

# Scalable Visualization of Spatial Data in 3D Terrain

*Dissertation*

A thesis submitted for the degree of  
*Doktor-Ingenieur (Dr.-Ing.)*  
Faculty of Computer Science and Electrical Engineering  
University of Rostock

*Steve Dübel*

born on July 25, 1983 in Schwerin  
resident in Niex, Germany

June 25, 2019

**Principal Advisor :**

Prof. Dr.-Ing. habil. Heidrun Schumann, Universität Rostock

**External Reviewers:**

Prof. Dr.-Ing. habil. Rainer Groh, Technische Universität Dresden

Dr.-Ing. Ursula Hoffmann, Diehl Aerospace GmbH, Frankfurt

**Date of Submission:**

June 25, 2019

**Date of Defense:**

November 12, 2019

Keywords: *Spatial Data Visualization, Terrain Rendering, Scalability, Visualization Design, Prioritization, Focus and Context, Evaluation, GPU, Ray Tracing, Avionics*

Classification (ACM 2012): *Human-centered computing* → *Visualization*; *Computing methodologies* → *Computer graphics* → *Rendering*

Steve Dübel: Scalable Visualization of Spatial Data in 3D Terrain

©December 18, 2019

## Abstract

Designing visualizations of spatial data in 3D terrain is challenging because various heterogeneous data aspects need to be considered, including the terrain itself, multiple data attributes, and data uncertainty. It is hardly possible to visualize all these data aspects at full detail in a single image. Therefore, it is necessary to focus on relevant information to be emphasized, while less-relevant information can be attenuated. This requires the visualization approach to be scalable with respect to the requirements at hand. However, how to address scalability in this context has not yet been answered in current literature.

The goal of this thesis is to devise such a scalable visualization approach. In a first step, we collect existing and develop new methods for visualizing 3D terrain, spatial data, and uncertainty, and systematize them in a novel characterization schema. The systematized methods serve as the basic building blocks to be combined in a scalable fashion. To accomplish scalability, we propose to utilize two strategies: Prioritization and Focus & Context. Prioritization operates at the level of data aspects. Sophisticated methods are employed to visualize a prioritized data aspect with a high degree of detail, while the other data aspects are shown with less detail. Focus & Context considers scalability at the level of data elements. Selected data elements are emphasized, while others are shown in a reduced fashion.

Prioritization and Focus & Context provide designers with the required flexibility when creating visual representations of spatial data in 3D terrain. Evaluating whether a certain design decision is effective or not, however, is often not trivial. Therefore, we propose a novel evaluation approach based on saliency maps. Saliency maps are an established means to assess the conspicuity of elements in an image. We utilize these maps to enable designers to assess in real-time whether the prioritized data aspects and the focused data elements are visually prominent.

To address scalability from the technical point of view, we implemented the developed methods and concepts in a comprehensive visualization system called TEDAVIS. The system's flexible and modular architecture facilitates the design and evaluation of scalable visualizations of spatial data in 3D terrain. On top of that, we propose a scalable, hardware-based architecture for interactive terrain ray tracing that allows to strike a balance between image quality, render time, and resource consumption. Eventually, we apply our strategies to create sophisticated visual representations for various aviation-related scenarios.

## Zusammenfassung

Die Erstellung von Visualisierungen für räumliche Daten im 3D-Gelände ist schwierig, da viele heterogene Datenaspekte wie das Gelände selbst, die verschiedenen Datenattribute sowie Unsicherheiten bei der Darstellung zu berücksichtigen sind. Im Allgemeinen ist es nicht möglich, all diese Datenaspekte gleichzeitig in einer Visualisierung darzustellen. Daher ist es notwendig, sich auf die Darstellung der wichtigen Informationen zu konzentrieren und die anderen, weniger relevanten Informationen nur in abgeschwächter Form zu zeigen. Dies setzt Visualisierungsstrategien voraus, welche mit Blick auf die gegebenen Anforderungen skalierbar sind. Wie solch eine Skalierbarkeit erreicht werden kann, ist jedoch zur Zeit eine ungelöste Forschungsfrage.

Ziel dieser Arbeit ist die Entwicklung einer solchen skalierbaren Visualisierungsstrategie. Dazu tragen wir in einem ersten Schritt existierende Methoden, aber auch Neuentwicklungen zusammen, welche die Darstellung von 3D-Gelände, räumlichen Daten und Unsicherheiten ermöglichen, und charakterisieren diese anhand einer neuen Systematisierung. Wir nutzen die systematisierten Methoden als Grundbausteine, welche wir flexibel miteinander kombinieren, um Skalierbarkeit zu erreichen. Dazu schlagen wir zwei neue Strategien zur Datendarstellung im Gelände vor: Priorisierung und Fokus & Kontext. Priorisierung arbeitet dabei auf Ebene der Datenaspekte. Das bedeutet, dass komplexe, detaillierte Darstellungsmethoden zur Kommunikation des priorisierten Datenaspektes angewendet werden, während andere Aspekte nur in abstrahierter Form gezeigt werden. Fokus & Kontext wiederum betrachtet Skalierbarkeit auf Ebene der Datenelemente. Ausgewählte Datenelemente werden dabei akzentuiert, während die anderen in reduzierter Form dargestellt werden.

Priorisierung und Fokus & Kontext stellen dem Visualisierungsautor dabei die nötige Flexibilität zur Darstellung von räumlichen Daten im Gelände zur Verfügung. Ob ein entwickeltes Visualisierungsdesign aber tatsächlich effektiv ist, kann nur schwer evaluiert werden. Daher stellen wir einen neuartigen Ansatz zur Evaluation von Visualisierungen anhand von so genannten Salienzkarten vor. Salienzkarten sind eine etablierte Methode um die Auffälligkeit von Bildelementen zu bewerten. Wir nutzen diese Karten, um dem Visualisierungsautor in Echtzeit die Möglichkeit zu geben, festzustellen, ob der priorisierte Datenaspekt oder die Datenelemente im Fokus tatsächlich prominent dargestellt werden oder nicht.

Um Skalierbarkeit auch von der technischen Seite aus zu betrachten, haben wir alle entwickelten Methoden und Konzepte in einem umfassenden Visualisierungssystem mit dem Namen TEDAVIS implementiert. Die flexible, modulare Architektur unseres Systems erlaubt die Erstellung und die Evaluation von skalierbaren Visualisierungen von räumlichen Daten im 3D-Gelände. Außerdem stellen wir eine neue, flexible Hardwarearchitektur für das Raytracing von Geländemodellen vor, welche es erlaubt zwischen Bildqualität, Renderzeit und Ressourcenverbrauch zu skalieren. Unsere Konzepte wenden wir schließlich an, um komplexe visuelle Darstellungen für verschiedene Szenarien im Kontext vom Flugverkehr zu erstellen.

”From one thing, know ten thousand things.”

*Miyamoto Musashi*

*Dedicated to my beloved family.*

## Acknowledgements

The work presented in this thesis was only made possible due to the support, aid, and collaboration of many people. I would like to take this opportunity to express by appreciation.

I owe my deepest gratitude to my supervisor and mentor Heidrun Schumann for guiding me through this work. I have greatly benefited from our many constructive conversations. They were illuminating—although sometimes only afterwards. Thank you for your everlasting patience, understanding and for your optimism and encouragements during writing this thesis.

I would like to thank Rainer Groh for his valuable feedback and for agreeing to review this dissertation. My gratitude goes also to our long-standing collaboration partners from Diehl Aerospace, in particular, Ursula Hoffmann, Joachim Bader, and Stefan Müller-Divéky. Thank you for your insightful comments and suggestions. It was a great pleasure and an invaluable experience to work with you. I also appreciate Ursula Hoffmann's agreement to review this thesis.

I am very grateful to my colleagues with whom I enjoyed many technical and sometimes philosophical discussions. A special thanks to Christian Tominski for giving me so much guidance and feedback, especially with respect to scientific writing. I also really appreciated our many diverting breakfast meetings. Thanks to Martin Luboschik for his valuable inputs regarding this work. My sincere thanks goes also to Martin Röhlig, Christian Brode-Richter, and Falko Löffler with whom I had the honor to collaborate on this project and whose scientific work considerably influenced me.

Finally, I would like to express my deepest appreciation to those who are most important to me—my family. I thank you, Tanja and Hermine, there are no proper words to express my heartfelt gratitude for your continued patience, your help, your encouragement, and your warm love. You are my source of joy and the center of my world.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Fundamentals and Related Work</b>	<b>7</b>
2.1	Terrain Visualization . . . . .	8
2.1.1	Terrain Model . . . . .	8
2.1.2	Visualization Process . . . . .	9
2.1.2.1	Rasterization . . . . .	9
2.1.2.2	Ray Tracing . . . . .	11
2.1.2.3	Design Choices . . . . .	13
2.2	Spatial Data Visualization . . . . .	17
2.2.1	Definition of Spatial Data . . . . .	18
2.2.2	Visualization Process . . . . .	19
2.3	Uncertainty Visualization . . . . .	22
2.3.1	Definition of Uncertainty . . . . .	22
2.3.2	Visualization Process . . . . .	23
2.4	Scalability Challenge . . . . .	25
2.5	Summary . . . . .	26
<b>3</b>	<b>Approach to Scalable Visualization of Spatial Data in Terrain</b>	<b>29</b>
3.1	Demonstrative Example . . . . .	30
3.2	Scalability . . . . .	32
3.3	Prerequisites on Scalable Visualization . . . . .	33
3.3.1	Presentation . . . . .	33
3.3.2	Processing . . . . .	33
3.4	Approach . . . . .	34
<b>4</b>	<b>Basic Visualization Strategies</b>	<b>37</b>
4.1	A Novel Systematization of 2D and 3D Presentation . . . . .	39
4.1.1	Systematization . . . . .	39
4.1.2	Characteristics and Discussion . . . . .	41
4.1.3	Conclusion . . . . .	47
4.2	Design Options for Terrain and Data Presentations . . . . .	48
4.2.1	Terrain Visualization . . . . .	48
4.2.2	Data Visualization . . . . .	52
4.2.2.1	Geometrical Data . . . . .	52
4.2.2.2	Numerical Data . . . . .	55
4.2.3	Data Quality . . . . .	58
4.2.4	Spatial Awareness . . . . .	59

---

4.3	Summary . . . . .	64
<b>5</b>	<b>Advanced Visualization Strategies</b>	<b>67</b>
5.1	Prioritization . . . . .	70
5.1.1	Motivation . . . . .	70
5.1.2	General Approach . . . . .	71
5.1.3	Design Strategies . . . . .	72
5.1.4	Discussion . . . . .	77
5.2	Focus & Context . . . . .	78
5.2.1	Motivation . . . . .	78
5.2.2	General Approach . . . . .	78
5.2.3	Procedure and Design Strategies . . . . .	79
5.2.4	Discussion . . . . .	85
5.3	Evaluation using Saliency Information . . . . .	88
5.3.1	Motivation . . . . .	88
5.3.2	Visual Clutter . . . . .	89
5.3.3	Saliency . . . . .	91
5.3.4	Evaluation of Visualizations . . . . .	94
5.3.5	Discussion . . . . .	98
5.4	Summary . . . . .	100
<b>6</b>	<b>Implementation Strategies</b>	<b>103</b>
6.1	TEDAVIS: A Scalable Terrain-based Visualization Framework . . . . .	105
6.1.1	General Approach . . . . .	105
6.1.2	Requirements . . . . .	105
6.1.3	Architecture . . . . .	107
6.1.4	Application of TEDAVIS . . . . .	110
6.1.5	Discussion . . . . .	111
6.2	Scalable Architecture for Ray Tracing 3D Terrain . . . . .	112
6.2.1	General Approach . . . . .	112
6.2.2	Architecture . . . . .	114
6.2.2.1	Dynamic Task Management . . . . .	114
6.2.2.2	Scalable Ray Tracer . . . . .	116
6.2.2.3	Functional Model . . . . .	119
6.2.3	Implementation . . . . .	120
6.2.4	Discussion . . . . .	123
6.3	Summary . . . . .	124
<b>7</b>	<b>Conclusion and Outlook</b>	<b>125</b>
7.1	Summary . . . . .	125
7.2	Comments on Design Process and Experts' Feedback . . . . .	126
7.3	Future Work . . . . .	127
	<b>Bibliography</b>	<b>131</b>

# List of Figures

2.1	Triangulation of the terrain surface. . . . .	10
2.2	Hybrid LOD strategies for terrain meshes. . . . .	11
2.3	Terrain representation through sketching. . . . .	15
2.4	Terrain representation through hatching. . . . .	16
2.5	Terrain representation through shading. . . . .	17
2.6	Terrain representation through textures. . . . .	18
2.7	The Visualization Pipeline (adapted from <a href="#">Haber and McNabb (1990)</a> ) . . .	19
2.8	Intrinsic and extrinsic visualizations of spatial data. . . . .	21
2.9	Uncertainty Reference Model (adapted from <a href="#">Brodie et al. (2012)</a> ) . . . . .	23
3.1	Synthetic vision display. . . . .	30
3.2	Triangle of tension regarding the quality of the visualization, the consumed resources, and the required render time. . . . .	34
4.1	Systematization of visualization techniques based on the dimensionality of the attribute space's and reference space's presentation. . . . .	40
4.2	Exemplary visualization techniques for each category of the proposed systematization. . . . .	41
4.3	Exemplary $\mathcal{A}^2 \oplus \mathcal{R}^2$ visualization techniques. . . . .	43
4.4	Exemplary $\mathcal{A}^2 \oplus \mathcal{R}^3$ visualization techniques. . . . .	44
4.5	Exemplary $\mathcal{A}^3 \oplus \mathcal{R}^2$ visualization techniques. . . . .	45
4.6	Exemplary $\mathcal{A}^3 \oplus \mathcal{R}^3$ visualization techniques. . . . .	46
4.7	Enhanced edges using line drawing. . . . .	48
4.8	Conveying fine 3D structures of the terrain surface using different illumination models. . . . .	50
4.9	Application of textures onto the terrain. . . . .	50
4.10	Texturing terrain in combination with other representation techniques. . . .	51
4.11	Visualization of a trajectory. . . . .	52
4.12	Improving communication of the position and path of the trajectory. . . . .	53
4.13	Communicating properties of the trajectory. . . . .	54
4.14	Strategies to show numerical data. . . . .	55
4.15	Numeric data representations using different visual encodings. . . . .	56
4.16	Intrinsic uncertainty visualizations using different levels of opacity. . . . .	57
4.17	Options for visualizing uncertainty. . . . .	58
4.18	Spatial awareness regarding trajectory visualization. . . . .	61
4.19	Improving spatial awareness regarding trajectory visualization using animated lines. . . . .	61
4.20	Improving spatial awareness regarding trajectory visualization using grids. . .	62
4.21	Enhancing the perception of the spatial distribution of objects within terrain. .	63

---

4.22	Application of volumetric light to improve spatial awareness. . . . .	64
5.1	Visualization of 3D terrain, wind speed forecast data, and forecast uncertainty. . . . .	70
5.2	Prioritization strategies for terrain. . . . .	73
5.3	Prioritization strategies for data. . . . .	74
5.4	Prioritization strategies for uncertainty. . . . .	75
5.5	Scalability of the visualization design with respect to the given priority. . . . .	77
5.6	Distinction between a spatial and a data focus. . . . .	79
5.7	Division of the visualization into focus, intermediate context, and remaining context. . . . .	80
5.8	Strategies for spatial focus selection. . . . .	82
5.9	Strategies for data focus selection. . . . .	83
5.10	Computation of the intermediate context. . . . .	84
5.11	Adjustment of the representation using abstraction. . . . .	85
5.12	Adjustment of the representation using highlighting. . . . .	86
5.13	Adjustment of the representation using distortion. . . . .	87
5.14	Strategies for reducing labels and POIs, using Focus & Context. . . . .	88
5.15	Application of the Canny Edge Detection. . . . .	90
5.16	Application of the Saliency Map. . . . .	94
5.17	Saliency Maps for each visualization described in Section 5.1. . . . .	95
5.18	Saliency Map for the Focus & Context example using abstraction. . . . .	97
5.19	Saliency Map for the Focus & Context example using highlighting through color adjustment. . . . .	98
5.20	Saliency Map for the Focus & Context example using highlighting through blurring. . . . .	99
5.21	Saliency Map for the Focus & Context examples using distortion. . . . .	100
5.22	Visual clutter analysis during a simulated flight towards a destination airport. . . . .	101
6.1	Overview screenshot of TEDAVIS. . . . .	106
6.2	Overview of the architecture of TEDAVIS. . . . .	108
6.3	Rendering architecture for scalable terrain ray tracing. . . . .	113
6.4	Global scheduling of the segmented stream. . . . .	115
6.5	Illustration of the beam tracing based fast start. . . . .	117
6.6	Different strategies for surface interpolation. . . . .	118
6.7	Surface generation through application of micro-structure. . . . .	119
6.8	Example for task generation from the scalable ray tracer to the functional model. . . . .	120
6.9	Exemplary function operators for ray generation, surface computation, and shading. . . . .	121
6.10	Results of the performance tests. . . . .	123

# List of Abbreviations

<b>DTM</b>	Digital Terrain Model	8
<b>DSM</b>	Digital Surface Model	8
<b>DEM</b>	Digital Elevation Model	8
<b>TIN</b>	Triangular Irregular Networks	9
<b>LOD</b>	Level-of-Detail	9
<b>CLOD</b>	Continuous Level-of-Detail	10
<b>BVH</b>	Bounding Volume Hierarchies	12
<b>MM</b>	Maximum Mipmap	12
<b>GIS</b>	Geo-Information Systems	17
<b>SVS</b>	Synthetic Vision Systems	30
<b>NPR</b>	Non-photorealistic Rendering	53
<b>PR</b>	Photorealistic Rendering	62
<b>POI</b>	Points of Interest	85
<b>DOI</b>	Degree of Interest	86
<b>GUI</b>	Graphical User Interface	88
<b>UI</b>	User Interface	107
<b>FPS</b>	Frames per Second	107
<b>API</b>	Application Programming Interface	110



# Chapter 1

## Introduction

Spatial data play a significant role in our world today. Knowledge about the environment, including terrain, infrastructure, population, weather, or ecology is utilized in nearly all scientific and industrial areas. While in long-established fields such as meteorology or geology spatial information has been used for quite some time, recently new fields of application have arisen, for example, climate change adaptation, virtual archeology, autonomous cars, and flight management systems. To satisfy the need of more and more high-resolution spatial data, our world today is measured automatically with an unprecedented accuracy using remote sensing technologies and sensor networks. Even more data are created, for instance, through simulation, tracking, geo-tagging, and digitalization. In sum, the amount of available data today is immense and grows continuously. [Hahmann and Burghardt \(2013\)](#) have shown that at least one quarter of all data available is directly geo-referenced. This means that today the amount of spatial data would total approximately ten billion terabytes ([Hajirahimova and Aliyeva \(2017\)](#)). Scientists and domain experts from various fields are using these data for extensive analysis. However, turning the data into insights is difficult due to its size, complexity, and heterogeneity. That is why most of the available data, though potentially useful, has not yet been analyzed. In this context, Data Visualization has been proven to be highly beneficial and thus has become one of the key technologies to assist the experts in decision making and knowledge creation processes.

### Motivation

Visualizing spatial data allows to see, explore, and understand large amounts of information at once by taking advantage of the human eye's broad bandwidth pathway into the mind ([Cook and Thomas \(2005\)](#)). It can efficiently communicate the information of interest as well as their spatial distribution and reveals structures and patterns within the data. For this purpose, tailored visualizations that fit the data's characteristics are necessary.

The visual analysis of spatial data requires the depiction of different *data aspects*, such as the frame-of-reference (e.g., the terrain), the data values themselves (e.g., weather data), and information about the data's quality (e.g., uncertainty). For instance, the daily weather forecast for a country is typically shown above a suitable map. In this example, the focus of the presentation lies on the depiction of the data values, while the depiction of the frame-of-reference aids the orientation. In some weather forecasts, further information about rain probability is common. This aids comprehension of potential uncertainties in the data. In other applications the depiction of the frame-of-reference is the most important aspect in the

visualization, while further data provides contextual information. For example, geologists might examine the structure of terrain, using data about soil characteristics or natural resources. In sum, different scenarios requires the depiction of different data aspects in a combined fashion. For spatial data, primarily the depiction of the data values together with their frame-of-reference is essential.

Creating suitable visualizations is an intricate process that is subject to research in various scientific fields, including information visualization. Due to the large amount of individual applications and data types, there exist many different, tailored visualizations for spatial data. Most typically, approaches in literature hereby depict the frame-of-reference as a 2D map. This might be due to the fact that maps are well known, easy accessible, and straightforward to depict. However, in other applications a 3D depiction of the terrain's topography is required (e.g., oceanography, aviation). Yet, data visualization using a 3D depiction of the terrain is a rarely considered research topic. With this thesis we aim to fill this gap.

## Challenges and Goals

Although spatial data visualizations have been proven to be very effective, various challenges still exist. Many of them are based on one fundamental issue: the massive size of the data. Visualizations not uncommonly deals with data sets that are too big to fit computational capabilities of current technology. This is particularly true if complex, high-resolution terrain models are involved, which size easily exceeds multiple gigabytes. Trying to show multifarious data along with the terrain aggravates this problem. To scale down the amount of data, usually reduction methods, such as filtering or aggregation, are applied. Still, in many cases it is indispensable to communicate many data together. This holds true for unrestricted data exploration tasks, for the analysis of complex interdependencies and patterns, and for in-depth examinations of terrain topography. In these situations reduction methods are applicable only to a certain extent and thus the data kept remain immense.

Another issue arises from the fact that visualizations are typically designed to facilitate a specific set of requirements, including the objectives of the user, the utilized data, and the available technology. In many situations these requirements are not static but can change during the usage of the presentation. This, in turn, affects the efficacy of the visual design.

