide, C. V. Dinh, G. Ghobadi, J. F. Kallehauge, M. Breeuwer, and A. Vilanova. Visual analytics for the exploration of tumor tissue characterization. In *Computer Graphics Forum*, volume 34, pages 11–20. Wiley Online Library, 2015.

[227] G. Rao, A. Amirkhanov, D. Salaberger, C. Heinzl, and J. Kastner. Damage characterization in SFRP using X-ray computed tomography after application of incremental and interrupted in situ quasi static tensile loading. *Industrial Computed Tomography, Leuven*, 2017.

[228] A. Reh, A. Amirkhanov, J. Kastner, E. Gröller, and C. Heinzl. Fuzzy feature tracking: Visual analysis of industrial 4D-XCT data. *Computers & Graphics*, 53:177–184, 2015.

[229] A. Reh, C. Gusenbauer, J. Kastner, M. E. Gröller, and C. Heinzl. MObjects— a novel method for the visualization and interactive exploration of defects in industrial XCT data. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2906–2915, 2013.

[230] A. Reh, B. Plank, J. Kastner, E. Gröller, and C. Heinzl. Porosity maps - interactive exploration and visual analysis of porosity in carbon fiber reinforced polymers. *Computer Graphics Forum*, 31:1185–1194, 2012.

[231] F. Reinders, F. H. Post, and H. J. W. Spoelder. Visualization of time-dependent data with feature tracking and event detection. *The Visual Computer*, 17(1):55–71, 2001.

[232] P. Rhienmora, K. Gajananan, P. Haddawy, M. N. Dailey, and S. Suebnukarn. Augmented reality haptics system for dental surgical skills training. In *Proceedings of ACM Symposium on Virtual Reality Software and Technology*, pages 97–98, 2010.

[233] E. Richardson, M. Sela, and R. Kimmel. 3D face reconstruction by learning from synthetic data. *CoRR*, abs/1609.04387, 2016.

168

[234] M. Rivi, L. Calori, G. Muscianisi, and V. Slavnic. In-situ visualization: State-of-the-art and some use cases. *PRACE White Paper*, pages 1–18, 2012.

[235] J. C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2007)*. IEEE, 2007.

[236] G. G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1325–1332, 2008.

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

[238] H. Rolland, N. Saintier, N. Lenoir, A. King, and G. Robert. Fatigue mechanisms description in short glass fibre reinforced thermoplastic by microtomographic observations. In *Proc. 20th International Conference on Composite Materials*, 2015.

[239] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: database and results. *Image and Vision Computing*, 47:3–18, 2016.

[240] T. Saito, H. N. Miyamura, M. Yamamoto, H. Saito, Y. Hoshiya, and T. Kaseda. Two-tone pseudo coloring: Compact visualization for one-dimensional data. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 173–180. IEEE, 2005.

[241] D. Salaberger, K. A. Kannappan, J. Kastner, J. Reussner, and T. Auinger. Evaluation of computed tomography data from fibre reinforced polymers to determine fibre length distribution. *International Polymer Processing*, 26(3):283–291, 2011.

[242] Z. Salah, B. Preim, A. Samii, R. Fahlbusch, and G. Rose. Enhanced intraoperative visualization for brain surgery: A prototypic simulated scenario. In *Proceedings of CURAC*, pages 125–130, 2010.

[243] C. R. Salama, M. Keller, and P. Kohlmann. High-level user interfaces for transfer function design with semantics. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1021–1028, 2006.

[244] H. Samet and M. Tamminen. Efficient component labeling of images of arbitrary dimension represented by linear bintrees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):579–586, 1988.

[245] S. Schkolne, H. Ishii, and P. Schröder. Immersive design of dna molecules with a tangible interface. In *IEEE Visualization 2004*, pages 227–234. IEEE, 2004.

[246] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic imaging*, 13(1):146–166, 2004.

[247] D. Silver and X. Wang. Tracking and visualizing turbulent 3D features. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):129–141, 1997.

[248] D. Sommerville. *Introduction to the Geometry of N Dimensions*. E.P.Dutton, 1929.

[249] J. Sorger, P. Mindek, P. Rautek, E. Gröller, G. Johnson, and I. Viola. Metamorphers: storytelling templates for illustrative animated transitions in molecular visualization. In *Proceedings of the 33rd Spring Conference on Computer Graphics*, pages 1–10. ACM, 2017.

[250] J. Sorger, M. Waldner, W. Knecht, and A. Arleo. Immersive analytics of large dynamic networks via overview and detail navigation. *arXiv preprint arXiv:1910.06825*, 2019.

[251] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Symposium on Geometry Processing*. The Eurographics Association, 2004.

[252] M. Spindler, S. Stellmach, and R. Dachselt. PaperLens. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '09*. ACM Press, 2009.

[253] P. Stefan, P. Wucherer, Y. Oyamada, M. Ma, A. Schoch, M. Kanegae, N. Shimizu, T. Kodera, S. Cahier, M. Weigl, M. Sugimoto, P. Fallavollita, H. Saito, and N. Navab. An AR edutainment system supporting bone anatomy learning. In *Proceedings of IEEE Virtual Reality*, pages 113–114, 2014.

[254] P.-A. Steinmassl, F. Klaunzer, O. Steinmassl, H. Dumfahrt, and I. Grunert. Evaluation of currently available cad/cam denture systems. *Int J Prosthodont*, 30(2):116–22, 2017.

[255] F. J. Swetz and V. J. Katz. Mathematical Treasures—Zhoubi suanjing. https://www.maa.org/press/periodicals/convergence/mathematical-treasures-zhoubi-suanjing. Accessed: 11.07.2021.

[256] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of RGB videos. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2387–2395, 2016.

[257] R. Thornhill and S. W. Gangestad. Facial attractiveness. *Trends in Cognitive Sciences*, 3(12):452–460, 1999.

[258] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann. A survey on interactive lenses in visualization. In *EuroVis (STARs)*, 2014.

170

[259] T. Torsney-Weir, T. Möller, M. Sedlmair, and R. M. Kirby. Hypersliceplorer: Interactive visualization of shapes in multiple dimensions. In *Computer Graphics Forum*, volume 37, pages 229–240, 2018.

[260] T. Torsney-Weir, A. Saad, T. Moller, H.-C. Hege, B. Weber, J.-M. Verbavatz, and S. Bergner. Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1892–1901, 2011.

[261] T. Torsney-Weir, M. Sedlmair, and T. Möller. Sliceplorer: 1D slices for multi-dimensional continuous functions. In *Computer Graphics Forum*, volume 36, pages 167–177, 2017.

[262] B. Tversky, J. B. Morrison, and M. Betrancourt. Animation: can it facilitate? *International Journal of Human-Computer Studies*, 57(4):247–262, 2002.

[263] P. Van der Geld, P. Oosterveld, G. Van Heck, and A. M. Kuijpers-Jagtman. Smile attractiveness: self-perception and influence on personality. *The Angle Orthodontist*, 77(5):759–765, 2007.

[264] L. Van der Maaten, E. Postma, and J. Van den Herik. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71):13, 2009.

[265] E. Van Groesen and J. Molenaar. *Continuum modeling in the physical sciences.* SIAM Mathematical Modeling and Computation, 2007.

[266] R. Van Pelt, J. O. Bescós, M. Breeuwer, R. E. Clough, M. E. Gröller, B. ter Haar Romenij, and A. Vilanova. Exploration of 4D MRI blood flow using stylistic visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1339–1347, 2010.

[267] J. J. van Wijk and R. van Liere. HyperSlice. In *Visualization*, pages 119–125. IEEE, IEEE Comput. Soc. Press, 1993.

[268] D. Vaughan. *The Challenger launch decision: Risky technology, culture, and deviance at NASA.* University of Chicago press, 1996.

[269] F. B. Viegas, M. Wattenberg, F. Van Ham, J. Kriss, and M. McKeon. Manyeyes: a site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, 2007.