In sum, designing visualizations in this context leads to two major challenges:

**Scalable Presentation** Designing suitable visualizations is always accompanied with the problem of deciding which part of the data to show. Giving all the details at once would inevitably overload the presentation, lead to visual clutter, and render the visualization ineffective. Thus, visualization methods typically seek a compromise between data presentation on the one hand and the available visual budget on the other. Fundamentally, two strategies are pursued. Either presenting the details of a smaller subset of the data or giving a rough overview of a larger portion of the data. There are also methods that attempt to combine both strategies through interactive means, such as Overview & Detail and Focus & Context. For general applications it is often difficult to decide how much information is *necessary* for an adequate overview and how much is *possible* for a detailed view. Striking an appropriate balance highly depends on requirements, including the task at hand, the available data, and the available hardware. Yet, these requirements are typically not static, but often shift

---

dynamically. A change of one requirement consequently enforces an adjustment of the whole presentation. Therefore, we need visualization strategies that allow us to scale between different demands in order to comply with changing conditions.

**Scalable Processing** Visualizing large data involves heavy computation for data preparation, mapping, and rendering. The most limiting factors hereby are computation time and memory capacity. This applies particularly to interactive rendering, where only several milliseconds are permissible and concomitantly only a few gigabytes of memory are available. These restrictive limits contradict with the demand for high quality visualizations of large, detailed terrain and complex data. Moreover, visualizations today are not only processed by high-end machines but also by mobile devices or embedded systems that computational power differs significantly. To deal with these challenges, we require visualization methods that are scalable to balance between computational capabilities and qualitative demand.

On these grounds, the goal of this thesis is to provide a concept for scaling the visualization with respect to large spatial data and changing requirements that affects information communication on the one hand, and the computation on the other. We hereby consider both the depiction of the data and the depiction of the terrain in one combined presentation and focus primarily on 3D presentations.

## Approach and Contribution

In this thesis we engage the aforementioned challenges by pursuing a scalable visualization approach. Scalability is one of the top research topics not only in visualization but in every field of data analysis and beyond. It describes the capability of a system to handle varying work loads whilst remaining effective in output. In terms of this thesis, it particular refers to the ability of visualization technology to deal with varying conditions with regard to task, data, and hardware capability. In contrast to technical systems, human abilities and skills cannot be extended at will beyond certain limits. With this in mind, scalability in visualization does not only refer to processing, but also to the ability to adapt large-scale problems to a level that humans can deal with ([Cook and Thomas \(2005\)](#)).

Obviously, such a scalability cannot be provided by a single visualization method. Hence, our basic idea is to provide a range of visualization methods that can be applied interchangeably and that vary, for one thing, in the abstraction level of the presentation and, for another thing, in the computational expense. For this purpose, we investigate existing methods and also develop new techniques. Creating comprehensive solutions with regard to holistic visualizations, however, is a complex task. Therefore, we first consider individual types of information—which we call data aspects—that are essential in spatial data visualizations, such as the frame-of-reference and the data but also meta-information such as uncertainty. Afterwards, we address the design of combined visualizations considering all aspects simultaneously. In sum, our approach can be outlined in the following four steps:

- 1. Analyzing characteristics of 2D and 3D spatial data visualization:** As the initial step, we review existing work to determine common characteristics for comparable visualization approaches. To this end, we introduce a novel systematization and group visualizations according to the dimensionality of their presentation—either

2D or 3D. We hereby consider the representation of two aspects, the data and the frame-of-reference, separately. It will be shown that most visualization approaches can be assigned to a combination of either 2D or 3D data representations with 2D or 3D frame-of-reference representations and that these groups can be matched with fundamental properties. This systematization was also presented at the *3DVis, an IEEE VIS International Workshop* under the title “*2D and 3D Presentation of Spatial Data: A Systematic Review*” (Dübel et al. (2014)).

**2. Development of scalable visualization methods for varying data:** Afterwards, visualization methods for the individual aspects, in particular terrain, data, and uncertainty, are investigated. We will examine diverse methods for each aspect and assess their characteristics regarding the conveyed details and abstraction levels. In this context, we consider existing techniques, but also introduce novel approaches. Most parts of this analysis appeared in the journal *IPRS International Journal of Geo-Informatics* under the title “*Visualization of Features in 3D Terrain*” (Dübel and Schumann (2017)). Aim of this step is to provide a set of options for different visual designs in order to facilitate scalability for varying data.

**3. Development of scalable visualizations with respect to varying tasks:** In this step, we address the question, how to create suitable visualizations by combining the various design options for the individual data aspects, depending on different tasks that arise from specific scenarios. For this purpose, we first present a strategy that determines suitable techniques for each aspect and combine them in order to create tailored and integrated presentations. Due to the proposed interchangeability of the visualization methods, we support scaling the presentation towards the given scenario. This concept was described in the paper “*Visualizing 3D Terrain, Geo-Spatial Data, and Uncertainty*”, which has been published in the international journal *Informatics* (Dübel et al. (2017)).

Afterwards, we address a more interactive approach to adapt integrated visualizations: Focus & Context. By determining subsets of the data as focus areas, different presentation techniques can be applied to emphasize data of interest whilst still providing contextual information. Considering the individual data aspects, we apply Focus & Context on our data representation in 3D terrain in order to support specific purposes, such as *direct* or *indirect* search (Andrienko and Andrienko (2006)). This concept was presented at the *EuroVis Workshop on Visual Analytics* and has been published in the proceedings under the title “*Visual Analysis of Geo-spatial Data in 3D Terrain Environments using Focus+Context*” (Richter et al. (2017)).

The proposed concepts to create scalable visualizations eventually entails the demand for a way to evaluate them. Therefore, we will propose a novel perception-based, computational approach to assess a visualization regarding its capability to emphasize the relevant data and regarding potential visual clutter. This novel approach has not been published yet.

---

**4. Implementation with respect to varying hardware environments:** In the fourth step, we will address scalability from a more technical point of view. To this end, we implement our strategies for scalable spatial data visualization into one sophisticated visualization system called TEDAVIS. It combines the various presentation techniques and the different concepts to design comprehensive spatial data visualizations. The application environment of TEDAVIS hereby scales from high- to medium-performance desktop PCs, but allows also the application for low-performance embedded target systems. This framework was published in the aforementioned paper “*Visualization of Features in 3D Terrain*” (Dübel and Schumann (2017)).

Finally, we develop an advanced, hardware-based architecture for terrain ray tracing that allows to strike a balance between render time, resource consumption, and image quality and that also considers different hardware capabilities. This approach was presented at the conference *International Summerschool on Visual Computing* and has appeared in the proceedings under the title “*A Flexible Architecture for Ray Tracing Terrain Heightfields*” (Dübel et al. (2015)).

## Thesis structure

This thesis is divided into six further chapters that are summarized in the following. [Chapter 2](#) provides basic definitions and termini regarding the visualization of spatial data. It reviews existing work by considering current presentation strategies with respect to different data aspects. Furthermore, the relevance of scalability for the information visualization will be explained. [Chapter 3](#) presents the application context of our spatial data visualizations in detail, deduces the need for a scalable approach and discusses problems and challenges in this context. From here, our approach for scalable spatial data visualization is outlined. [Chapter 4](#) lays the foundation of our approach by analyzing the characteristics of feasible visualization methods. We will start with a systematization of 2D and 3D visualizations with respect to data and terrain depictions. Afterwards, each aspect is further examined towards its level of complexity and appropriate presentation techniques are introduced. The following [Chapter 5](#) addresses scalability in combined visualizations. To this end, two strategies are presented: Scalability through adapting the representation of each information aspects using Prioritization and scalability through adapting the representation of parts of the information aspects using Focus & Context. Moreover, we will present a perception-based evaluation approach in this chapter. Eventually [Chapter 6](#) deals with scalability in terms of varying hardware configurations. In this context, we will introduce our comprehensive visualization framework TEDAVIS and a scalable ray tracing architecture. Finally, this thesis is concluded in [Chapter 7](#). It provides a summary and a recap of our collaboration with the domain experts, before the thesis is ended by an outlook to future work.



## Chapter 2

# Fundamentals and Related Work

In this chapter we will introduce basic concepts regarding spatial data visualization and set them into context with the current state of the art. To this end, we examine three aspects of spatial information that are the terrain, the data, and uncertainty. First, we will discuss, how terrain can be depicted and how different presentation styles affect the surface depiction. Secondly, we will provide fundamentals about spatial data and their visualization. Eventually, we will describe the visualization of uncertainty information, before finally touching upon the scalability challenge with respect to information visualization.

### Contents

---

2.1	Terrain Visualization . . . . .	8
2.1.1	Terrain Model . . . . .	8
2.1.2	Visualization Process . . . . .	9
2.1.2.1	Rasterization . . . . .	9
2.1.2.2	Ray Tracing . . . . .	11
2.1.2.3	Design Choices . . . . .	13
2.2	Spatial Data Visualization . . . . .	17
2.2.1	Definition of Spatial Data . . . . .	18
2.2.2	Visualization Process . . . . .	19
2.3	Uncertainty Visualization . . . . .	22
2.3.1	Definition of Uncertainty . . . . .	22
2.3.2	Visualization Process . . . . .	23
2.4	Scalability Challenge . . . . .	25
2.5	Summary . . . . .	26

---

## 2.1 Terrain Visualization

Regarding terrain visualization, a lot of research has been done, especially in the context of geo-science. Here, digital terrain presentations are ubiquitously used to facilitate studies of natural and anthropogenically-influenced phenomena (Bonaventura et al. (2017)). Thus, a large number of visualization techniques for terrain data has been developed. A comprehensive review on the history of terrain visualization, in particular for geographic information systems, is given by Ruzinoor et al. (2012). In this section, we will focus on the current state of the art to depict terrain surfaces. Hence, we will first describe the creation of terrain models before going into details of the presentation methods. We will hereby take a look on two fundamental render methods, that are, rasterization and ray tracing. Moreover, we will further examine different choices of render styles that allow to adjust the presentation of the terrain to given requirements, for example, realistic and artistic depiction.

### 2.1.1 Terrain Model

Basis of the visualization are the terrain data. Sources are, for example, airborne and terrestrial laser scanning (LIDAR), photogrammetry, and satellite images. In other cases, terrain data are synthesized artificially through procedural computing or terrain model designer (e.g., Belhadj (2007)). Most commonly, terrain data are represented in form of a Digital Terrain Model (DTM)<sup>1</sup>. In contrast to its counterpart Digital Surface Model (DSM), DTM consists only of height information of the terrain without taking vegetation or buildings into account (Hirt (2014)). In literature the more general term Digital Elevation Model (DEM) is also used, however, this term is applied ambiguously as a synonym either for DTM, DSM, or both. Thus, in this thesis we use the term DTM only. Toppe (1987) distinguishes two types of DTM: primary models and secondary models. The term primary models is correlated with measured or raw data where height information is unstructured, often given on an irregular grid. In contrast, secondary models reference processed data where the information is often interpolated to fill missing data, smooth inaccuracies, or to structure the data along a regular grid. Height data that is regularly organized on a grid is also called heightfield.

A heightfield can be described as a 2D function  $h: \mathbb{R}^2 \mapsto \mathbb{R}$  with  $(x, y) \mapsto (z)$ , which provides an elevation  $z \in \mathbb{R}$  for each coordinate  $(x, y) \in \mathbb{R}^2$ . Heightfields are the most common terrain model due to their simple structure and easy accessibility. They are often represented by gray-scale images where each pixel along a row ( $x$ ) and column ( $y$ ) respectively has a gray value assigned, which is construed as the height value ( $z$ ). The ratio of pixels per area unit constitutes the resolution of a heightfield, which directly correlates with their quality and their memory footprint. The more pixel given per area unit, the more accurately the terrain can be represented. At the same time, the memory size is increased. For example, one square kilometer terrain represented by an uncompressed heightfield at a resolution of ten meters results into approximate 40 kilobyte of data, while the same heightfield with a resolution of one meter already requires 4 megabytes of data. Heightfields provide only one function value (height) per argument. Therefore, it describes a closed surface without holes, overhangs, or caves. This, however, also limits heightfields

---

<sup>1</sup>The 3D terrain models depicted in this thesis were generated from the following elevation data: Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) Global Digital Elevation Model (GDEM), a product of the Ministry of Economy, Trade, and Industry (METI) of Japan and the United States National Aeronautics and Space Administration (NASA) NASA and METI (2019).

to represent only simple surfaces. Due to this reason, heightfields are also considered 2.5D models.

The depiction of terrain data in form of heightfields is a complex process, since they usually cannot be rendered directly. This is due to the fact that heightfields hold height values only for discrete sample points and do not explicitly provide a continuous, smooth surface. Thus, they must be converted into other representations that allows the rendering algorithm to immediately generate the surface. The concrete type of representation, however, depends on the applied rendering strategy during the visualization.

### 2.1.2 Visualization Process

The visualization of the terrain is a technically sophisticated process due to model size and high demands on image quality and render speed. In principle, there are two basic render strategies that we will describe in this section: rasterization and ray tracing.

#### 2.1.2.1 Rasterization

Rasterization is the most common and fastest render strategy for terrain data and beyond. Therefore, current graphics hardware is optimized for rasterization only<sup>2</sup> and can render millions of primitives extremely fast. The main idea is to project the graphical primitives (points, lines, or triangles) onto an image plane and afterwards convert (rasterize) this image plane into a discrete matrix of pixel.

Rendering terrain heightfields through rasterization still poses a difficult task. To render terrain heightfields via rasterization, the conversion into graphical primitives is necessary. Most work hereby concentrates on the creation of Triangular Irregular Networks (TIN) through Delaunay (e.g., Fowler and Little (1979)) and other triangulation methods (e.g., Garland and Heckbert (1995) and Wu and Amaratunga (2003)). Less commonly used are regular triangular meshes (e.g., Zyda et al. (1993)), which usually consist of more triangles than TIN but can have advantages in compactness and accessibility. In Figure 2.1 a regular and an irregular triangular mesh are illustrated.

To represent heightfields accurately, meshes that consist of a large number of triangles are necessary. However, such meshes are typically too large to be rendered in interactive frame rates. Thus, adequate approximations, depending on a given error tolerance, are used. Common types of errors are the geometrical error of the mesh either in object space or in image space. The error in object space describes the deviation of the approximation from the original mesh. The image space error, on the other hand, describes the deviation of the approximation's rendered image from the original mesh's rendered image. The image space error is especially important to maintain image quality. To compare the quality of different approximative meshes, various error metrics are used, such as mean square error, Hausdorff distance (Aspert et al. (2002)), or quadric error metric (Garland and Heckbert (1997)).

Approximative meshes can be created by using diverse simplification algorithms, such as vertex removal (Schroeder et al. (1992)), vertex clustering (Rossignac and Borrel (1993)), wavelets (Gross et al. (1995)), or iterative edge contraction (Hoppe (1997)). Commonly, not one but multiple surface approximations are used to create a number of Level-of-Detail (LOD). By doing so and depending on the current viewing situation, the most fitting

---

<sup>2</sup>Some of the cutting-edge graphics hardware is also equipped with processing units that are optimized for ray tracing, though their function lies mainly in shading and not in rendering per se.

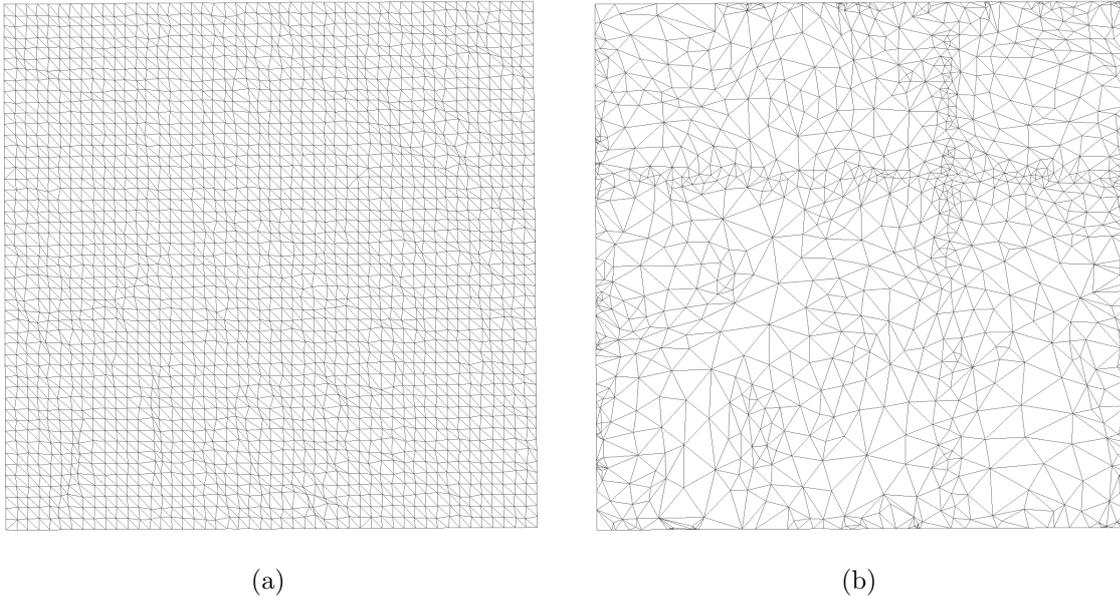


Figure 2.1: Triangulation of the terrain surface: (a) a regular and (b) an irregular network.

mesh can be chosen during the render process. Due to the spatial extent of terrain models, discrete **LOD** are often insufficient. When viewing the terrain in horizontal direction, a single detail level would either provide too much details for regions far away or too few details for regions near to the observer.

To engage this problem Continuous Level-of-Detail (**CLOD**) simplify the mesh depending on their projection in image space. For instance, Lindstrom et al. (1996) describe an algorithm that dynamically generates a mesh of the terrain surface by controlling the density and size of the triangles in the mesh depending on the distance to the camera. However, updating the whole mesh while the camera moves around is quite expensive. Thus, they exploit frame-to-frame and object-space coherences during the update phase. In contrast Duchaineau et al. (1997) provide a guaranteed error bound for their **CLOD** approach. This can assure a certain image quality through a given error threshold.

Moreover, hybrid approaches exist that try to combine the advantage of both—the simplicity of discrete **LOD** and the effectiveness of **CLOD**. Losasso and Hoppe (2004), for example, propose the use of geometric clipmaps to divide the mesh into discrete, concentric sectors around the viewer. As illustrated in Figure 2.2a, each sector extending from the center outwards is represented by a coarser **LOD** and the center point of this structure is updated as the viewer’s position changes more than a given threshold. In another approach, Ulrich (2002) divides the mesh into four large patches. Similar to a quadtree, each patch is repeatedly divided into four smaller patches consisting of more triangles (meaning more detail) than the former one. This process is carried forward to create a tree, where each level represents a **LOD**. By choosing an appropriate **LOD** for one patch and descending the tree only if necessary, computation is quite fast. At the same time regions near to the camera as well as regions far away can be threaten properly (see Figure 2.2b).

After the determination of a suitable **LOD**, the rasterization process computes the visibility of the polygons by first projecting them onto the image plane, then rasterizing them into fragments, and eventually separating the visible fragments (pixels) from those occluded by comparing their distance to the camera. Subsequently, a shading step computes

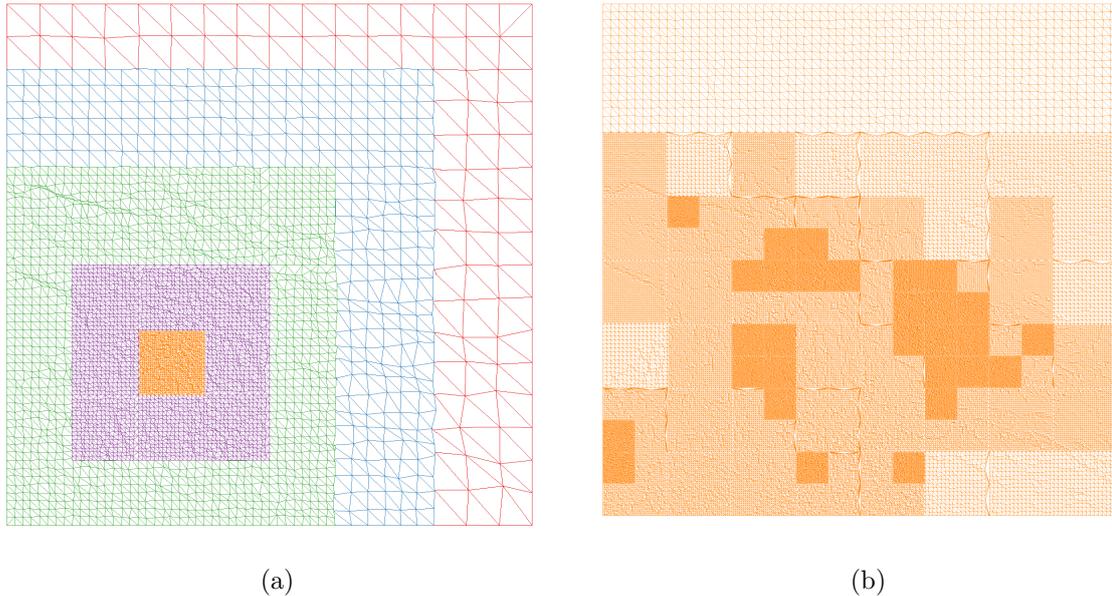


Figure 2.2: Hybrid LOD strategies for terrain meshes: (a) Clipmap and (b) Chunked LOD.

the final color of each fragment, typically by means of material properties and the given lighting situation. Since rasterization does not implicitly calculate lighting or shades within the scene, certain illumination models, for example, Phong reflection model (Phong (1975)), are applied. However, the shading step is not necessarily limited to solely illumination but can be arbitrarily complex to achieve various representation designs. We will describe this aspect in more detail in Section 2.1.2.3.

### 2.1.2.2 Ray Tracing

Ray tracing is the process of prosecuting the path of light through the scene to the view point in inverse direction. The algorithm can be divided into two parts, the ray casting, which determines the visible surfaces, and the traversal of light rays for illumination computation. Ray casting starts from the view point where one ray per pixel starts its travel, first through the center of the pixel in the image plane and then through the scene. The ray traverses the scene until it hits an object, which is then determined visible in the specific pixel where the ray started. In the second part, the color of the pixel is determined by computing the illumination of the object's surface. For this purpose, the reflection is simulated by casting further rays into the scene. Depending on the specific algorithm, for instance, recursive ray tracing (Whitted (1980)), distributed ray tracing (Cook et al. (1984)), path tracing (Kajiya (1986)), or bidirectional path tracing (Lafortune and Willems (1993)), either one or multiple rays are traced until they hit a light source that ends their path, or hit another surface, where they can be absorbed or be reflected again. Eventually, the lighting information and the material properties of the surface are used to determine the color of the pixel. More details on various ray tracing algorithms can be found in literature, for instance by Glassner (1989) and by Wald (2005).