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

[271] Z. Wakaf and H. A. Jalab. Defect detection based on extreme edge of defective region histogram. *Journal of King Saud University - Computer and Information Sciences*, 30(1):33–40, 2018.

[272] J. Wang, X. Liu, H. Shen, and G. Lin. Multi-resolution climate ensemble parameter analysis with nested parallel coordinates plots. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):81–90, 2017.

[273] J. Wang, H. Suenaga, K. Hoshi, L. Yang, E. Kobayashi, I. Sakuma, and H. Liao. Augmented reality navigation with automatic marker-free image registration using 3-D image overlay for dental surgery. *IEEE Transactions on Biomedical Engineering*, 61(4):1295–1304, 2014.

[274] J. Wang, H. Suenaga, L. Yang, E. Kobayashi, and I. Sakuma. Video see-through augmented reality for oral and maxillofacial surgery. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 13(2):e1754, 2017.

[275] Y. Wang, X. Huang, C.-S. Lee, S. Zhang, Z. Li, D. Samaras, D. Metaxas, A. Elgammal, and P. Huang. High resolution acquisition, learning and transfer of dynamic 3-D facial expressions. *Computer Graphics Forum*, 23(3):677–686, 2004.

[276] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 110–119. ACM, 2000.

[277] R. Wegenkittl, H. Löffelmann, and E. Gröller. Visualizing the behavior of higher dimensional dynamical systems. In *Proceedings. Visualization '97*, pages 119–125. IEEE Computer Society Press, IEEE, 1997.

[278] T. Weise, H. Li, L. Van Gool, and M. Pauly. Face/off: Live facial puppetry. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 7–16, 2009.

[279] J. Weissenböck, A. Amirkhanov, E. Gröller, J. Kastner, and C. Heinzl. Porosity-Analyzer: Visual analysis and evaluation of segmentation pipelines to determine the porosity in fiber-reinforced polymers. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2016.

[280] J. Weissenböck, A. Amirkhanov, W. Li, A. Reh, A. Amirkhanov, M. E. Gröller, J. Kastner, and C. Heinzl. Fiberscout: An interactive tool for exploring and analyzing fiber reinforced polymers. In *Proceedings of 2014 IEEE Pacific Visualization Symposium (PacificVis) (2014)*, pages 153–160. IEEE Computer Society, 2014.

[281] J. Weissenböck, B. Fröhler, E. Gröller, J. Kastner, and C. Heinzl. Dynamic volume lines: Visual comparison of 3D volumes through space-filling curves. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):1040–1049, 2019.

[282] R. T. Whitaker, M. Mirzargar, and R. M. Kirby. Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2713–2722, 2013.

172

[283] W. Widanagamaachchi, C. Christensen, V. Pascucci, and P.-T. Bremer. Interactive exploration of large-scale time-varying data using dynamic tracking graphs. In *Large Data Analysis and Visualization (LDAV), 2012 IEEE Symposium on*, pages 9–17, 2012.

[284] A. Wilhelm. Linked views for visual exploration. In *Handbook of Data Visualization*, pages 199–215. Springer Berlin Heidelberg, 2008.

[285] L. Williams. Performance-driven facial animation. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pages 235–242, 1990.

[286] G. Wills. Linked data views. In *Handbook of data visualization*, pages 217–241. Springer, 2008.

[287] J. Woodring, M. Petersen, A. Schmeißer, J. Patchett, J. Ahrens, and H. Hagen. In situ eddy analysis in a high-resolution ocean climate model. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):857–866, 2015.

[288] C. Wu, D. Bradley, P. Garrido, M. Zollhöfer, C. Theobalt, M. Gross, and T. Beeler. Model-Based Teeth Reconstruction. *ACM Trans. Graph.*, 35(6):220:1–220:13, 2016.

[289] J. Xu, L. Jin, L. Liang, Z. Feng, D. Xie, and H. Mao. Facial attractiveness prediction using psychologically inspired convolutional neural network (pi-cnn). In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1657–1661. IEEE, 2017.