Recently, ray tracing has become more and more popular for terrain rendering. This is, for one thing, due to the realistic, high quality images that can be produced, on the other hand, ray tracing does not require triangle networks or other heavily preprocessed model

representations. It still requires some preprocessing, but this is limited to plain spatial ordering and does not involve any complex computations (Wald and Slusallek (2001)). Thus, ray tracing enables direct rendering of terrain heightfields, which results in reduced preprocessing time and low memory consumption.

On the downside ray tracing is a computational expensive method with regard to rendering due to the large number of rays that must be traversed and the many intersection point calculations. Yet, recent research has shown that various strategies can sufficiently speed up this process (Sakalauskas (2007); Parker et al. (2010); Wald et al. (2017)), making ray tracing a viable alternative to rasterization. This particularly holds true if models become large, as it applies to terrain heightfields. In general, there are four aspects that can facilitate interactive ray tracing: data structures, ray coherences, approximation, and hardware improvements.

**Data Structures** To achieve acceptable frame rates, auxiliary data structures are necessary. Aim of these data structures is to simplify or reduce the number of intersection point calculations. Most typically, Bounding Volume Hierarchies (BVH), kd-trees, and grids (Wald et al. (2007)) are used to overestimate the geometry. Through this, the geometry can be excluded from intersection tests at an early stage. BVH can be updated very fast, but do not adapt to the geometry as good as kd-trees do. Alternatively grids are used to produce a fixed, uniform subdivision of the space, though they do not provide a hierarchy and no adaptability at all (Ize et al. (2006)). However, this structure fits well to terrain heightfields. Especially the approach of Tevs et al. (2008), who extend the grid by a hierarchy to create a so-called Maximum Mipmap (MM) data structure, similar to a quad-tree, has been proven to facilitate interactive ray tracing of heightfields.

**Ray Coherences** Exploiting coherences is another important acceleration technique for ray tracing. For instance, Lee and Shin (1997) proposed to use vertical ray coherence for terrain heightfields. It defines ray lines that are the intersection of the projection plane on the one hand and a plane perpendicular to the terrain plane on the other. Through these ray lines the rays can be advanced faster through the scene. Other approaches reduce intersection tests by tracing multiple rays at the same time, either by combining arbitrary rays, for instance ray packets (Wald et al. (2001)), or by using neighbored rays as beams (e.g., Peterson and Porter (2013)).

**Approximations** Instead of accurate calculation, approximations for intersection points or shading information can be exploited to further accelerate ray tracing. For instance, Policarpo et al. (2005) proposed an efficient surface-ray-intersection algorithm for heightfields that is based on a combination of uniform and binary search instead of an exact calculation. Though this can cause visible artifacts, this problem is later solved through relaxed cone stepping Policarpo and Oliveira (2007). Akimoto et al. (1991) use image-based approximation by reducing the number of rays by adaptive undersampling. This method interpolates the color of pixels for regions with low contrast or smooth color gradients without tracing rays. Other approaches simplify the rendering by only using parts of the ray tracing algorithm. To this end, either the ray casting is replaced with rasterization (e.g., Stamminger et al. (2000)) or the tracing of rays to the light sources is replaced by precomputed lightmaps.

**Hardware** Besides algorithmic improvements, hardware evolution is one of the major factors that facilitates interactive ray tracing. Focus of hardware improvements are efficient intersection tests, which contribute up to 75% to the total run time, and parallelization, which allows tracing multiple rays at the same time. To this end, numerous customized hardware solutions, for example, Nah et al. (2011), Lee et al. (2013) and Nah et al. (2014), have been developed. Aila and Laine (2009) and Dick et al. (2009), on the other hand, propose ray tracing approaches that are completely realized on common GPUs, utilizing the vast amount of parallel cores.

In sum, many tailored approaches for accelerating ray tracing have been developed to facilitate real-time applications. However, rendering approaches also have to take close interrelations between rendering quality, rendering time, and available resources into account. For instance, fully illuminated parts of the terrain in the front need to be rendered in high quality, while parts of the terrain that are in the dark or far away could be rendered with less effort to decrease rendering time. Alternatively, terrain ray tracer on mobile hardware might prioritize resource efficient rendering over image quality. To this end, scalable ray tracer that can flexibly adjust quality and performance are necessary. However, current solutions are often tailored to specific requirements and do not allow customizations during the rendering process. Hence, there is still need for novel approaches. In Section 6.2 we will propose a more scalable approach based on functional programming and introduce a flexible ray tracer for rendering terrain heightfields.

### 2.1.2.3 Design Choices

Aim of the terrain visualization is primarily the communication of its topography. This involves the description of the shape and features of the surface as well as their naturally (e.g., vegetation or rivers) or artificially (e.g., streets or buildings) permanently connected objects. Depending on the task at hand, terrain visualizations can focus, for example, on conveying the relief, specific landforms, or the infrastructure. The requirements on presentation designs of terrain depictions are as different as the areas of application. However, the effectiveness of information communication and thus problem solving and decision making performance varies enormously (100:1) with different presentations (Hanrahan (2011)). Thus, the design choice plays a significant role in terrain visualization and various tailored forms exist.

Regarding the design of terrain representations, we must first decide which dimensionality we should use, meaning whether to show the terrain as a 2D map or as a 3D surface. Afterwards, we have to decide what representation style should be used. This issue refers to the desired look and feel of the representation and involves questions, such as, should the terrain be presented in a realistic or in a more artistic way, which style is appropriate, and what color, for example, textures or plain color, should be used. In the following, we will take a closer look on these design choices.

**2D or 3D Representations** Regarding the represented geometry, there are two fundamental approaches to visualize the terrain. This is either 2D or 3D. 2D representations are assembled only from 2D graphical elements, such as points, lines, and polygons. Terrain visualizations from 2D graphical elements are called maps. Maps have a long history in spatial visualizations due to their familiarity in real life and straightforward usability. Advantages of maps are, for instance, the avoidance of self-occlusion of the terrain topography

(except for caves and overhangs) and the absence of projections for graphical elements on 2D output devices. However, space distortions may appear because the surface of the earth is curved and uneven, although this might be noticeable only on larger scale. Though maps primarily convey 2D terrain characteristics, for example, the position and size of elements, certain adaptations to the presentation can be used to communicate 3D characteristics, such as height or curvature. To this end, multiple views, labels, or specific presentation styles (cf. Section 2.1.2.3) can be used.

Three-dimensional terrain representations, on the other hand, utilize 3D graphical elements, such as solids or freeform-surfaces. 3D terrain depictions are typically more complex, but communicate the three-dimensional structure more intuitively. To convey the topography, depth cues, such as shades, but also projection, parallax and occlusion can be exploited.

The fundamental choice whether to use 2D or 3D applies to the context of terrain representations as well as to data visualizations in general. Both 2D and 3D representations have immanent advantages and drawbacks. Related research indicates that the answer to the question of using 2D or 3D depends on various factors such as the task, data complexity, display technology, or application context.

One exemplary study refers to the relation of available screen-space and the number of items to display. Vion-Dury and Santana (1994) presented a case study that focus on the application of 2D and 3D for visualizing object-oriented systems. Here, the perception of a given presentation is evaluated using the ratio of the number of objects perceived and the total number of objects. For a given display resolution ( $400^2 px$ ), their research indicates the existence of a boundary value ( $\sigma \approx 250$ ) at which 3D presentations exhibit higher context perception than 2D presentations .

Further researches on the effects of 2D and 3D on spatial memory shows no significant differences (Cockburn (2004) and Cockburn and McKenzie (2002)). Tory et al. (2006) conducted a number of experiments of 2D, 3D, and combined visualizations for estimation tasks of relative positioning and orientation as well as region selection. Their results show that 3D can be effective for approximative navigation and relative positioning, but 2D is more suitable for precise measurement and interpretation.

Moreover, combining 2D and 3D achieves a good to superior results and increases confidence during problem solving. These examples support the thesis that the question whether to use 2D or 3D for data visualization is difficult to decide. Especially, the respective advantages and disadvantages of 2D and 3D presentations, including occlusion, clutter, distortion, or scalability, have to be considered for an effective visualization design. In Section 4.1 we will investigate this issue in further detail with respect to spatial data visualization in terrain.

**Presentation Styles** In literature there are many different types of terrain representations. Fundamentally, three different ways to depict the terrain topography can be distinguished: sketching, shading, and texturing. Sketching, a technique that originates from traditional painting, is used to depict the terrain in an artistic or stylized manner. Shading techniques primarily consider the light distribution in the scene and computes shadows in order to convey a realistic representation of the terrain surface. Texturing, in turn, uses precomputed color information that is draped over the entire terrain, for example, to give an impression of surface material or vegetation.

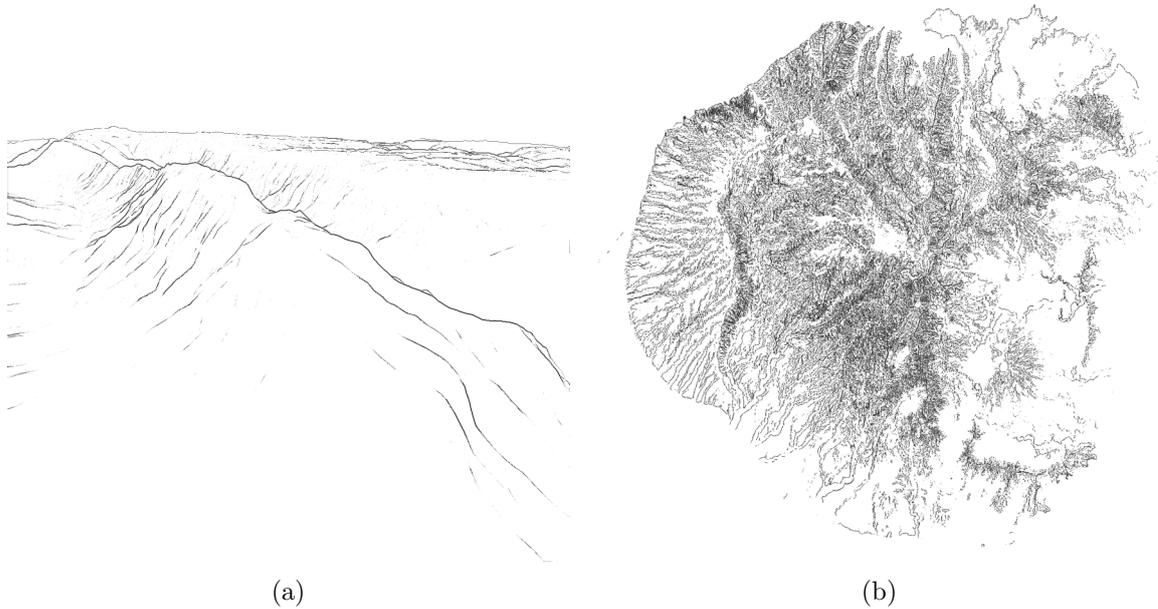


Figure 2.3: Terrain representation through sketching for (a) a 3D terrain and (b) a 2D map.

**Sketching** Sketching of terrain surfaces comprise various techniques to illustrate the topography by strokes, stipples, or other artistic means. Originating from manual visualizations in cartography, today sketching is an independent field of research in non-photorealistic rendering (Kennelly and Kimerling (2006a)). An excellent review on artistic sketching for terrain models is given by Dowson (1994). In general, sketching aims to highlight the most important characteristics of the surface. A simple method would be the rendering of silhouettes (Hertzmann (1999)). A silhouette is defined as the edge between surface points that face towards the viewer and surface points that face backwards. In other words, silhouettes are drawn at those surface points which normal vector faces perpendicular to the view vector. As illustrated in Figure 2.3a, drawing silhouettes already creates good cues on the terrain topography, though this technique is inherent only applicable on 3D scenes. To generate a more detailed presentation of the terrain, additional edges, such as ridges, must be shown. Ridges can be identified by using information about depth and variations of surface normals, for instance through the curvature (Weinkauff and Günther (2009)). As Figure 2.3b shows, this method is also applicable on 2D maps. Suggestive contours, proposed by DeCarlo et al. (2003), is another sophisticated sketching technique. This approach draws lines on normally clearly visible parts of the terrain, where a true contour would first appear with a minimal change in viewpoint.

Besides contours, hatching (Hertzmann and Zorin (2000)) is another artistic render approach. Hatches are short lines that either indicate the direction of slopes in terrain heightfields (especially in 2D presentations), or are used to indicate shaded surfaces, as shown in Figure 2.4. In this context, hatches serve the same purpose as stipples (Lu et al. (2002)). Since hatches and stipples also communicate a certain vagueness, they are today less often used for terrain visualization, though still popular in non-photorealistic rendering for its artistic effect.

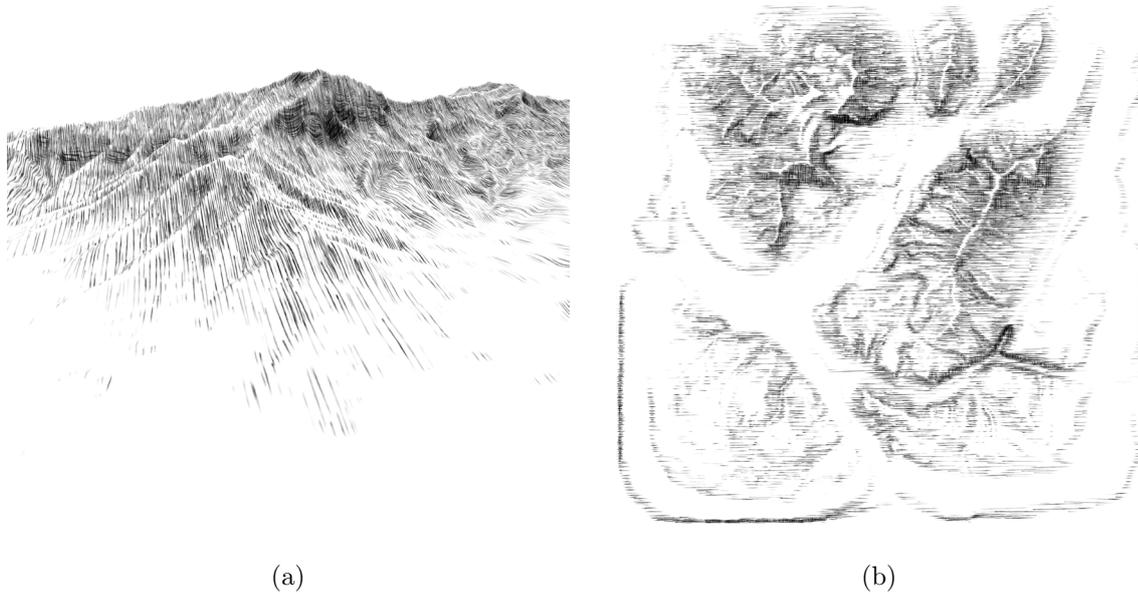


Figure 2.4: Terrain representation through hatching for (a) a 3D terrain and (b) a 2D map.

**Shading** Shading of the terrain surface relates to the lighting calculation of the scene in order to convey the structure of terrain through shadows. Numerous shading techniques exist that vary largely in complexity and quality. Fundamentally, two approaches can be distinguished, local illumination models and global illumination models. While local illumination models calculate the lighting intensity of a surface point only by means of its geometric characteristics, global illumination also considers other elements of the surface in the surrounding, and thus provides real shadow casts.

In the area of terrain rendering, one of most frequently used local illumination technique in terrain visualization is hill shading, which originates from manual drawn, traditional maps (see Figure 2.5a). Yoëli (1967) applied hill shading, using the cosine between an imaginary, perpendicularly or obliquely placed light source and the surface normal to scale the intensity of the surface color by Lambert’s cosine law, and thus making hillsides appear darker than areas that are flat. Instead of Lambert’s cosine law, other local illumination models, such as Phong reflection model, can be applied, but there are only little differences in quality and performance. Since a single light source can sometimes be insufficient to highlight local structures (Zakšek et al. (2011)), some approaches use multiple light sources (e.g., Hobbs (1995)).

Global illumination (see Figure 2.5b) considers not only the direct illumination by a light source but also the light distribution within the scene, which makes terrain scenes appear more realistic. One popular method, which calculates high-quality global illumination, is ray tracing (cf. Section 2.1.2.2), though it is computational very expensive. Thus, often approximative global illumination models are applied. Prominent representatives are ambient occlusion (Pharr and Green (2004)) that focus on diffuse interreflections between the surface elements, openness (Doneus (2013)) that varies the intensity of the surface color depending on the amount of occluded sky, and ambient aperture lighting (Oat and Sander (2007)) that considers the occlusion of the sky as well as the occlusion of direct sunlight, which facilitates the generation of mild shadows. More sophisticated methods for global illumination are, for instance,

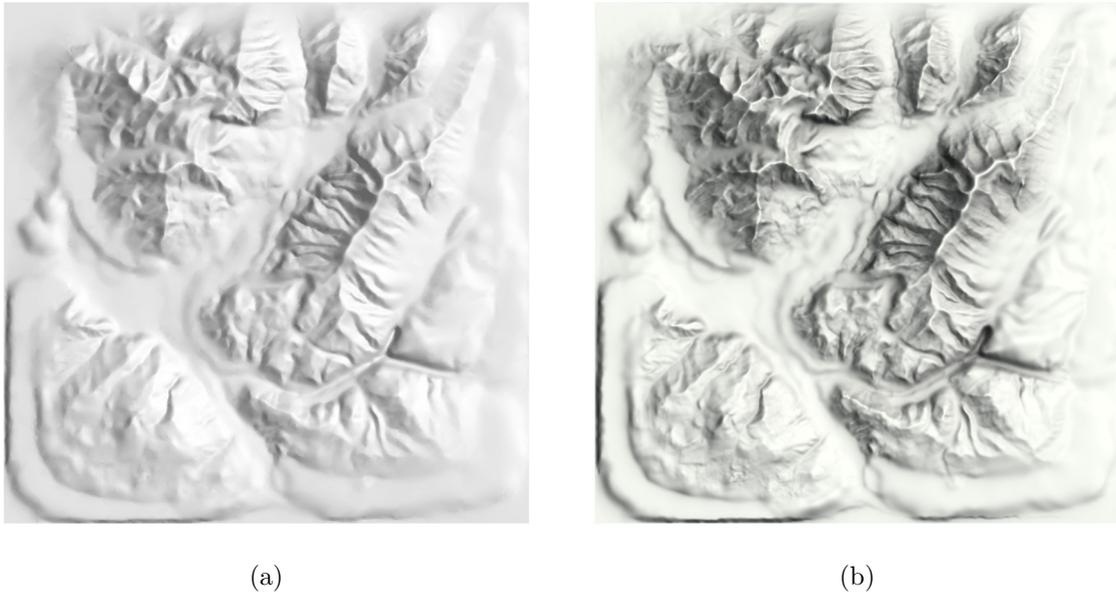


Figure 2.5: Terrain representation through shading by (a) a local illumination model and (b) a global illumination model.

precomputed radiance transfer by Sloan et al. (2002) and fast global illumination by Nowrouzezahrai and Snyder (2009).

**Texturing** Textures provide color information for the terrain model that are mapped onto its surface. This color information can be captured from the real world, for example, through aerial images (see Figure 2.6a), or they can be precomputed, for instance based on given material or on artistic styles. Other approaches use textures with specific coloring, for example, based on height information, so-called hypsometric tints (see Figure 2.6b) or surface gradients (e.g., Local Relief Model (Hesse (2010))).

Though sketching, shading, and texturing are distinct types of terrain representations, there are also approaches that combine techniques in order to create different effects. For instance, Stevens (1980) and Bolton et al. (2007) have investigated how textures of sketched lines, points, or grids can be used to improve surface perception. Moreover, textures are often combined with shading techniques to create detailed or realistic looking surfaces, and some sketching approaches (e.g., Kennelly and Kimerling (2006b)) incorporate illumination information to create terrain illustrations.

In general, different presentation styles have a significant impact on how the terrain is perceived and thus influence which information about the surface are predominantly conveyed. This knowledge, in turn, opens the path to a flexible terrain visualization that can be used to scale the presentation design in order to fulfill the visualization goal. However, there is little work on the question how to use this scalability to adapt to different visualization tasks, for instance. In Section 4.2.1, we will further investigate these different design options and systematize them in order to approach this issue.

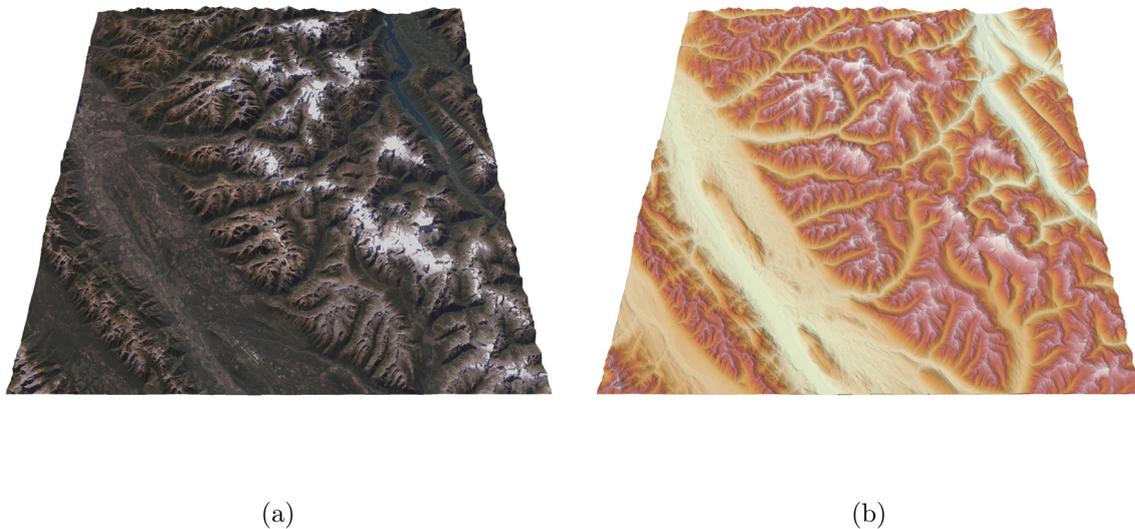


Figure 2.6: Terrain representation through textures using (a) an aerial image and (b) hypsometric tints.

## 2.2 Spatial Data Visualization

Spatial data analysis is a fundamental field of research for cartographers, geo-scientists, and Geo-Information Systems (GIS) and is indispensable in our world today. While acquisition, storage, and accessibility of spatial data has become increasingly easier, processing, preparation, and presentation remain challenging. However, as Keim et al. (2010) have worked out, proper data communication is critical since information persisting in raw data remains uninterpretable and useless for the user. Visual representation of spatial data facilitates the comprehension of information and makes it accessible by interactive means.

### 2.2.1 Definition of Spatial Data

All spatial data are given in a spatial reference space ( $\mathcal{R}$ ), which, in our case, corresponds to the terrain. These data are defined within the terrain by data points. The location of the data points are determined by their spatial coordinates, which are called the independent variables of the data (Keller et al. (1994)). The number of independent variables determines the dimensionality of the data. Typically, we speak of one-dimensional data (e.g., data along a time line or a given path), two-dimensional data (e.g., data on a map), or three-dimensional data (e.g., data in a volume). Also higher dimensional data exists, such as four-dimensional spatio-temporal data. In general, we can speak of  $n$ -dimensional data with  $n$  independent variables. It is necessary to note that “ $n$ -dimensional” ultimately refers to the dimensionality of the coordinates, used to unequivocally determine the location in the reference space. It does not necessarily correspond to the dimensionality of the data’s representation, which could independently consist of 0D, 1D, 2D, or 3D primitives.

There are many different categorizations of data, for example, by Shneiderman (1996), Keim (2002), and Kehrer and Hauser (2013). These categorizations are based on various data characteristics, such as dimensionality, space- or time-dependence, structure, or prove-

nance. In this work, we focus on spatial data. Furthermore, we distinguish spatial data into *geometrical data* on the one hand and *numerical data* on the other. Geometrical data are defined by a concrete shape, which can be zero-, one-, or multidimensional. Typical geometrical data would be points of interest, roads, or buildings (e.g., Beck (2003)). In contrast, numerical data have no inherent shape and thus must be mapped to appropriate graphical elements during visualization process.

Spatial data typically possess one or more attributes, also referred to as the dependent variables. These attributes are observed, measured, or computed either artificially (e.g., simulation results) or in the real world (e.g., temperature). The separate attributes define the  $m$ -dimensional attribute space  $\mathcal{A}$ , where  $m$  is the number of individual attributes. If there is only one attribute, we speak of *univariate* data, whereas data with more than one attribute are called *multivariate* (Keim (2002)).

The individual attributes are characterized by their value range. In general, we distinguish qualitative and quantitative attributes. Qualitative attributes are given in a non-metric scale. If no order relation is defined, we call the attributes categorical or nominal, otherwise we refer to ordinal scales. Categorical attributes would be, for instance, names of mountains, where we can test for equality, but there is no real order between the names. Ordinal data, in turn, can be ordered though we cannot express their value in an actual number nor do arithmetics. An example would be the categorization into calm, breeze, and storm. Quantitative attributes, on the contrary, manifest into concrete numbers that support arithmetic comparison. Examples would be the height of mountains or wind speed measurements.

Another important characteristic of spatial data is their scope that describes the space, for which the data's attributes are in effect. Typically, the scope is a 0D point, meaning the data apply only at that individual position. However, higher dimensional scopes exist. Thus, data points can be effective along a 1D line segment (e.g., a trajectory), a 2D area (e.g., a terrain surface), or a 3D volume (e.g., a cloud).

### 2.2.2 Visualization Process

Aim of a visualization is the (interactive) communication of the data's characteristics by visual means. To this end, the data are transformed into a visual presentation, while considering three fundamental principles: expressiveness, effectiveness, and appropriateness (Mackinlay (1986); Schumann and Müller (2000)). The expressiveness principle imposes that a visualization should show data as authentic as possible and only the information that exists within the data. Effectiveness, in turn, means that a visualization should convey the information in the best possible way. Nevertheless, this is difficult to achieve, since it heavily depends on the specific end-user. The last maxim, the appropriateness, alludes to the trade-off between the benefit of a presentation and the effort of its creation. The process of creating expressive, effective and appropriate visualizations is challenging, particularly in consideration of large and complex spatial data.

In order to formalize the visualization process, Haber and McNabb (1990) proposed the well-known visualization pipeline, which is shown in Figure 2.7. This pipeline transforms the data into visual presentations through three phases, that are, *data preparation*, *mapping*, and *rendering*.

The first phase, the data preparation, transforms the raw data in order to adjust them for the presentation. During this phase data gaps may be completed by interpolation, the amount of data may be reduced by filtering, or high-frequent data may be adjusted

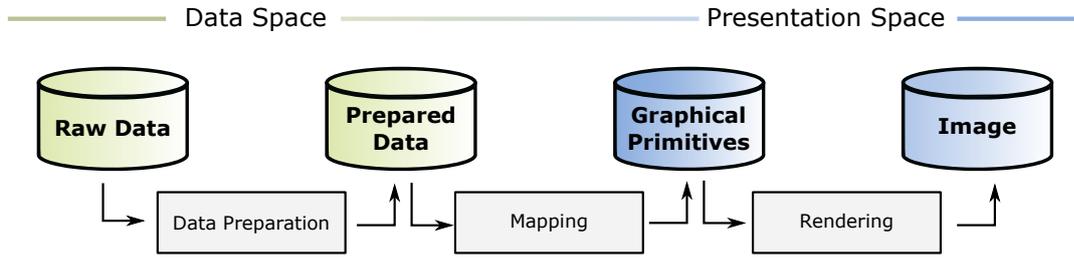


Figure 2.7: The Visualization Pipeline (adapted from Haber and McNabb (1990))

by smoothing. Depending on the data’s characteristics, however, not all procedures are admissible. For example, the interpolation between sparsely distributed data points might violate the principle of expressiveness. An overview of methods for data manipulation and computation with regard to spatial data visualization is given by Andrienko and Andrienko (2006).

The second phase is mapping. This phase is the core of the visualization, as it constitutes the transition from the data space into the presentation space. Here, the preprocessed data are transformed to graphical primitives, such as points, lines, or planes and to their characteristics, for example, color or texture. With regard to geometrical data, the shape of the graphical primitives are predefined. For numerical data, however, the graphical primitives are freely selectable and the choice primarily depends on how effective the encoding can convey the information. This encoding is not only based on shape, but on multiple visual channels, referred to as *visual variables* that have been initially formalized by Bertin (1983). Originally, he distinguishes seven options: position, size, shape, value, color, orientation, and texture. However, the number of visual variables varies for different approaches in literature. For instance, Mackinlay (1986) adds new variables, such as connection and containments, and also subdivides existing variables, such as size divided into length, area, and volume. Recently, Hall et al. (2016) further stated depth, illumination, transparency, blur, saturation, and motion as graphical marks that can express the data.

When considering Bertin’s visual variables it is conspicuous that in terms of spatial data not all variables can be used arbitrarily. Position, for instance, is almost always used to encode the independent variables (i.e., the coordinates) of the data. Similarly, the visual variables size and shape typically imply the scope of spatial data, and are scarcely used to encode the data attributes. This circumstance has to be considered when encoding spatial data. For more details on data encoding and visual variables we refer the comprehensive book by Munzner (2014).

The final phase of the visualization pipeline is the rendering step that works on presentation level and creates the final image. Here, methods from computer graphics and image processing play an important role and especially technical challenges, such as image quality and render time have to be considered. However, the preservation of reliability of rendered objects in terms of accuracy and perceptibility is also an important objective. Thus, methods for level-of-abstraction (Trapp (2013)), occlusion management (Röhlig et al. (2017)), or visual emphasis (Hall et al. (2016)) may be applied during this step.

Haber and McNabb’s visualization pipeline allows an abstraction of the visualization process to the level of data space and presentation space. The transition between them is formed by the mapping phase, where the data are encoded to visual variables. Since

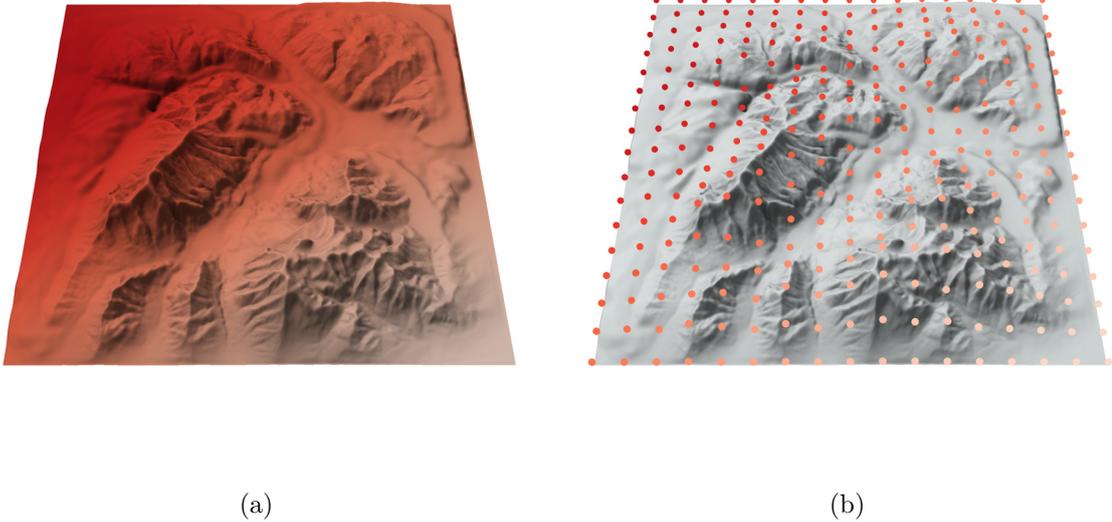


Figure 2.8: Visualization of spatial data (air pressure distribution) through (a) intrinsic technique (color map ranging from white to red) and (b) extrinsic technique (glyphs using the same color map).

spatial data are usually visualized together with the terrain, it has to be incorporated into the visualization. This leads to two fundamental approaches for the encoding: *intrinsic* techniques, where the data are mapped onto the terrain surface and *extrinsic* techniques, where additional primitives are used.

**Intrinsic Techniques** When spatial data are encoded into the visual variables, such as value, color, orientation, or texture, we require graphical primitives that corresponding characteristics can be altered. For this purpose, intrinsic presentation techniques utilize the depiction of the terrain. A common intrinsic technique, for instance, is the coloring of the terrain surface with respect to the data’s attributes by means of an appropriate color scale. [Figure 2.8a](#) shows such an approach. By using intrinsic presentation techniques, the spatial mapping between the data and the reference space is communicated intuitively, but the number of visual variables that can be used for encoding is limited to the terrain surface. Thus, intrinsic techniques are particularly suited to visualize only one or two attributes of data. Since typically each surface point of the terrain can be modified, intrinsic techniques fit well with a dense set of data points.

**Extrinsic Techniques** Instead of altering the terrain surface, extrinsic presentation techniques use supplementary graphical primitives to represent the data (see [Figure 2.8b](#)). According to their dimensionality in presentation space, we can distinguish 2D and 3D graphical primitives. Two-dimensional primitives would be planes. Points and lines are also considered 2D primitives. This is, because these primitives cannot be represented directly due to their lack of a spatial extent. Instead they must be mapped to small 2D areas in presentation space. Three-dimensional prim-

itives would be solids or free-form-surfaces. For data representation often multiple elements are combined to form icons, billboards, and glyphs. With extrinsic presentation techniques visual variables can be used more unrestrictedly than with intrinsic presentation techniques. In that respect, carefully designed glyphs can facilitate the visualization of multiple attributes. Moreover, extrinsic techniques are typically more prominent and easy to read, though a large number of additional graphical primitives would be prone to occlusion. Thus, extrinsic techniques are more suited to represent data sets with only few observation points.

Both intrinsic and extrinsic techniques are widely used to visualize spatial data and in sum a vast number of different presentation techniques exist. It would be beyond the scope of this work to give a comprehensive overview of all existing approaches. However, in [Section 4.2.2](#) we will investigate how in general intrinsic and extrinsic techniques can be applied to communicate characteristics of spatial data and how to facilitate scalability using selected techniques in data visualizations.

## 2.3 Uncertainty Visualization

The interpretation of spatial data strongly depends on knowledge about the data’s quality. Hence, visualizing uncertainty in the data is an important objective ([Pang et al. \(1997\)](#)). In the early 1990s, [MacEachren \(1992\)](#) and [Goodchild et al. \(1994\)](#) already pointed out the importance of showing uncertainty in geo-science. However, in many visualizations information on quality is missing. As [Brodlie et al. \(2012\)](#) noted, “we often encounter error bars on graphs, but rarely see the equivalent on contour maps”. Yet, a representation that does not communicate uncertainty may suggest that the depicted data are perfectly accurate, what they usually are not. This, in turn, can lead to incorrect assumptions and—at worst—to wrong decisions. Arising from this observation, the visualization of uncertainty information has become an important field of research.

### 2.3.1 Definition of Uncertainty

The term uncertainty covers a complex concept that is ambiguously defined and somewhat difficult to grasp. [Gahegan and Ehlers \(2000\)](#) describe uncertainty as inconsistency and doubt about data. [Thomson et al. \(2005\)](#) state that uncertainty implies an imperfection in the user’s knowledge about a dataset, process, or result. This lack of knowledge is, according to [J. Hunter and F. Goodchild \(1993\)](#), responsible for hesitancy in accepting such data without caution. A more formal definition is given by [Draper \(1995\)](#), who proposes uncertainty as the unknown quantities  $y$  on the basis of known quantities  $x$  based on a model  $M$  that formulates their relationship. In general, many authors ([Goodchild et al. \(1994\)](#); [Klir and Wierman \(1999\)](#); [Griethe and Schumann \(2006\)](#)) consider uncertainty as a composition of different facets, such as:

- *errors* that describe a discrepancy from a true value,
- *imprecisions* if the resolution of data is coarser than required,
- *inaccuracies* if data are specified in a vague interval,
- *linage* of data, e.g., data from (un)trustworthy sources,

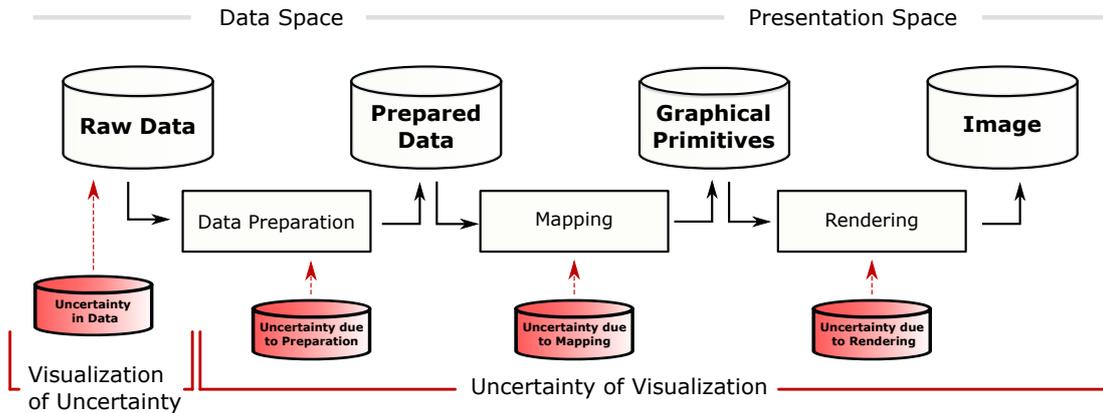


Figure 2.9: Uncertainty Reference Model (adapted from Brodlie et al. (2012))

- *non-specificity* if a data attribute lacks distinctions for data values,
- *noise* if there are background influences.

Some of these facets such as errors are measurable and objective, while others such as lineage remain more elusive and subjective (Pang (2001)). However, while these facets may partially differ significantly, they all have in common that they represent a certain degree of data quality.

As the number of definitions shows, there has gone significant effort into approaching uncertainty in data visualizations. There are several surveys that provide excellent overviews and taxonomies for the visualization of data and uncertainty, such as Thomson et al. (2005) and Potter et al. (2012). The survey by MacEachren et al. (2005) is particularly relevant with regard to spatial data, as it considers the different types of uncertainty with respect to the reference space and the attribute space.

Sources of uncertainty are numerous and ubiquitous, starting from the acquisition of data on to the processing up to the visualization. During acquisition, inaccuracies in measurement, misconceptions in simulation, or inadequate storage of the data can compromise their quality. Further uncertainty can arise from data processing in preparation of the visualization, for example, through normalizations, interpolations, or approximations. And finally, uncertainty can occur during the data visualization, for example, due to projection or quantization. To categorize sources of uncertainty, Brodlie et al. (2012) propose an extension of the visualization pipeline, the uncertainty reference model (see Figure 2.9). By this, they distinguish between *visualization of uncertainty* that originates from inaccuracies in data acquisition and processing, and *uncertainty of visualization*, which originates from inaccuracies occurring during the visualization process.

Uncertainty is often considered for individual data values (e.g., inaccuracies in the scalar), but can also affect a certain attribute (e.g., measuring tolerance for a data attribute), certain data points (missing or interpolated data points), and even whole data sets (e.g., distrust of data due to its lineage). Thus, uncertainty can be assigned to any component of the data. The actual quantity of uncertainty must be determined by certain measures. To this end, Thomson et al. (2005) propose a certain propagation model, while Klir and Wierman (1999) suggest informal measures, such as plausibility and fuzziness. According to Griethe and Schumann (2006), other common concepts would be error

percentage, distance from true values, standard deviation, or qualitative descriptions, for instance, “estimated by three experts”. The eventually assigned uncertainty quantities form the basis of the visualization that aims to raise the awareness for potential imperfections in the data and thus facilitates a more substantiated visual data analysis.

### 2.3.2 Visualization Process

Due to the importance of uncertainty communication, visualizing uncertainty together with the data has attracted much attention over the last two decades. A recent survey, including a broad overview of existing uncertainty visualization techniques is given by [Kinkeldey et al. \(2014\)](#). To classify these techniques, they identified multiple dichotomous categories in literature, that are:

- *explicit/implicit*, e.g., encoding the uncertainty quantities directly into the visualization or showing multiple presentations of different possible outcomes (e.g., [Deitrick and Edsall \(2006\)](#)),
- *visually integral/separable*, referring to the perceptual separability of the uncertainty values and the data values (e.g., [MacEachren et al. \(1998\)](#)),
- *coincident/adjacent*, meaning that the uncertainty information is visualized in the same or a separate view (e.g., [MacEachren \(1992\)](#)),
- *static/dynamic*, referring to the use of static visualizations or animations (e.g., [Brown \(2004\)](#)), and
- *intrinsic/extrinsic*, e.g., altering existing visual variables or using additional graphical primitives (e.g., [Howard and MacEachren \(1996\)](#); [Gershon \(1998\)](#)).

These categorization are not clearly distinct, but can overlap. For instance, visually integral visualizations techniques bear similar properties as intrinsic visualizations, though there are some exceptions. In correspondence to our categorization of data visualization techniques (see [Section 2.2.2](#)), we will focus on the common distinction of intrinsic and extrinsic uncertainty visualization.

**Intrinsic Techniques** Intrinsic techniques visualize uncertainty by altering existing graphical representations of the data, in particular the visual variables that are not used for encoding the data itself. Beside the common visual variables typical for data visualization, such as position, size, shape, value, color, orientation, and texture, uncertainty visualizations often apply alternatives, such as fuzziness, focus, clarity, transparency, or edge crispness. In principle, all visual variables can be used, but [MacEachren. et al. \(2012\)](#) showed in experiments that some visual variables might be more suitable than others. Though these studies were limited to specially-designed data glyphs (so-called visual semiotics), the experiments do suggest that, for instance, fuzziness and transparency are more intuitive with regard to uncertainty visualization than shape or color. This might be because they convey a certain vagueness and ambiguity. In this sense, the application of noise, fog, or blur (e.g., [Guo et al. \(2015\)](#)) are also frequently used, intrinsic visualization techniques since they manipulate the presentation of the data.

**Extrinsic Techniques** Extrinsic techniques integrate supplementary graphical primitives into the presentation, such as contours (Osorio and Brodlie (2008)) or glyphs (Luo et al. (2003)). Being visually independent of the data visualization, extrinsic presentation techniques for uncertainty evince a higher variability than intrinsic approaches. Studies, such as Drecki (2002), Grigoryan and Rheingans (2002), and Newman and Lee (2004), have found that extrinsic techniques perform either as good as intrinsic techniques or even better in terms of accuracy and effectiveness. MacEachren. et al. (2012), on the other hand, noted that the effectiveness of intrinsic or extrinsic techniques may vary depending on the given task, the type of uncertainty, and the preferences of users.

Regarding the dimensionality of uncertainty presentations, the majority of research of uncertain spatial data has addressed the representation in a 2D context, that is on a map (e.g., Kardos et al. (2005), Sanyal et al. (2010), and Cox et al. (2013)). 3D representations are only rarely considered in the literature. An example is the work by Johnson and Sanderson (2003). However, they do not investigate spatial data. In general, presenting uncertainty in 3D terrain is still a concern of ongoing research. In Section 4.2.3 and Section 5.1 we will take up this subject and examine how uncertainty and spatial data can be visualized together with the terrain.

## 2.4 Scalability Challenge

Among the first who formulated the scalability challenges were Cook and Thomas (2005). They noted that the amount and size of available data grows continuously and that with current technology these data can be computed at ever increasing speed. However, human cognitive abilities cannot change that significantly over time. They call this circumstance the information overload or information glut, meaning that more information is accessible than humans can possibly process. It is a fact, though, that carefully applied technology can facilitate the usage of a higher percentage of natural abilities. To this end, methods that deal with the scalability issue are needed. In the last two decades scalability has become one of the top challenges in information visualization and much effort has gone into researching adequate approaches.