[290] C. Xue, F. Li, T. He, G.-P. Liu, Y. Li, and X. Zhang. Classification of real and pseudo microrna precursors using local structure-sequence features and support vector machine. *BMC Bioinformatics*, 6(1):310, 2005.

[291] T. Yamaguchi, S. Nakamura, and S. Hashimoto. An efficient crack detection method using percolation-based image processing. In *2008 3rd IEEE Conference on Industrial Electronics and Applications*, pages 1875–1880. IEEE, 2008.

[292] W. Yang, N. Marshak, D. Sỳkora, S. Ramalingam, and L. Kavan. Building anatomically realistic jaw kinematics model from data. *The Visual Computer*, 35(6-8):1105–1118, 2019.

[293] S. Zachow, H.-C. Hege, and P. Deuflhard. Computer assisted planning in cranio-maxillofacial surgery. *CIT*, 14:53–64, 2006.

[294] K. Zakopcanova, M. Rehacek, J. Batrna, D. Plakinger, S. Stoppel, and B. Kozlikova. Visilant: Visual support for the exploration and analytical process tracking in criminal investigations. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):881–890, 2021.

[295] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: High resolution capture for modeling and animation. *ACM Transactions on Graphics*, 23(3):548–558, 2004.

[296] H. Zhou, S. Yang, and J. Zhu. Illumination invariant enhancement and threshold segmentation algorithm for asphalt pavement crack image. In *2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, pages 1–4, 2010.

[297] M. Zollhöfer, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Pérez, M. Stamminger, M. Nießner, and C. Theobalt. State of the art on monocular 3D face reconstruction, tracking, and applications. *Computer Graphics Forum (Eurographics State of the Art Reports 2018)*, 37(2), 2018.

[298] G. Zoss, T. Beeler, M. Gross, and D. Bradley. Accurate markerless jaw tracking for facial performance capture. *ACM Transactions on Graphics*, 38(4):1–8, 2019.

Curriculum Vitae

Contact

Name: Aleksandr Amirkhanov
Email: aleksandr@amirkhanov.net
Personal homepage: amirkhanov.net
Location: Vienna, Austria

Summary

- Research interest in medical, mathematical, and material visualization

- Strong knowledge in C++

- Solid knowledge in mathematics and algorithms

Employment History

Sep 2017–present

Researcher
Visualization Group, Institute of Visual Computing &
Human-Centered Technology, Vienna, Austria

Research on previewing dental treatment outcomes
in AR and assisting dental technicians in designing
aesthetically pleasing and functional prosthetic devices.

C++, C#, OpenCV, Boost, Dlib, Emscripten, CMake,
Visual Studio, JetBrains Rider, UWP, OpenGL, GLSL,
ImGui, Git

Mar 2014–Sep 2017

Researcher
CT Research Group, FHOOE, Wels Campus, Austria

Research on visualization and analysis of 3D CT images of composite materials. Development of the open_iA framework.

C++, Qt, Visualization Toolkit (VTK), Insight Toolkit (ITK), CMake, Visual Studio, Tortoise SVN, Git

Mar 2012–Nov 2012

Software engineer
LingPlay, Ufa, Russia

Game development for mobile platforms. I was involved in the development of Sheep Happens, and Shower Defense games.

C#, Unity, MonoDevelop, Xcode, PhysX

Nov 2011–Mar 2012

Software engineer
Research and production company "Progress," Ufa, Russia

Software development for medical healthcare services.