The aim of scalable information visualization is to help closing the gap between the growing complexity of data and the human ability to understand them. In fact, the analysts do usually not need to investigate the entire data. But instead, they must look at the right pieces of information extracted from a subsets of the data. These must be prepared and presented in a proper way to comprehend and further utilize them in order to gain new knowledge, create hypotheses, and make decisions. In sum, the scalability problem ultimately boils down to the questions: How to extract the relevant information from large data and how to visually communicate them with respect to the given requirements. This thesis aims to provide initial answers to this question in the context of spatial data visualization in 3D terrain.

In general, the term scalability refers to multiple aspects. Lu et al. (2017), for instance, divided the problem into *visual scalability*, meaning the difficulty of visualization techniques to cope with data sets of increasing size, secondly *interactive scalability*, which refers to dynamic changes of large data sets, and thirdly *data efficiency* that refers to the issue of organizing data with different scales in an effective way. In an analogous manner, Cook

and Thomas (2005) and Robertson et al. (2009) distinguish the following multiple facets of scalability:

- *Information Scalability* that implies the capability to extract information from massive data, for example, by filtering and abstraction, and to the ability to deal with dynamic data sets.
- *Visual Scalability* that refers to the capability to communicate information from large-scale and complex data that is characterized, for instance, by a large number of data points, a high dimensionality, various data types, and multiple data sources. Since the number of representable elements in a visualization is limited, the mapping of these multifaceted data is difficult.
- *Display Scalability* that is the ability of visualizations to adapt to the available screen sizes, instead of assuming fixed dimensions. Today screen sizes can vary from large wall-sized displays to small mobile displays. Scalable visualizations would adjust to different conditions and still communicate the required amount of information.
- *Human Scalability* that refers to visualizations that adapt for different user groups whose experience, ability, and background knowledge may vary. It also considers the flexibility to respond to single-user scenarios in the same way as to collaborative-work scenarios.
- *Computational Scalability*, which is the capability to automatically process very large data sets in order to analyze them and/or to prepare them for visualization.
- Further challenges, including *software scalability*, *temporal scalability*, *cross-scale issues*, *security issues*, and *language issues*.

Accordingly, scalability in information visualization refers to the capability of a presentation to visually communicate large-scale data by interactive means, taking dynamically changing conditions, such as the task, the user, the hardware, or the data itself, into account. This is not an issue limited to a single presentation, but affects also interactive visualizations and even complete visualization tools (Eick and Karr (2002)).

The identification and extraction of the actual relevant information is a difficult process that cannot be fully automated. Instead, the human must be involved into the data selection process at an early stage. To this end, various interactive visualization approaches exist. Among them, the concept of Focus & Context is a good example to illustrate the aforementioned scalability issues.

*Focus & Context* has arisen from the need of showing an overview of the data as well as tailored detailed information where the user demands them, which leads to the multiple challenges of scalability. In general, the quantity of data is too large to be shown at once, thus the data must be reduced, for example, through filtering (cf. information scalability). How much of the data has to be hidden depends on multiple factors, such as the available display area (cf. display scalability) and processing capacity (cf. computational scalability). During the visual analysis the user will identify regions of interest where more details are required to continue the investigation (cf. user scalability). Typically, the demanded details will be shown in a separate view, which relates to the concept called Overview & Detail (Zhu and Chen (2006)). In contrast, Focus & Context tries to seamlessly combine these

two separate views into one presentation. By this, both the relevant parts of the data (focus) and an overview of related data (context) are simultaneously visible. Beside the two regions, focus and context, an intermediate region is additionally used in some approaches (e.g., [Preim and Bartz \(2007\)](#)) to facilitate a smoother transition. The most typical Focus & Context approach would be interactive lenses ([Tominski et al. \(2014\)](#)) that determine the focus region by selecting a spatial region, emphasizing this focus region and simultaneously suppressing the adjacent region. To this end though, multiple data abstraction levels are necessary (cf. visual scalability).

In sum, Focus & Context is a well-known concept that tackles multiple scalability challenges. How these challenges are solved in detail depends on the individually applied method. A comprehensive overview on various Focus & Context strategies are given by [Cockburn et al. \(2009\)](#) and by [Hall et al. \(2016\)](#). Typical approaches consider the focus region either in the data space ([Doleisch et al. \(2003\)](#); [Jankun-Kelly and Ma \(2003\)](#)) or in the terrain ([Wang and Chi \(2011\)](#); [Trapp \(2013\)](#)). However, in some application fields the characteristics of data as well as geometric features of the terrain need to be analyzed together. Hence, Focus & Context must also be applied to the data space as well as to the terrain. We will address this problem in our approach of using Focus & Context for spatial data in terrain in [Section 5.2](#).

## 2.5 Summary

In this chapter, we analyzed the fundamentals and challenges of spatial information visualization regarding three information aspects: the terrain, the data, and uncertainty. With regard to visualizing these individual aspect, much research has been conducted. However, approaches that show terrain, data, and uncertainty together in one holistic visualization are scarce. As one example, [Wittenbrink et al. \(1996\)](#) visualize uncertainty in wind data sets embedded into a coarse 3D terrain presentation. For communicating uncertainty information, they focus on designing wind glyphs. Moreover, [Davis and Keller \(1997\)](#) propose a 3D visualization of slope stability. In their work, uncertainty is visualized through animation, which is a viable approach, though in some cases animation might introduce problems such as change blindness. Static representations are used by [Schmidt et al. \(2004\)](#) for underwater scenarios and by [Kunz et al. \(2011\)](#) for natural hazards over terrain. Both propose solutions for presenting data and uncertainty, but no specific techniques for terrain. Moreover, all these visualizations are fixed and cannot be adjusted depending on the application context. However, we have worked out that adaptive visualizations that scale with the current task, with the available data, and with given hardware are required. In this context, the data computation as well as the data communication have to be considered. Thus, in the following we will investigate how to develop scalable visualizations for spatial data in terrain.



## Chapter 3

# Approach to Scalable Visualization of Spatial Data in Terrain

We showed in the previous chapter that visualizing spatial data in terrain faces various challenges. A lot of these issues are tackled by highly tailored state of the art visualization techniques. These techniques are usually attuned to particular requirements, such as the task at hand, while depending on underlying constraints, such as data size or hardware requirements. In this sense, these customized techniques are most effective in their particular application domain. However, they often face problems when conditions and constraints are not static. As a matter of fact, the principle commonly applies that the more specific the technique, the more difficulties occur when attempting to apply them to changing situations. First and foremost, this is because specialized techniques benefit precisely from the particular conditions they are tailored to. If these conditions change though, their advantage declines and it takes increased effort to adapt to new conditions. For this purpose, we need appropriate, scalable approaches.

This chapter is dedicated to substantiate and outline our approach of scalable visualization of spatial data in terrain. To this end, we will first give a short introduction to our particular research domain and explain their accompanying issues. From this, we will deduce the need for a scalable approach. Afterwards, we will describe the prerequisites on scalable visualizations, before finally outlining the procedure of our approach.

### Contents

---

3.1	Demonstrative Example . . . . .	30
3.2	Scalability . . . . .	32
3.3	Prerequisites on Scalable Visualization . . . . .	33
	3.3.1 Presentation . . . . .	33
	3.3.2 Processing . . . . .	33
3.4	Approach . . . . .	34

---



Figure 3.1: Exemplary synthetic vision display taken from Arthur et al. (2005).

### 3.1 Demonstrative Example

There are various fields of application for spatial data and terrain visualizations. As a starting point for our research, we collaborated with domain experts from the aviation sector (Richter et al. (2018)) with focus on spatial data visualization in terrain for avionic environments. Such visualizations are subject to different requirements, depending on the particular application context. Changing use cases demand reconsiderations and adjustments of the visualization design. In our work, we mainly consider visualizations intended for cockpits in aircraft.

Cockpits of modern aircraft are packed with electronic systems used for navigation, communication, and system management also known as avionics. In this context, the application of Synthetic Vision Systems (SVS) in cockpit displays has become more and more relevant. It typically shows a visualization of the surrounding terrain, and thus provides improved flight safety to the operator. While 2D presentations are well established in this context, 3D visualizations are still rare, though they might derive benefits regarding spatial awareness (Endsley et al. (1998)). Therefore, our research particularly focuses on 3D visualizations. An exemplary 3D SVS is depicted in Figure 3.1. By combining a terrain model database with the aircraft's present position, the SVS gives the operator a clear, uncovered view on the current surroundings, which becomes particularly beneficial in situations of low visibility. In order to further assist the pilot, recent approaches do not exclusively show the terrain but also information about buildings, obstacles, and also weather data. Utilizing these data allows to design complex visualizations tailored to specific application scenarios.

**Flight Scenarios** The tasks and objectives of a pilot during a flight vary for different scenarios between the start and the landing of the aircraft. Accordingly, the visualization goal for the SVS changes during the flight. Generally, we distinguish between four scenarios, that are, take-off, cruise flight, landing approach, and go-around. Each scenario, in turn, is subdivided into certain flight phases, which we will briefly describe in the following.

*Take-off* describes the procedure of the lift off until reaching cruising altitude. During this scenario, the aircraft undergoes three phases, that are, start, first climb, and second climb. Objective of the start phase is the acceleration of the aircraft necessary for liftoff. With the liftoff begins the first climbing phase, where the plane is further accelerated to allow for a quick increase in height. Hereby it is necessary to navigate through the terrain along a predefined flight path, while concurrently avoiding obstacles. The second climb is characterized by a particular steep increase until reaching cruising altitude and direction. Due to the initially low altitude during the take-off an aligned visualization needs to particularly facilitate spatial awareness. Moreover, wind conditions must be monitored and taken into account.

The *cruise flight* takes place at a height between 9500 m and 12500 m. Normally, this scenario is not subdivided any further. Typical visualization goals would be the orientation and navigation, while looking out for hazardous weather phenomena and other aircraft.

The *landing approach* is divided into four phases: descent, initial approach, final approach, and landing. During the descent the altitude is decreased. Hereby, it is necessary to get flight details and an overview of present weather conditions in order to potentially adapt the descent. During the initial and the final approach, altitude is further decreased (below 3000 m). Focus of tailored visualizations would—similarly to the take-off—lie on spatial awareness and monitoring environmental parameters. The eventual landing phase starts with the touch down and includes slowing the plane down to roll speed. Thereby, the geometry of the runway, surrounding obstacles, as well as wind conditions are important information.

*Go-around and re-planning* defines the abort of the landing approach and the subsequent climb and re-planning. This might occur due to various safety-related issues, such as an approach that is too high or too fast. An aligned visualization would first have to deal with the fast transition and communicate particularly spatial features of the environment and potential obstacles, including other aircraft. The subsequent re-planning phase is usually done in sufficient height. Here, the criteria for the abort have to be considered and a new approach has to be planned.

As described, tasks and objectives of the pilot changes with different flight phases and the same holds true with the visualization goal. In order to comply with these goals, we require different aspects of data, which will be discussed in the following.

**Data** In the aviation domain, we work with a multitude of data in the aircraft that is recorded, stored, and eventually communicated. With respect to the SVS, the foundation of the visualization are terrain data of diverse sizes and resolutions. The presented part of the terrain and the level of detail varies depending on the current position, view direction, and display parameters. Accordingly, the way the terrain is represented hinge on the given scenario. Similarly, with different scenarios miscellaneous data aspects need to be communicated. Particularly important in this context are various types of weather data, for instance, wind speed and direction that are especially relevant during take-off and landing approach. During cruise flight, on the other hand, hazardous thunderstorm cells are

particularly important. In typical scenarios, weather data are partially transferred from ground control and partially measured by the aircraft's sensors. In both cases they can be subject to sudden occurrences, changes, and disappearances, making the conveyance of the relevant information challenging. Typically, weather data has no inherent shape and thus is categorized as numerical data (cf. Section 2.2.1). In contrast, data about obstacles and trajectories would be categorized geometrical data due to their fixed form. Knowledge about obstacles, including wind power plants and cable cars, is primarily relevant when flying in low altitude. Representations of trajectories can facilitate the flight along predefined paths or help assessing the path of other aircraft. Besides the data attributes themselves, the interpretation of the data strongly depends on knowledge about the data's quality. Hence, visualizing the associated uncertainty is another important objective in this context.

In sum, in aviation a large amount of data is utilized in order to support the pilot's task. Size and complexity of the data, however, can make their communication challenging. In order to reduce the cognitive work load on the pilot, typically well established 2D presentations are applied. However, in order to improve spatial awareness, recently 3D visualizations have come more and more into the focus of research. Yet, with 3D visualizations we need adequate visualization methods in order to facilitate situation awareness on the one hand and prevent visual overload on the other. In addition, the applicability of such visualization methods depends also on the available hardware.

**Hardware** Considering the large amount of data and modern sophisticated visualization techniques, hardware capability has become a decisive factor. At present, proper visual designs are implemented and evaluated using powerful hardware, utilizing fast processing units and a large amount of memory. However, technology is different with displays and graphics engines in cockpits. Due to safety reasons, hardware here is designed to be fail-proof with a large amount of redundancies. Accordingly, there are considerable limitations for the graphics system, which in turn, restricts the options for appropriate visualizations, in particular for 3D. Moreover, cockpit technology can fundamentally change with different technology generations and different aircraft. To that effect, the utilizable visualization methods and techniques depend on the actual capability of the hardware and must be considered when designing the presentation.

## 3.2 Scalability

The previous section described a number of scenarios, where the pilots need to deal with different tasks. In order to perform their task, the pilots require information about their current environment, including their surrounding terrain, potential obstacles, and weather conditions. Tailored visualizations allow to convey these information and thus support the aircraft's operator. However, during the flight the scenarios can change and so does the pilot's task. Consequentially, we need scalable visualizations that are able to flexibly adapt to the changing objectives. Switching between fundamental different presentations, however, can lead to a lost of the mind map and to disorientation. Thus, scenario-depending modifications of the presentation should be done with caution and plausible transitions are needed.

In addition, scalable visualizations do not only need to align to the given task, but also to the relevant data aspects that have to be presented. The relevant data are subject to the given scenario, but can also dynamically change, for instance, if hazardous weather

conditions arise. In such situations, the visualization needs to adapt by highlighting the relevant data in order to facilitate situation awareness. Furthermore, the amount of available data can also exceed the capability of current visualizations. This refers to the visual budget on the one hand, but also to the hardware limitations on the other. Therefore, we need a certain hardware scalability in order to make use of the respective capability of the graphics system. How to achieve such scalability, however, is still an unresolved question, which we will engage with this work. On these grounds, we will analyze the prerequisites for scalable visualizations in the next step.

## 3.3 Prerequisites on Scalable Visualization

In order to fulfill the demand for scalability, we have to consider the requirements on the respective visualization, depending on the given scenario. In general, we can differentiate between requirements on presentation side and requirements on processing side.

### 3.3.1 Presentation

The requirements on presentation side are mainly described by the principles of effectiveness and expressiveness (cf. [Section 2.2.2](#)). The expressiveness principle imposes that a visualization should show all the relevant information of the data, and nothing more. The relevance of information in a visualization depends on both the task and the data. Effectiveness, in turn, means that a visualization should convey the information in the best way possible. This implies that the visualization should exploit the cognitive capabilities of the user based on the characteristics of the data and the visualization goal ([Schumann and Müller \(2000\)](#)).

Principally, a visualization must fulfill both expressiveness and effectiveness. In practice though, finding such a solution is always difficult. Additionally, the requirements are interconnected to the present task and data. This implies that changing conditions also change the requirements on the presentation. A challenge for a scalable visualization is therefore the identification of an expressive and effective presentation for the given conditions and the preservation of the fulfillment of these requirements in changing scenarios. In our particular application context—the spatial data visualizations in cockpit environments—this means finding a suitable solution for the communication of the terrain’s topography, the characteristics of numerical and geometrical data, as well as their inherent uncertainty. Due to the large amount of available visualization methods for each individual aspect and the variety of possible parameter setups, this is an intricate issue. Moreover, in order to create a holistic presentation, we need to show not only a single aspect, but several data aspects together, though currently little research has been conducted to this regard. Finally, we also have to deal with the evaluation of chosen visual designs. To this end, we need means to assess the presentation with regard to the given scenario.

### 3.3.2 Processing

Besides expressiveness and effectiveness, we also introduced a third important criterion for purposeful visualizations in [Section 2.2.2](#): appropriateness. This principle alludes to the trade-off between the benefit of a presentation and the effort regarding its creation. The objective of this principle is to achieve expressive and effective presentation while minimizing its costs. Costs, hereby, includes the workload on processing side, in particular the demands

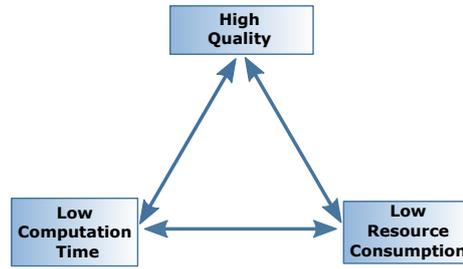


Figure 3.2: Triangle of tension regarding the quality of the visualization, the consumed resources, and the required render time.

for render time and resources. In contrast to the maxims of expressiveness and effectiveness that are imperatively required for a qualitative visualization, appropriateness entails striking a balance between the visualization quality on the one hand and the necessary resources and computation time on the other. Additionally, an interdependency exists between the resource consumption and the required computation time. This is because render algorithms that focus on speed, usually utilize many resources for precomputed data or heavily depend on processing capability, and vice versa.

A simple example would be the presentation of the 3D terrain using appropriate illumination models: In order to create the highest quality for illumination, a full recursive ray tracing would be required, which on the other hand is slow and would not enable render times suited for real-time applications. Instead, a simple phong illumination would be very fast and consume little resources but yields to quite low image quality. A compromise would be, for instance, the use of ambient occlusion that is sufficiently fast and produces images in acceptable quality. As this brief example shows, a triangle of tension exists between the three facets quality, render time, and resource consumption, as illustrated in [Figure 3.2](#).

Obviously, appropriateness heavily depends on the given hardware capability. To that effect, the implementation of the visualization technique must be tailored to the given hardware. If hardware conditions changes, algorithms need to be adjusted. Accordingly, visualizations designed for avionic systems need to be scaled along the different hardware conditions and apply solely algorithms suitable to their capability.

### 3.4 Approach

Goal of this work is to provide a strategy for scalable visualizations of spatial data in terrain that adapts to the given scenario (task), data, and hardware by finding suitable visualization methods meeting the requirements on presentation and rendering side. With respect to our approach, we first need to lay the foundation for scalable visualizations, that is, comprehension about the characteristics of feasible visual designs in order to relate to their suitable application. We will do so as our first step in [Chapter 4](#).

Hereby, we will initially investigate the fundamental question, whether to use 2D or 3D visualizations in [Section 4.1](#). It is a fact that both two-dimensional and three-dimensional presentation techniques exhibit different advantages and disadvantages with respect to various perceptual and technical aspects, such as occlusion, clutter, distortion, and scalability.

To facilitate problem awareness and comparison of distinct visualization techniques with regard to our varying scenarios, we will introduce a novel systematization. This systematization is based on presentation characteristics that enable a categorization with respect to combinations of 2D and 3D presentations of attributes (i.e., the data) and their spatial frame-of-reference (i.e., the terrain).

Afterwards, we will examine the characteristics of concrete visualization methods in [Section 4.2](#). This allows for reconfiguring the presentation based on the given situation and hence to support scalability. We will thereby go into detail of the implications of visualization techniques and their configuration. To this end, we will in turn investigate the properties of methods for terrain, data, and uncertainty, respectively.

In our next step ([Chapter 5](#)), we will focus on scalability regarding the task at hand. For this purpose, we will first address the question how to combine the individual data aspect into one scalable presentation, depending on the given scenario. We will propose a visualization strategy in [Section 5.1](#) based on prioritized presentation of one selected data aspect and presenting the remaining with less detail. We will discuss various design options that allow us to obtain differently prioritized visual representations of the given data and thus to enable a scaling of the presentation along changing tasks.

In the following [Section 5.2](#) we will present a strategy that scales the visualizations by utilizing the concept of Focus & Context. To this end, we allow to specify the focus with regard to both, the terrain and the data, utilizing different problem-related strategies. Based on the specified focus, the associated context is derived automatically. Data within focus are emphasized, whereas context information is shown with less detail, using diverse visualization methods.

At the end of [Chapter 5](#), we will tackle the problem about assessing our scalable visual design. To this end, we propose in [Section 5.3](#) a novel perception-based, automatic evaluation approach for data visualization in terrain. By this means, we will be able to identify areas that attract attention within the visualization and get hints to cluttered images. That allows us to better understand the implication of different configurations and facilitates steering the parameters to convey the relevant information.

In [Chapter 6](#), we will focus on the processing side of scalable visualizations. Initially, we will deal with scalability using different software configuration in [Section 6.1](#). To this end, we will introduce our comprehensive visualization framework TEDAVIS that facilitates the various visualization strategies and configurations proposed in this thesis. TEDAVIS hereby serves two purposes. On the one hand, it demonstrates the feasibility of our approach by unifying the presented concepts and strategies. On the other hand, it enables the design and evaluation of scalable spatial data visualizations in 3D terrain under varying hardware conditions.

Finally, we will consider scalability regarding the hardware in [Section 6.2](#). For this purpose, we will propose a prototypical, scalable hardware architecture for terrain ray tracing that permits the dynamic composition of complex and recursive shaders. Through this means, we are able to map a large number of dynamically created render processes with inter-dependencies onto the graphics system, and thus allowing for automatic scalability based on hardware capability.



# Chapter 4

## Basic Visualization Strategies

In typical application contexts the visual designer is confronted with a large number of choices that can differ significantly from each other. Which design is suitable for the respective application is usually difficult to decide. Hence, the first step of our approach towards a scalable spatial data visualization requires the examination and characterization of feasible presentation design options. To that effect, this chapter is dedicated to the analysis of basic visualization strategies.

The first part of this chapter deals with the decision whether to represent the information by 2D or 3D graphical elements. In this context, we will propose a novel systematization for spatial data visualization and work out their respective properties. In the second part of this chapter, we will examine the diversity of visualization options for terrain, data, and uncertainty respectively.

### Contents

---

4.1	A Novel Systematization of 2D and 3D Presentation . . . . .	<b>39</b>
4.1.1	Systematization . . . . .	39
4.1.2	Characteristics and Discussion . . . . .	41
4.1.3	Conclusion . . . . .	47
4.2	Design Options for Terrain and Data Presentations . . . . .	<b>48</b>
4.2.1	Terrain Visualization . . . . .	48
4.2.2	Data Visualization . . . . .	52
4.2.2.1	Geometrical Data . . . . .	52
4.2.2.2	Numerical Data . . . . .	55
4.2.3	Data Quality . . . . .	58
4.2.4	Spatial Awareness . . . . .	59
4.3	Summary . . . . .	<b>64</b>

---

---

Major parts of [Section 4.1](#) were released in the following publication:

[Dübel et al. \(2014\)](#): Steve Dübel, Martin Röhlig, Heidrun Schumann, and Matthias Trapp. *2D and 3D presentation of spatial data: A systematic review*. 3DVis (3DVis), 2014 IEEE VIS International Workshop on, 2014, 11-18

The author of this thesis contributed to the publication as first author and wrote major parts of the paper. The systematization was conceived and designed through collaborative work. The demonstrating examples were generated by the author.

For this thesis, the publication was rearranged and partially reworded to fit the wording of this work. Moreover, most figures were improved or replaced.

---

Parts of [Section 4.2](#) are based on passages from the following two journal articles:

1. [Dübel and Schumann \(2017\)](#): Steve Dübel and Heidrun Schumann. *Visualization of Features in 3D Terrain* ISPRS International Journal of Geo-Information, 2017, 6, 357:1-357:20

The author of this thesis conceived, designed, and implemented the approach and wrote most parts of the article.

2. [Dübel et al. \(2017\)](#): Steve Dübel, Martin Röhlig, Christian Tominski, and Heidrun Schumann. *Visualizing 3D Terrain, Geo-spatial Data, and Uncertainty* Informatics, 2017, 4

The author of this thesis conceived larger parts of the approach and wrote major parts of the paper. The implementation of the uncertainty presentations is based on the master's thesis *Flexible Visualization of Weather Data in a Spatial Context with Regard to Uncertainty* by Martin Röhlig ([Röhlig \(2014\)](#)), which the author substantially supervised.

For this section, parts of these two publications were extracted and rewritten to fit the concept and wording of this thesis. Moreover, the figures have been replaced.

---

## 4.1 A Novel Systematization of 2D and 3D Presentation

Within the scope of this thesis we were repeatedly confronted with the question whether 2D or 3D presentations are more suitable for a given visualization problem. In Section 2.1.2.3 we showed that this question is difficult to decide, not least because the choice for either alternative has a large impact on the quality of a visualization. This is because the effectiveness of information communication varies enormously with different types of presentations. Specifically, the suitability of a presentation towards a given context influences the speed the information is absorbed, the number of errors made, as well as the comprehension and visual working memory capacity during this process (Plumlee and Ware (2003)).

With regard to the goal of this thesis, that is, achieving scalable visualizations, we require indications when to use which presentation design. Consequentially, a proper systematization can help to design context-aligned presentations. While previous work focused on a global categorization of 2D or 3D techniques, this section goes deeper into detail. For this purpose, we will introduce a systematization that differs between the presentation of the attribute space and the spatial reference space with respect to their dimensionality.

### 4.1.1 Systematization

In Section 2.2.1 we introduced the concept of distinguishing between the data space and the presentation space. The data space is determined by the characteristics of the attribute space  $\mathcal{A}$  (constructed by the dependent variables) and the characteristics of the reference space  $\mathcal{R}$  (constructed by the independent variables). The number of dependent variables hereby defines the dimensionality of the attribute space and the number of independent variables defines the dimensionality of the reference space. For spatial data,  $\mathcal{R}$  is typically two- or three-dimensional, whereas  $\mathcal{A}$  can be multi-dimensional.

The presentation space is constructed from graphical elements used to depict the data. These graphical elements consist of visual variables (e.g., size, shape, color, and texture). Two-dimensional presentations are assembled only from 2D graphical elements (e.g., points, lines, and polygons), while three-dimensional presentations utilize 3D graphical elements (e.g., solids or freeform-surfaces). Given these definitions on the data space and the presentation space, a global distinction of 2D and 3D presentations is no longer sufficient. We rather have to distinguish between the *presentation of the attribute space* and the *presentation of the reference space*.

Accordingly, we propose a systematization with respect to combinations of 2D and 3D presentations of the attribute space  $\mathcal{A}$  and the reference space  $\mathcal{R}$ . To this end, we introduce a notation to index a particular category of the systematization:  $\mathcal{A}^i \oplus \mathcal{R}^j$ , with  $i, j \in \{2, 3\}$  reading:

- $\mathcal{A}^i$ : attributes are visualized using i-dimensional graphical elements,
- $\mathcal{R}^j$ : the reference space is visualized using j-dimensional graphical elements.

Figure 4.1 shows an overview of the categorization based on this systematization. The horizontal axis shows exemplary manifestations of 2D and 3D presentations of the spatial reference (i.e., terrain), while the vertical axis shows exemplary manifestations of 2D and 3D presentations of the attribute space (i.e., data values).

Based on this proposed systematization, existing visualization techniques can be categorized as either  $\mathcal{A}^2 \oplus \mathcal{R}^2$ ,  $\mathcal{A}^2 \oplus \mathcal{R}^3$ ,  $\mathcal{A}^3 \oplus \mathcal{R}^2$ , or  $\mathcal{A}^3 \oplus \mathcal{R}^3$ :

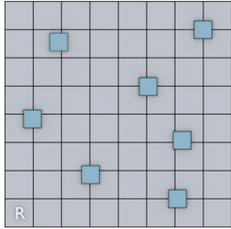
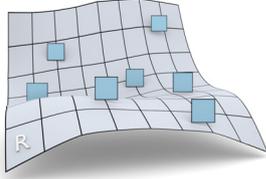
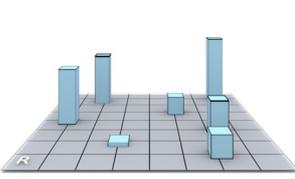
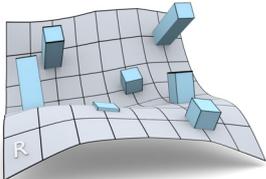
		Dimensionality of Reference Space	
		2D	3D
Dimensionality of Attribute Space	3D	 $\mathcal{A}^2 \oplus \mathcal{R}^2$	 $\mathcal{A}^2 \oplus \mathcal{R}^3$
	2D	 $\mathcal{A}^3 \oplus \mathcal{R}^2$	 $\mathcal{A}^3 \oplus \mathcal{R}^3$

Figure 4.1: Systematization of visualization techniques based on the dimensionality of the attribute space's and reference space's presentation ( $\mathcal{A}$  and  $\mathcal{R}$  respectively). For simplicity and clarity, the visual variables of the attribute representations are limited to a single color (blue) and a single item shape (square).

$\mathcal{A}^2 \oplus \mathcal{R}^2$  relates to presentations of data on 2D maps. As Figure 4.2a illustrates, the data values are presented by 2D graphical elements directly within the 2D presentation of the reference space without the need of further projections.

$\mathcal{A}^2 \oplus \mathcal{R}^3$  presentations shows the attribute space in 2D and the reference space in 3D, which allows not only to present the data values in a given 3D spatial context, but also enables the user to explore and understand the structure of a 3D terrain. The visual complexity of the data value's presentation though is limited to the use of 2D graphical elements only. An exemplary visualization is given in Figure 4.2b.

$\mathcal{A}^3 \oplus \mathcal{R}^2$  refers to visualizations, where the data values can also be presented in 3D, while the underlying spatial reference is shown in 2D. The third dimension allows for a greater flexibility in encoding the data attributes. It can be used to encode either the altitude of the data point or other attributes, such as time. For instance, Figure 4.2c shows an exemplary trajectory visualization above a planar map, where the different altitude of the trajectories encodes different points in time.

$\mathcal{A}^3 \oplus \mathcal{R}^3$  presentation use 3D for both the data values and the spatial reference. Presenting  $\mathcal{A}$  in 3D within a 3D depiction of  $\mathcal{R}$  allows for a natural perception of the attribute space's structure (e.g., distribution, extent, and correlation) as well as the terrain (e.g., shape). For example, Figure 4.2d shows an exemplary presentation of wind glyphs above the terrain.

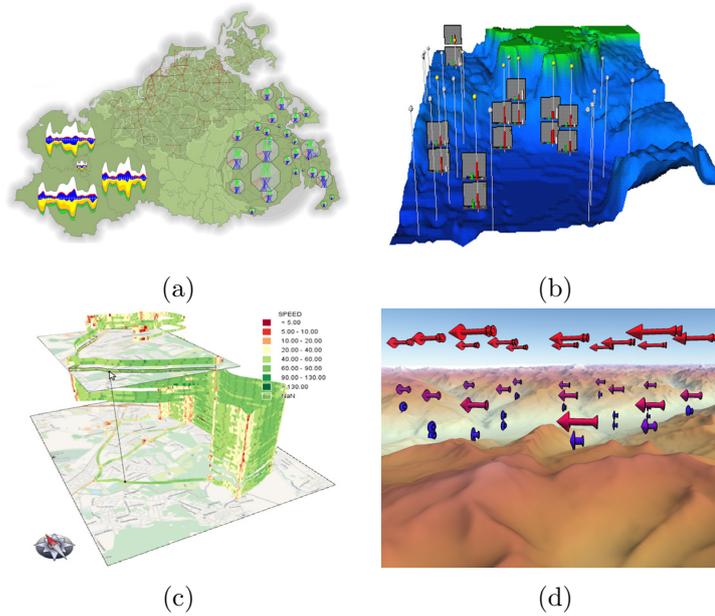


Figure 4.2: Exemplary visualization techniques for each category of the proposed systematization, showing (a) 2D diagrams on a 2D map [Fuchs and Schumann \(2004\)](#), (b) 2D diagrams on billboards and 3D ocean floor [Kreuseler \(2000\)](#), (c) 3D stacked trajectories over a 2D map [Tominski et al. \(2012\)](#), and (d) 3D wind glyphs in 3D terrain (TEDAVIS).

In general, the techniques of one category share common characteristics. Comparing these characteristics helps us to understand implications of using a 2D or 3D presentation of the attribute space and the reference space. The following section discusses each category in detail, examines respective visualization techniques, and points out challenges, problems, and possible solutions.

#### 4.1.2 Characteristics and Discussion

In our discussion we will focus on basic characteristics of the spatial data visualization techniques to emphasize the key factors of 2D and 3D attribute and reference presentations. This way, we aim to show how our systematization approach can deepen the understanding of advantages, disadvantages, and implications of the different visualization designs. To this end, we propose selected fundamental properties of the four combinations of 2D and 3D presentations, listed in [Table 4.1](#). For each property its general occurrence is marked by  $\bullet$ , while the absence is marked by  $\circ$ .

The shown criteria are reasonable for our means, though they represent only a selection and could be extended arbitrarily by further properties. While some properties in the table are obvious, others need further explanation. In the following, we will go into detail of the characteristics of each combination by means of exemplary visualization, generated by our visualization framework TEDAVIS (see [Section 6.1](#)).

#### $\mathcal{A}^2 \oplus \mathcal{R}^2$ Characteristics

The 2D presentation of attribute values in a 2D depiction of the reference space has a long history with many established systems and application areas. Among presentations of

Properties	$\mathcal{A}^2 \oplus \mathcal{R}^2$	$\mathcal{A}^2 \oplus \mathcal{R}^3$	$\mathcal{A}^3 \oplus \mathcal{R}^2$	$\mathcal{A}^3 \oplus \mathcal{R}^3$
No occlusion of $\mathcal{A}$ by $\mathcal{R}$	●	○	●	○
No self-occlusion of $\mathcal{A}$	●	○	○	○
No occlusion of $\mathcal{R}$ by $\mathcal{A}$	○	○	○	○
No self-occlusion of $\mathcal{R}$	●	○	●	○
Perspective distortion of $\mathcal{A}$	○	○	●	●
Perspective distortion of $\mathcal{R}$	○	●	●	●
Preservation of geometric properties of $\mathcal{A}$	●	○	○	○
Preservation of geometric properties of $\mathcal{R}$	○	●	○	●
Mapping preserves 2D spatial structure of elements in $\mathcal{A}$	●	●	●	●
Mapping preserves 3D spatial structure of elements in $\mathcal{A}$	○	○	●	●
Representability of 2D spatial distribution of elements in $\mathcal{A}$	●	●	●	●
Representability of 3D spatial distribution of elements in $\mathcal{A}$	○	○	●	●
Matching presentation of elements of $\mathcal{A}$ to $\mathcal{R}$	●	○	○	○
Using third dimension to encode attributes	○	○	●	●
Scalability of number of perceivable graphical elements	○	○	●	●

Table 4.1: Overview of the identified characteristics of 2D and 3D presentation of the attribute space ( $\mathcal{A}$ ) and 2D and 3D presentation of the reference space ( $\mathcal{R}$ ). The table shows the typical occurrence (●) or absence (○) of general properties for each category of our systematization.

this category are numerous well-known and widely-used visualizations, such as cartographic maps showing political borders or the topography. Such visualizations are solely constructed from 2D graphical elements. Hence, no projections or visibility computations are required to display them on a 2D image. However, in case of a geo-spatial reference space, distortions can appear because the surface of the earth is curved and uneven. Having said this, the distortion might only be noticeable on a larger scale. A further advantage of  $\mathcal{A}^2 \oplus \mathcal{R}^2$  is that appropriate designs and layouts for the graphical elements of the attribute space help to prevent occlusions. Consequently, data values can be easily read from uniform 2D displays, making such presentations particularly effective.

Figure 4.3a shows an illustrative visualization of multiple weather attributes. Specifically, temperature, wind direction, and wind speed are depicted on top of a geographic map. In this approach,  $\mathcal{R}$  is shown by hill shading.  $\mathcal{A}$  is densely encoded by several 2D graphical elements. For one thing, arrow glyphs encode the wind direction through the arrows tip. Wind speed is redundantly encoded through the length of the glyph and their color, ranging from blue (low speed) to red (high speed). Additionally, the temperature is encoded on the map's surface by a white (cold) to red (hot) color scale.

In this form of presentation, the multivariate weather attributes can be directly viewed in their spatial context and on their respective location. Moreover, the 2D presentation style facilitates a clear examination of the visual variables. However, the number of perceivable elements is limited, since too many graphical elements can easily result in visual clutter, which makes the identification of single objects as well as general patterns hardly possible.

Another approach is to combine multiple attributes or temporal changes to design more complex diagrams, iconographic displays, or movement patterns. For instance, Figure 4.3b shows approaching aircraft trajectories towards an airport. The line path represents the way the aircraft moves above the map and the color encodes their altitude (dark blue means low

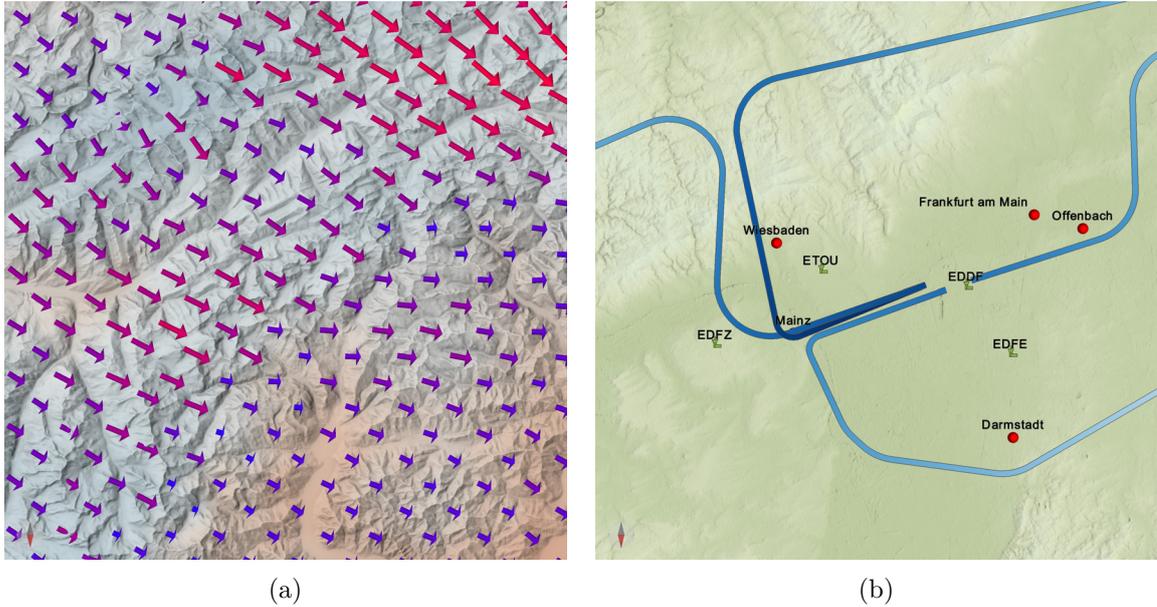


Figure 4.3: Exemplary  $\mathcal{A}^2 \oplus \mathcal{R}^2$  visualization techniques, showing (a) weather attributes (wind direction and strength through wind glyphs and temperature through color on the map) and (b) approaching aircraft trajectories towards the Frankfurt am Main Airport (EDDF) on a 2D map.

altitude and light blue means high altitude). Such a visualization facilitates an overview of different aircraft's position during the time of the landing procedure and so their movement patterns.

In sum, visualizations of the category  $\mathcal{A}^2 \oplus \mathcal{R}^2$  typically have a clear and easy to read layout. Occlusion of graphical elements usually poses no larger problem, though multiple data elements on the same position could still conceal information. The 2D allocation of the attributes within the spatial context is straightforward and no perspective distortion affects the representation. Due to these properties,  $\mathcal{A}^2 \oplus \mathcal{R}^2$  visualizations are often applied to communicate the 2D distribution of spatial data and to facilitate an overall view of the data. Next, we will investigate visualization designs using a 3D reference space.

### $\mathcal{A}^2 \oplus \mathcal{R}^3$ Characteristics

Visualization techniques of this category are constructed by combining 2D and 3D graphical elements. Consequently, projections and visibility computations are partially required for the display. The reference space is visualized using three display dimensions, which allows to represent a 3D spatial context in its full extent. For example, in visualization of geo-spatial data the third dimension is often used to depict virtual 3D models, and thus aiming for a less abstract presentation compared to 2D maps. Furthermore, such presentations of  $\mathcal{R}$  can support the interpretation of 3D spatial structure, such as the occurrence of attribute values in correlation with specific topographical characteristics, for example, mountains or valleys. The presentation of the attribute space is assembled from 2D graphical elements, most frequently billboards.

An example is given in Figure 4.4a. Here, the attribute space is represented using 2D billboards to indicate obstacles in form of poles for cable-ways along a valley. In general, the

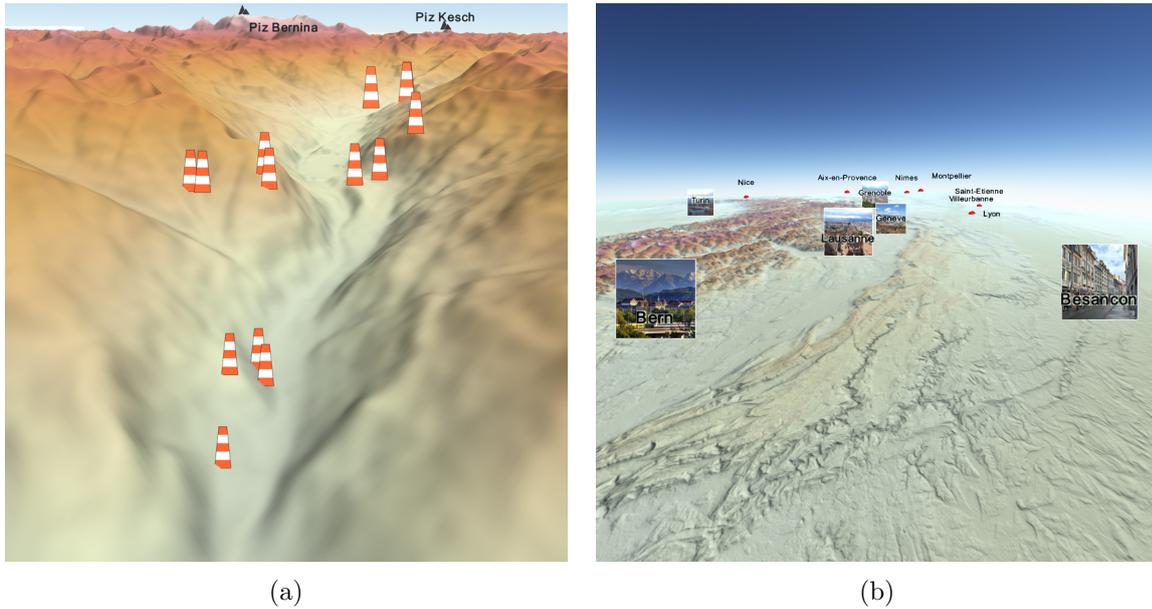


Figure 4.4: Exemplary  $\mathcal{A}^2 \oplus \mathcal{R}^3$  visualization techniques, illustrating the use of 2D billboard representing (a) obstacles in form of poles for cable-ways along a valley, and (b) city locations over a 3D representation of the terrain.

shape of billboards can range from icons to complex diagrams. They are typically placed at selected spatial locations and always face the viewer to counteract perspective distortions and orientation problems. However, the interpretation of such presentations might still be affected by perspective foreshortening, making the content of billboards far from the viewer more difficult to perceive. Figure 4.4b illustrates this problem. Moreover, this figure shows that in contrast to  $\mathcal{A}^2 \oplus \mathcal{R}^2$ , the spatial affiliations of the graphical elements to the reference space is often not as clearly identifiable. Especially if the attribute presentations are placed with a distance to corresponding location on the reference space, additional visual links, such as lines, are required to establish the associations.

A general challenge for visualization techniques of this category is occlusion, caused by the 3D depiction of the spatial reference. For example, near-surface perspectives usually involve a high ratio of occluded elements. Hence, suitable visualization techniques or other enhanced methods have to be incorporated into the presentation.