C#, JavaScript, ASP.NET MVC, Visual Studio

Education

2014–present

Doctor techn.
TU Wien, Vienna, Austria

Research in scientific visualization

2007–2012

Master of Science
Ufa State Aviation Technical University, Ufa, Russia

Study focus on computer science and mathematics

176

Professional Skills

Programming languages
C++, C#, Python (basics)

Tools
Qt, CMake, OpenGL, GLSL, OpenCV, Emscripten, Visualization Toolkit (VTK), Insight Toolkit (ITK), ImGui

Version control
Git, SVN

Languages

- Russian—native

- English—fluent

- German—basics

Interests

- Classical guitar

- Photography

- Skiing

- Fishing

- Game development

- Traveling

Publications

[1] H.-Y. Wu, *A. Amirkhanov*, N. Grossmann, T. Klein, D. Kouřil, H. Miao, L. R. Luidolt, P. Mindek, R. G. Raidou, I. Viola, M. Waldner, and M. E. Gröller, "Visualization working group at TU Wien: Visible facimus quod ceteri non possunt," Visual Informatics, 2021, ISSN: 2468-502X. DOI: 10.1016/j.visinf.2021.02.003.

[2] *A. Amirkhanov*, M. Bernhard, A. Karimov, S. Stiller, A. Geier, M. E. Gröller, and G. Mistelbauer, "Visual analytics in dental aesthetics," Computer Graphics Forum, vol. 39, no. 7, pp. 635–646, 2020. doi: 10.1111/cgf.14174.

177

[3] *A. Amirkhanov*, I. Kosiuk, P. Szmolyan, A. Amirkhanov, G. Mistelbauer, M. E. Gröller, and R. G. Raidou, "Manylands: A journey across 4D phase space of trajectories," Computer Graphics Forum, vol. 38, no. 7, pp. 191–202, 2019. doi: 10.1111/cgf.13828.

[4] *A. Amirkhanov*, A. Amirkhanov, M. Bernhard, Z. Toth, S. Stiller, A. Geier, E. Gröller, and G. Mistelbauer, "WithTeeth: Denture Preview in Augmented Reality," in Vision, Modeling and Visualization, F. Beck, C. Dachsbacher, and F. Sadlo, Eds., The Eurographics Association, 2018, ISBN: 978-3-03868-072-7. doi: 10.2312/vmv.20181250.

[5] C. Heinzl, *A. Amirkhanov*, and J. Kastner, "Processing, analysis and visualization of ct data," in Industrial X-Ray Computed Tomography, S. Carmignato, W. Dewulf, and R. Leach, Eds. Cham: Springer International Publishing, 2018, pp. 99–142, ISBN: 978-3-319-59573-3. doi: 10.1007/978-3-319-59573-3_4.

[6] R. Raidou, O. Casares-Magaz, *A. Amirkhanov*, V. Moiseenko, L. Muren, J. Einck, A. Vilanova, and M. Gröller, "Bladder runner: Visual analytics for the exploration of rt-induced bladder toxicity in a cohort study," Computer Graphics Forum, vol. 37, no. 3, pp. 205–216, 2018. doi: 10.1111/cgf.13413.

[7] *A. Amirkhanov*, D. Salaberger, J. Kastner, C. Heinzl, and E. Gröller, "Comparison of final fracture extraction techniques for interrupted in situ tensile tests of glass fiber reinforced polymers," in 7th Conference on Industrial Computed Tomography, 2017.

[8] *A. Amirkhanov*, A. Amirkhanov, D. Salaberger, J. Kastner, M. E. Gröller, and C. Heinzl, "Visual analysis of defects in glass fiber reinforced polymers for 4dct interrupted in situ tests," Computer Graphics Forum, vol. 35, no. 3, pp. 201–210, 2016. doi: 10.1111/cgf.12896.

[9] A. Reh, *A. Amirkhanov*, J. Kastner, E. Gröller, and C. Heinzl, "Fuzzy feature tracking: Visual analysis of industrial 4d-xct data," Computers & Graphics, vol. 53, pp. 177–184, 2015, ISSN: 0097-8493. doi: 10.1016/j.cag.2015.04.001.

[10] J. Weissenböck, A. Amirkhanov, W. Li, A. Reh, *A. Amirkhanov*, E. Gröller, J. Kastner, and C. Heinzl, "Fiberscout: An interactive tool for exploring and analyzing fiber reinforced polymers," in 2014 IEEE Pacific Visualization Symposium, Mar. 2014, pp. 153–160. doi: 10.1109/PacificVis.2014.52.