In sum,  $\mathcal{A}^2 \oplus \mathcal{R}^3$  presentations allow to communicate the 3D structure of the reference space very well, though this might also introduce occlusion and perspective distortion of geometric properties of the attribute values. Thus, such visualizations are particular suited to communicate the terrain topography and data without particular geometric properties, such as points of interest (0D).

### $\mathcal{A}^3 \oplus \mathcal{R}^2$ Characteristics

Visualizations of this category use three-dimensional graphical elements to depict the attribute values, whereas the reference space is represented by 2D elements. Here, the depiction of the data values can range from 3D bar charts and glyphs to trajectories and more complex objects. Generally, 3D graphical elements can be utilized to encode multiple attributes. For instance, the size and the shape of icons can be used to visualize the values

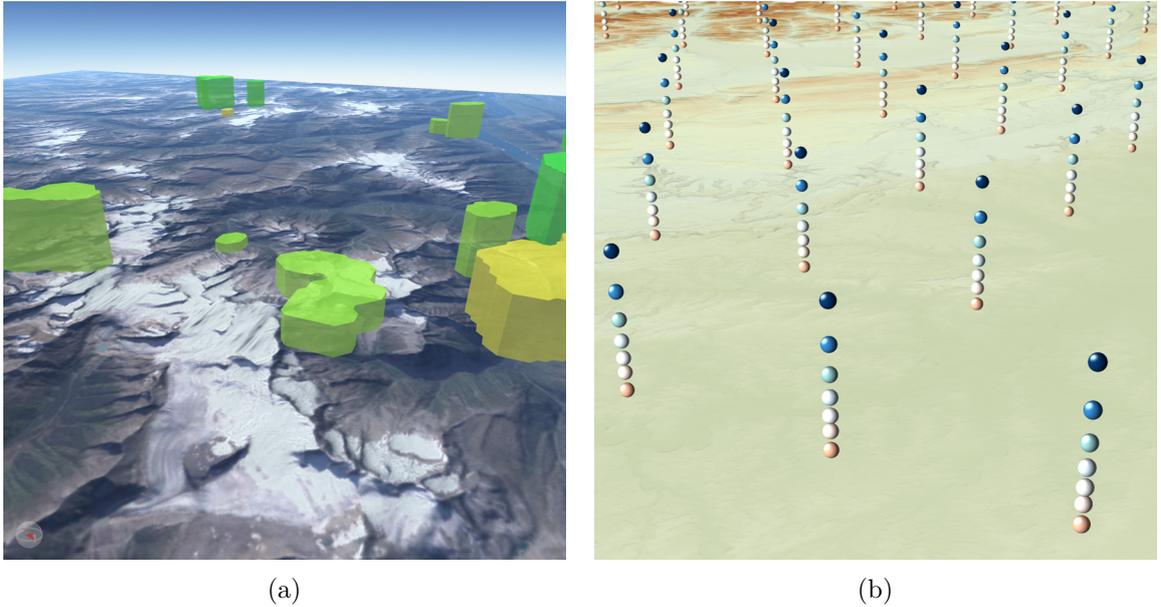


Figure 4.5: Exemplary  $\mathcal{A}^3 \oplus \mathcal{R}^2$  visualization techniques. (a) Abstract shapes, symbolizing extent and form of thunderstorm cells ( $x, y$  coordinates) and their severity (height) and movement speed (color). (b) Temperature measure points in different heights depicted by spheres of varying color above a map.

of two different attributes. Figure 4.5a illustrates such an approach. While the reference space is presented using an oblique view onto a 2D map textured with satellite images, the data values (here thunderstorm data) are depicted by 3D objects. The color of the objects (green to yellow) encodes the moving speed. Actually, the depicted thunderstorm cells have only a 2D extent, but the shapes are extruded along the  $z$ -axis to form 3D prisms. The  $z$ -axis is used to encode the severity of each particular cell. This is a common approach. Another frequent attribute that is encoded in height is time—which would lead to the classical space-time-cube (Hägerstraand (1970)). This strategy allows for a good overview of the location and 2D shape of the data, as well as the respectively encoded data attributes.

However, the perspective projection of the 3D elements might lead to distortions that in turn can influence the readability and measurability of the attribute values. To avoid such problems, an orthographic projection can be utilized. But as a result, the depth perception decreases as well as the amount of data that can be depicted. Moreover, accurately determining the location of data values within their spatial reference becomes a difficult task.

$\mathcal{A}^3 \oplus \mathcal{R}^2$  data visualizations are not as widely used as  $\mathcal{A}^2 \oplus \mathcal{R}^2$  presentations. However, the communication of the 3D distribution of the data can be helpful. Figure 4.5b shows measuring points above a map, which encodes the attribute temperature into the color of a 3D sphere. The  $z$ -coordinate this time represents the real height of the data points. Through this, the visualization communicates the decreasing temperature of the air with increasing altitude.

To conclude,  $\mathcal{A}^3 \oplus \mathcal{R}^2$  visualizations facilitate the conveyance of the 3D spatial distribution of the data, though identifying the exact 2D location on the map may be difficult if the distance between  $\mathcal{A}$  and  $\mathcal{R}$  becomes too large. Due to the perspective distortion, both the data and the map can partly forfeit readability, which has to be considered in the

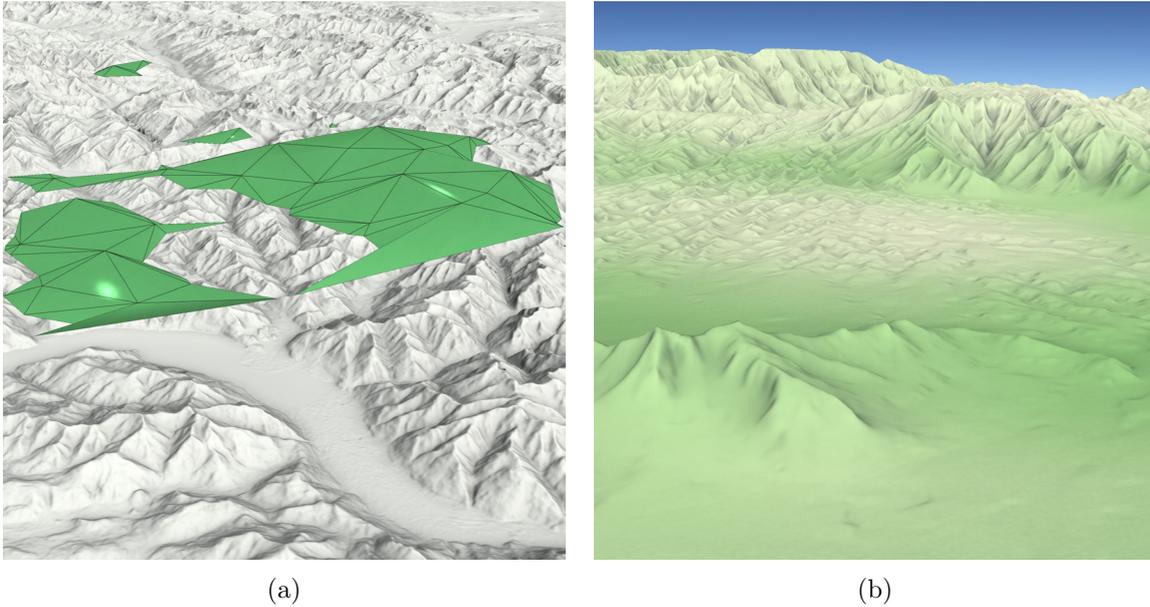


Figure 4.6: Exemplary  $\mathcal{A}^3 \oplus \mathcal{R}^3$  visualization techniques. (a) Isosurfaces (green polygons) representing the dew point of water in the air and thus giving indications, where clouds may formate. (b) Textured terrain model, using light green colors to encode low humidity and dark green colors to encode high humidity.

visual design. An advantage is the fact that the third dimension can be flexibly used to encode various attributes, including altitude and time. Therefore, such presentations can be beneficial when multivariate data must be communicated or if the 3D structure of the data is important.

### $\mathcal{A}^3 \oplus \mathcal{R}^3$ Characteristics

A 3D presentation of the attribute and reference space facilitates an intuitive perception of the 3D shape and extent of data values, as well as the structure of the underlying spatial context. Moreover, 3D spatial distributions are communicated effectively. However, as for every 3D presentation, a higher density of data leads to occlusion. But as an advantage, [Vion-Dury and Santana \(1994\)](#) pointed out that the scalability with regard to the number of perceivable elements is generally higher for 3D than for 2D representations.

Three-dimensional data visualizations are typically used for analyzing and forecasting the distribution of meteorological phenomena, such as clouds or airflows. For instance, we use 3D isosurfaces to visualize the 3D extent of locations where clouds may appear in [Figure 4.6a](#). The volumetric nature of this specific data though results in rather large occluders that can hinder the communication of parts of the data. A typical approach to engage this issue is to use transparency. However, with transparency the fore- and background cannot be distinguished well. Therefore, transparency should be used carefully to minimize ambiguity of color and structure.

A much more different approach of  $\mathcal{A}^3 \oplus \mathcal{R}^3$  is the application of textures to communicate the data, as shown in [Figure 4.6b](#). Though the categorization of texturing into  $\mathcal{A}^3 \oplus \mathcal{R}^3$  could be argued with regard to the 2D nature of the applied image data, we are still considering them as 3D graphical elements that follow the 3D shape of the terrain and therefore share

all properties of real three-dimensional primitives. For the same reason, we would also categorize textured maps as  $\mathcal{A}^2 \oplus \mathcal{R}^2$  presentations.

Texturing—an intrinsic data visualization—allows a consistent display of attribute values at every point in the presentation of the reference space. However, it is crucial that the raster data has a sufficient resolution and that appropriate filtering methods are used to prevent texturing artifacts, such as stretching or aliasing. Such artifacts may result in undefined visual representations and can lead to misreadings of attribute values, making advanced approaches, including level-of-abstraction methods or tailored texture projection techniques necessary. Similarly, the lighting and shading of the spatial context can influence the expressiveness of presentations with color-coded attributes. The introduced variations in brightness can impair the perception of colors and thus the identification of encoded values. Still, lighting is often necessary to communicate the spatial structures of the reference space.

In summary, presentations of the category  $\mathcal{A}^3 \oplus \mathcal{R}^3$  communicate the 3D characteristics of  $\mathcal{R}$  and the 3D spatial distribution of  $\mathcal{A}$  particular well. Perspective distortion and occlusion pose a certain difficulty, but scalability of cognitive processable elements might be an advantage of 3D presentations. As a result, such visualizations are especially suited for cases where spatial awareness matters or for preferably realistic representations of the terrain and the data.

### 4.1.3 Conclusion

In conclusion, we discussed typical visualization techniques for each category of the proposed systematization. Considering the selected properties in Table 4.1, the four categories can be characterized by the occurrence of distortion and occlusion, which arise naturally when using 3D either for  $\mathcal{A}$  or  $\mathcal{R}$ . This also implies that visual variables representing the attribute values can be distorted. Moreover, matching the spatial location of elements of the attribute space to their reference can be difficult when 3D is used. On the other hand, the comprehensive presentation of 3D spatial structures and distributions of elements of  $\mathcal{A}$  and  $\mathcal{R}$  is an advantage of 3D presentations. Furthermore, the number of perceivable graphical elements can be increased when the attribute space is presented in 3D.

As our discussion suggests, there is no universal answer to the question whether to use 2D or 3D. Rather, both strategies are valid and pose advantages and disadvantages. This holds particular true if considering the attribute space and the reference space individually, as it enables problem understanding for designing visualizations that combine them.

This serves our objective regarding scalable representations, that is, to provide different visualization options for different requirements. To this end, we discussed various choices for 2D and 3D presentations. By means of our systematization, we facilitate a better comprehension of characteristics and implications regarding 2D and 3D visualization of spatial data and hence enable choosing appropriate visualization strategies for the given application context.

With respect to our application scenarios, that is, spatial visualizations in cockpit environments, we can also deduce that spatial awareness is essential, meaning that the preservation of geometric properties and the representability of the spatial structure of the terrain have a high priority. In such cases, 3D presentations of the terrain are particularly beneficial. On the other hand, the absence of occlusion and perspective distortion, facilitating a decreased cognitive workload, is a strong argument for 2D presentations in cases where the spatial features are less relevant.

Simultaneously, the most critical stages of an aircraft's flight are typically the take-off and landing scenario of aircraft (Ochi and Kanai (1999)). During these scenarios, the visualizations of the surroundings in the SVS becomes particularly important and as a result 3D presentations of the terrain become essential. However, in practice 3D visualizations are still rare. In the following, we will focus particularly on 3D terrain presentations. With regard to the data, the choice whether to use 2D or 3D may depend on the importance of certain presentation's properties, such as the representability of 3D spatial distribution of the data elements or the perspective distortion of the elements. For either case, embedding the data representation into the terrain representation remains challenging. We will explore this further in the following section.

Nevertheless, the dimensionality is only one—though important—aspect of a presentation that must be considered for the visual design. In the next section, we will investigate in detail particular visualization methods for various data aspects.

## 4.2 Design Options for Terrain and Data Presentations

Representing terrain topography and spatial data aims to communicate their relevant characteristics, and thus tailored visualization techniques are required. With respect to scalable visualizations we need diverse presentation options that convey the important information for the given data. In this section, we will examine different options for 3D presentation of the terrain and for 2D and 3D presentation of the data. Hereby, we will discuss different levels of visual abstraction and their properties. Afterwards, we will extend our considerations onto the presentation quality. In this context, we will first discuss options to communicate uncertainty within the data. Secondly, we will go into detail of visualization techniques that improve the quality of data presentation, in particular the spatial awareness.

### 4.2.1 Terrain Visualization

Terrain visualization is a wide field of research with many different techniques. In Section 2.1 we already systematized three fundamental presentation approaches, that is, sketching, shading, and texturing. In this section we will take a closer look on these techniques in order to work out their respective characteristics regarding the communication of the terrain topography, which allows us to scale the presentation as needed.

We will first investigate the conveyance of the larger shapes that characterize the surface. In case that these characteristics are of particular interest, presentations can abstract from the detailed structures and focus on the fundamental shape of the terrain. Especially suited for such a presentations are sketching techniques. The most elementary form is the drawing of silhouettes. Figure 4.7a illustrates how the depiction of such a silhouette can already communicate characteristics of the terrain at a very coarse level of abstraction. In order to find and draw the silhouette, edge detection filters are used that base on depth information. The filter detects points in the image, where depth values change above a certain threshold between two adjacent pixel.

To generate a more detailed presentation of the terrain, more characteristic lines are needed. To this end, additional types of information of the terrain model are used, such as the curvature. Figure 4.7b shows a technique that looks for variations in the surface normal and thus adds edges where variations are large. Even more details can be made identifiable by adding so-called suggestive contours (cf. Section 2.1.2.3), which is illustrated by Figure 4.7c.

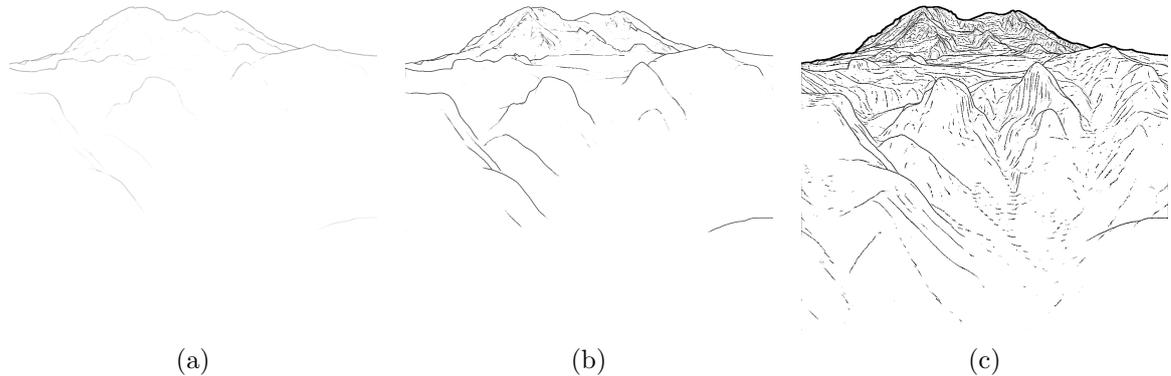


Figure 4.7: Enhanced edges using line drawing: (a) silhouettes are emphasized, (b) additional contours along ridges are drawn and (c) additional suggestive contours are drawn.

In principle, many different surface properties can be utilized to be enhanced with lines, such as the depth, the curvature, but also illumination or certain geographical characteristics. However, the number of lines that can be used is limited by the available visual budget. Hence, sketching techniques should focus on the communication of coarse geomorphological features.

In order to convey finer details of the terrain topography, a lower level of abstraction is necessary. Showing the 3D model by means of characteristic lines already provides a good overview on the terrain, though small variation in the elevation of the surface easily go unnoticed. Since these small protrusions neither occlude nor change with different viewing angles, they are almost only perceived by unconsciously interpreting shades on the surface. Thus we need appropriate shading techniques. Thereby, the quality of shading determines how well these structures can be recognized.

Most commonly, terrain surfaces are shaded using local illumination, such as the phong illumination model. Such local models can be computed quite fast because they depend on the orientation of the surface towards the light source only. In many cases this is sufficient to convey the terrain curvature, as shown in [Figure 4.8a](#). However, because brightness values are uniformly distributed along all potential surface orientations, small variations are hardly perceivable.

To improve the depiction of such fine structures, an approximative global illumination model, such as ambient occlusion, can be applied on top of a local illumination. [Figure 4.8b](#) shows that ambient occlusion makes fine grooves and scarps better recognizable. Ambient occlusion can be computed in a pre-processing step, making it almost as fast as local illumination. However, the memory footprint is increased.

Higher qualities of shading though can only be achieved through more sophisticated global illumination models. Global illumination enables casting shadows from elevations onto the surface, creating a more realistic representation of the terrain. Moreover, global illumination achieves higher contrast for local elements, and in this way making them more prominent. As a result, finer structures of the terrain can be emphasized using such sophisticated, high quality shading methods. On the downside, computing global illumination is considerably more expensive and memory consumption increases. [Figure 4.8c](#) shows the application of global illumination on the terrain surface.

The third option to convey particular characteristics of the terrain is the usage of tex-

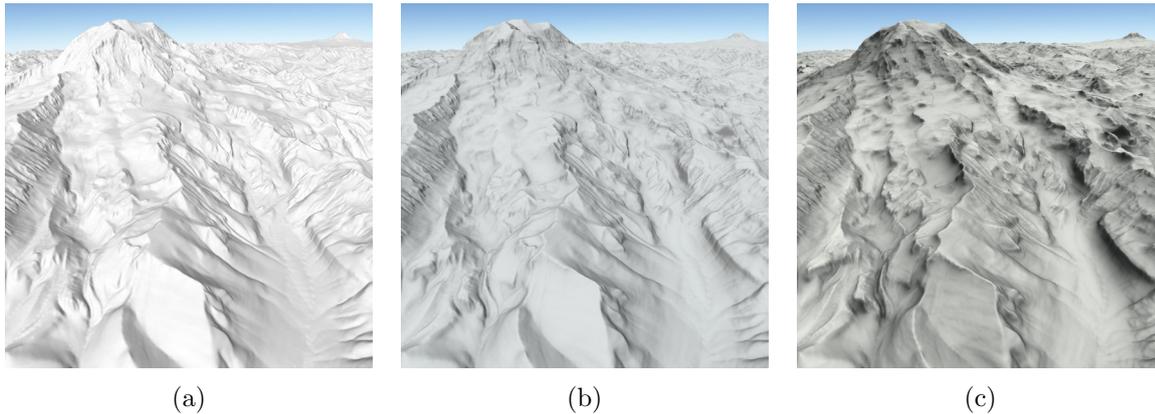


Figure 4.8: Conveying fine 3D structures of the terrain surface using different illumination models: (a) local illumination, (b) local illumination combined with ambient occlusion and (c) global illumination.

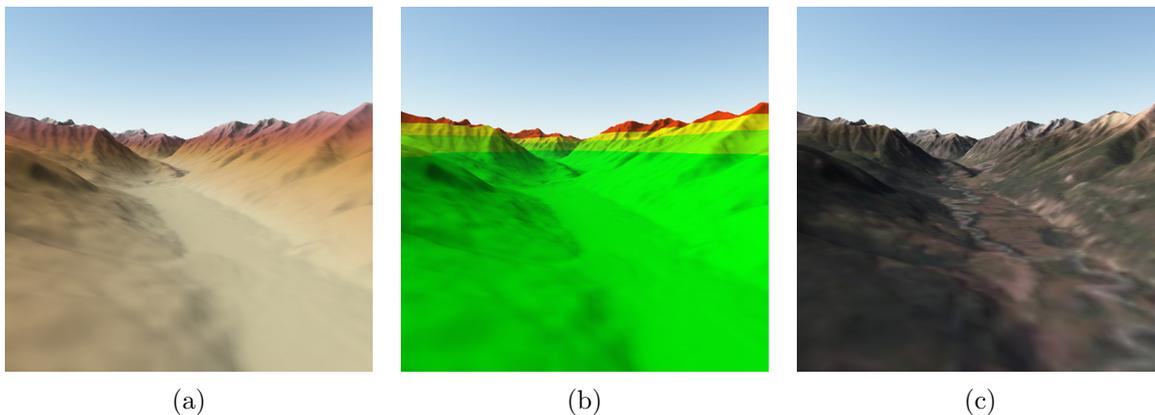


Figure 4.9: Applying textures to the terrain: (a) using colors relative to the viewer's height: green, yellow, and red (b) using hypsometric tints, and (c) using an aerial photo.

tures. Textures provide color information for each surface point and thus are able to communicate additional topographic features. A typical example is the use of hypsometric tints to show the terrain height. [Figure 4.9a](#) illustrates this type of texture, using a common color sequence ([Patterson and Jenny \(2011\)](#)).

Coloring the terrain based on its height can also be beneficial in situations where the relative height of the viewer to the terrain is crucial for safety. For example, in flight scenarios coloring can warn against dangerous elevations. In [Figure 4.9b](#) the terrain is colored relative to the viewer's height. Areas far below the viewer are colored green, while areas that are near to the viewer's height pose a certain danger and are colored yellow. Areas above the viewer are considered hazardous and thus are shown in red.

[Figure 4.9c](#) shows another example for texturing, the usage of aerial images. This technique can improve the realism of the presentation and conveys characteristic surface compositions, such as vegetation, water, or urban areas.

In general, textures alone cannot convey the spatial structure. [Figure 4.10a](#) illustrates this problem. Thus, usually terrain surfaces are not represented using texturing alone. Instead, textures are usually applied in combination, for instance with sketching ([Figure 4.10b](#))

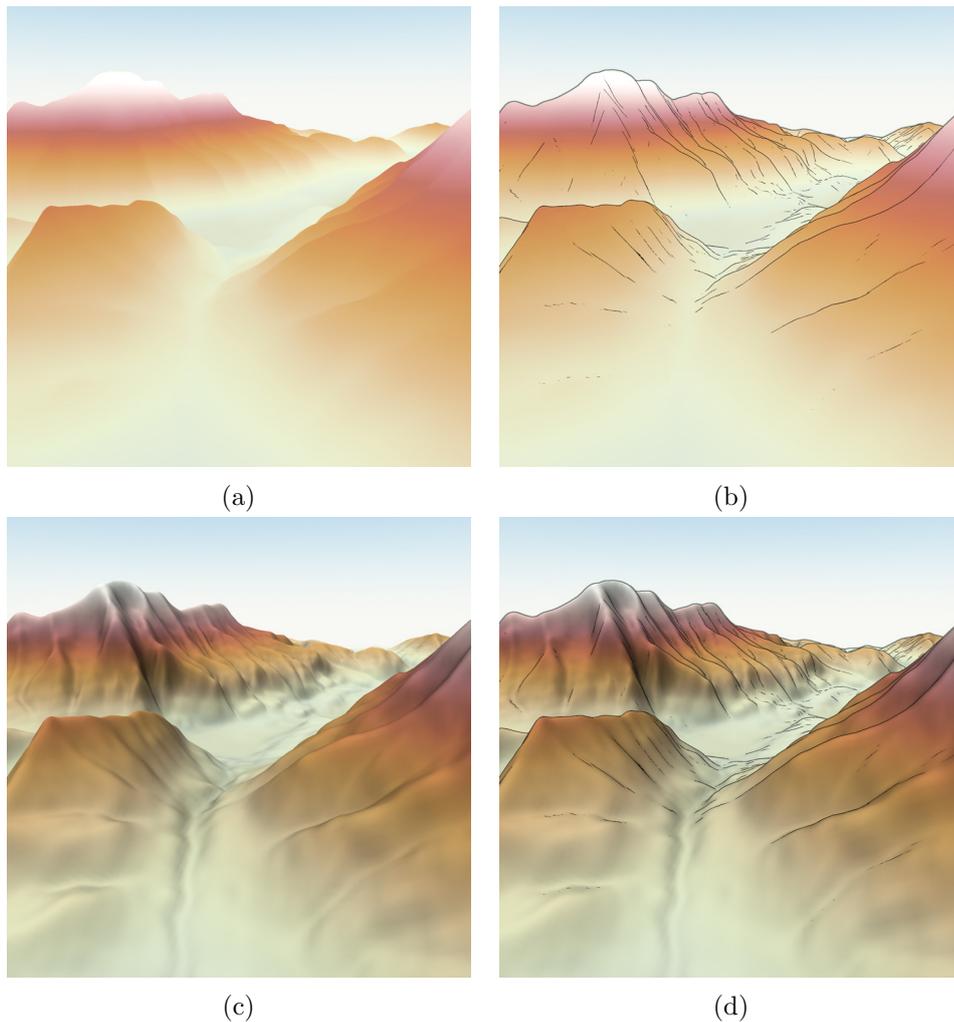


Figure 4.10: Terrain representations using texturing: (a) without additional techniques, (b) in combination with sketching, (c) in combination with shading, and (d) in combination with shading and sketching.

that communicates the contours of the surface or with shading (Figure 4.10c) that shows the detailed shape of the surface. Finally, all three visualization strategies can be combined, as shown in Figure 4.10d.

In a nutshell, the terrain visualization strategies sketching, shading, and texturing provide us with different options to communicate the terrain characteristics. Briefly outlined, we can state:

- Sketching is in particular suited to convey the fundamental shape of the topography and principally comprises uncomplex and fast techniques.
- Shading, in turn, facilitates the communication of finer features of the terrain, though involves a larger effort on computation.
- Finally, textures can further improve the visualization by communicating additional details of the terrain.

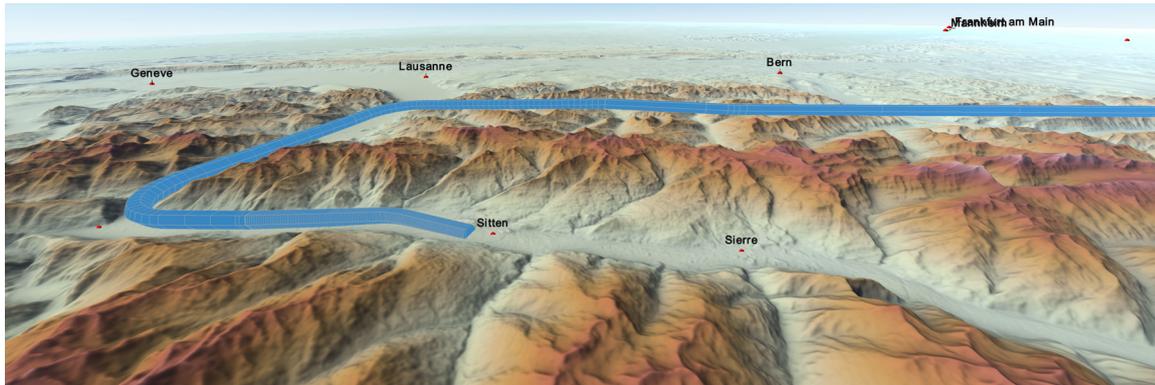


Figure 4.11: Visualization of an approach trajectory to the airport of Sion (Sitten), Switzerland. The trajectory is depicted as a hexagonal tube.

With respect to our approach these options allow scalability on presentation side and processing side (cf. [Section 3.3](#)). Regarding the presentation we can facilitate different levels of abstraction. Sketching poses a rather high abstraction level, while shading allows for a more natural presentation of the terrain and thus provides a lower level of abstraction. Texturing, in turn, facilitates a better adjustment of the presentation towards a given use case, especially in combination with sketching or shading.

On processing side, we can refer to the triangle of tension (cf. [Section 3.3.2](#)) to illustrate scalability by delineating our three strategies. In general, sketching has a low footprint of resource consumption and render time. Depending on the applied technique and parametrization, however, the amount of emphasized edges can be steered to either showing less or more features, though this has also an impact on render time. Shading typically requires a larger effort on processing side, though different techniques can also vary significantly in render time, resource consumption, and image quality. Exploiting this fact, we can use differently sophisticated shading methods to strike a balance for a given application scenario. Texturing, on the other hand, is quite fast, though image quality depends on the resolution of the texture data and thus depends on the available resources, such as memory.

## 4.2.2 Data Visualization

The visualization of spatial data is a complex field of research and various techniques exist for different types of data. Data used in the field of aviation (cf. [Section 3.1](#)) can be divided into geometrical data and numerical data. In the following, we will examine feasible techniques to depict data within a terrain context and provide different options to facilitate our goal of scalability. Hereby, we will concentrate on geometrical data by the example of trajectory depiction, before touching upon numerical data by the example of weather conditions. The large number of visualization options will be again substantiated with exemplary images, generated with our visualization tool.

### 4.2.2.1 Geometrical Data

As terrain visualization commonly shows the earth's surface, in many instances prominent geometrical objects are embedded. These can be real world objects, including buildings,

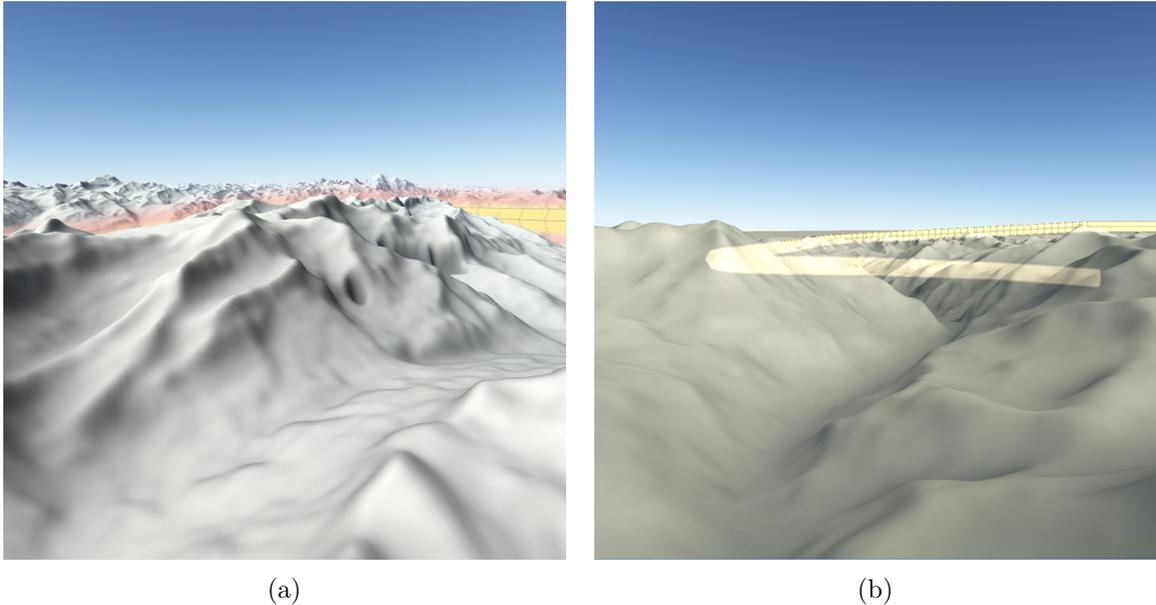


Figure 4.12: Improving communication of the position and path of the trajectory. (a) depicts a halo around the trajectory to increase its prominence; in (b), the geometry of the trajectory shines through the terrain if it is occluded.

trees, and roads, but also synthetic objects, such as representations of radar measured obstacles or trajectories. In general, these objects are positioned with the terrain using their geo-coordinates, for example, latitude and longitude. Geometrical objects are primarily defined by their shape, though they can also exhibit additional attributes, such as height, density, or speed. In the following, we will go into detail of geometric data visualization and describe different options. To this end, we will take the presentation of trajectories as our representative example.

Trajectories are line-based objects that typically lack information of the size or the width. To still be able to represent them prominently within terrain a certain extent must be presumed. Figure 4.11 illustrates this by the example of a trajectory depiction above terrain. The original trajectory data set consists of temporally ordered positions only. This, however, would define just a 1D path. To create a suitable visualization, we define a 3D bounding tube (Breuel (2013)). Concerning the size of this tube, it is important to balance out the perceptibility and the occlusion induced by the trajectory.

Objective of the visualization of geometric data in general and in particular of trajectories is conveying their location and their characteristic shape. Both are often difficult to observe. Due to their small size, trajectories are easily occluded by elevations of the terrain surface. One option would be to enlarge the trajectory. But, when viewed from close up, the trajectory would in turn occlude much of the terrain. Thus, we apply semi-transparent halos. In contrast to an enlargement, the halo is less obtrusive and does not fully occlude the terrain. Moreover, the size of the halo can be adjusted, so that it increases when far away and decreases, when viewed from a close-up position. Figure 4.12a shows how the trajectory is visually highlighted and how the path is roughly recognizable, even when occluded by terrain.

However, occlusion through the terrain can still occur. In order to engage this issue, we propose the use of ghosted views, a common Non-photorealistic Rendering (NPR) ren-

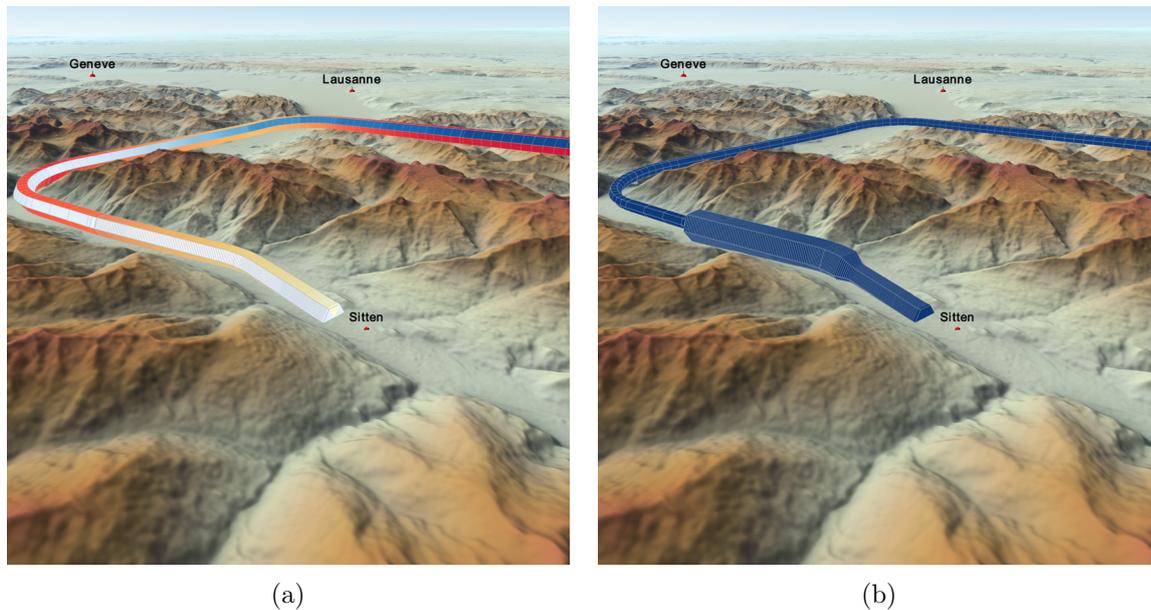


Figure 4.13: Communicating properties of the trajectory. (a) depicts two parameters encoded into color: wind speed (yellow to red) and the amount of cross wind (light blue to dark blue). (b) shows outliers in fuel consumption along the path by the width of the trajectory (greater width means larger fuel consumption).

der technique. By means of ghosted views, we can let the geometry of the trajectory shine through occluders and thus, permanently communicate their location and shape. Figure 4.12b illustrates this approach.

To this point, we depicted the location and shape of the trajectory only. If additional information, such as measured or calculated attributes, must be communicated as well, we need to adjust the representation. Common attributes for trajectories would be, for instance, height, speed, or surrounding wind conditions. To convey these characteristics, we utilize different visual variables, including color, texture, and—to a certain extent—the size of the trajectory.

Color, for instance, is particularly suited to encode quantitative attributes (Carpendale (2003)). In Figure 4.13a we apply a technique that allows us to color the trajectory by even two color scales. One color (light blue to dark blue) shows the amount of cross wind during airport approach and the other color (yellow to red) represents wind speed that affects the aircraft along the path. Both attributes together sum up to dangerous wind conditions along the path. By using appropriate forms for the trajectory’s representations, such as a hexagonal shape, we can ensure that both colors are visible for each viewing angle. On the downside, two color scales at the same time are generally more difficult to interpret than just one.

Figure 4.13b shows another example, where we encode the attribute using the visual variable size. While quantification of size in a 3D presentation is quite difficult, side-by-side comparison is still facilitated. Thus, we can use size to expose outliers, as they would result into an abrupt change of extent. In this example, the disproportional fuel consumption shortly before landing is communicated.

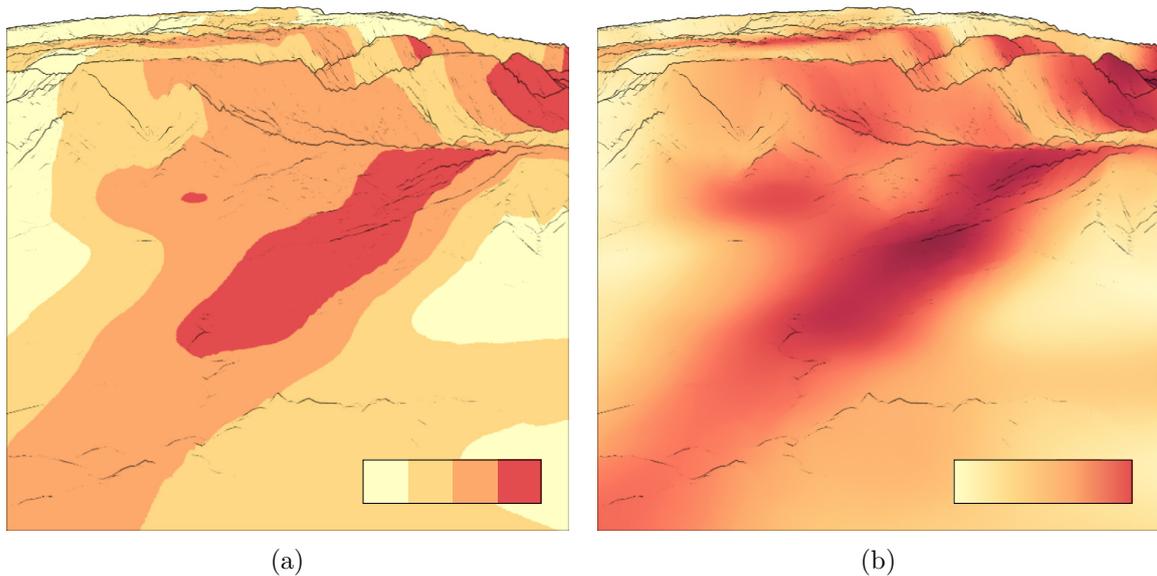


Figure 4.14: Strategies to show numerical data. (a) Representation via a segmented color scale shows basic qualitative characteristics of the data. (b) More detail can be discerned when a continuous color scale is used.

To briefly outline, the described visualization options can be used to facilitate different levels of abstractions:

- To convey exclusively the location and shape of the trajectory, we use a bounding tube for representation.
- To further emphasize these characteristics, halos or ghosted views are applied.
- Additional numerical attributes are communicated by utilizing visual variables, such as color or size.

#### 4.2.2.2 Numerical Data

Numerical data refers to measured or calculated data values that—even though spatially referenced—have no predefined geometrical form. As a consequence, numerical data values are mapped to appropriate graphical elements in the visualization process (cf. [Section 2.2.2](#)). There is a huge diversity of numerical data types and hence there are a lot of visualization methods. It would be beyond this work to delve into the very details of all of these methods, but we will—in accordance to our objective scalability—focus on examining the options to communicate different levels of visual abstractions. This corresponds to the questions, whether a more or less detailed visual representation is needed. To this effect, the principle approach is to show the data either by segmented or continuous visual encodings.

With regard to aggregation different methods can be applied, including clustering, classification, or segmentation. As a result, individual data values are composed into larger units or groups. These units capture basic characteristics of the data and provide a good overview, yet they lack details.

[Figure 4.14a](#) shows an example for data visualization using segmented color within terrain. The figure shows the intensity of wind gusts within the terrain. Four different

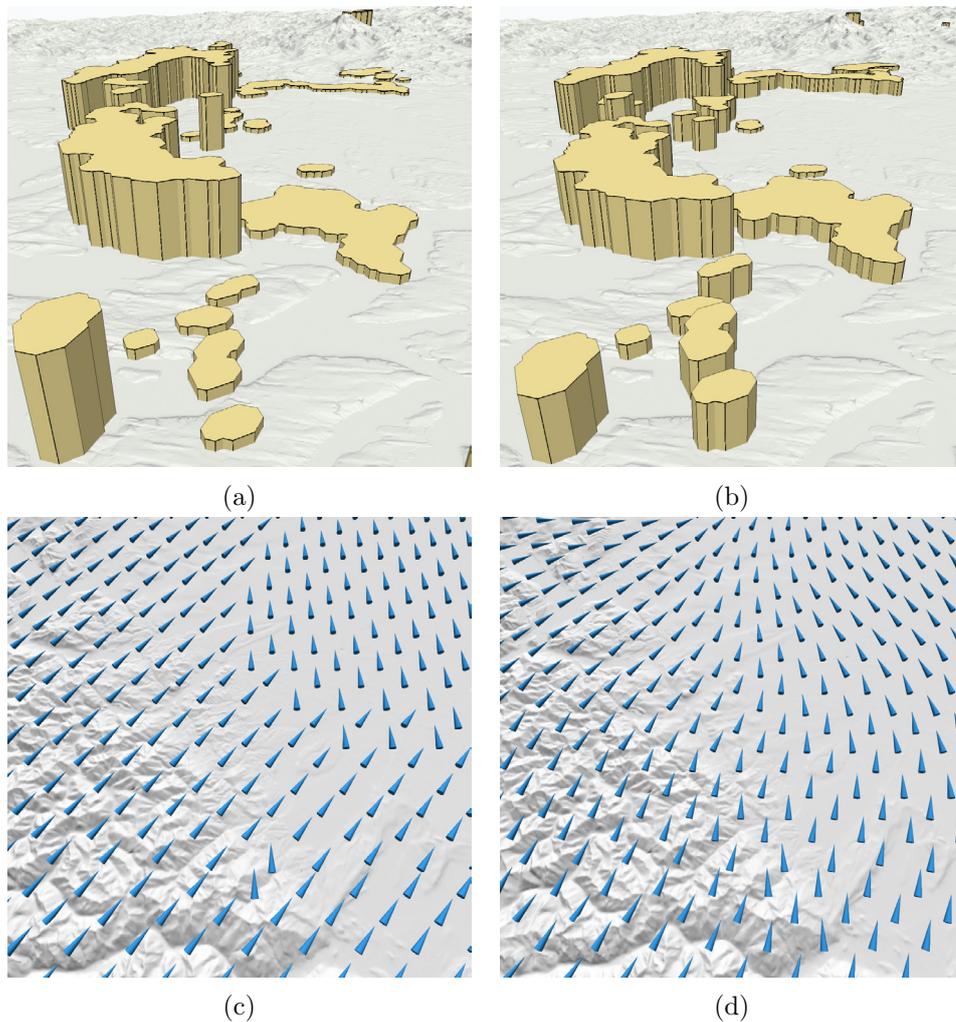


Figure 4.15: Numeric data representations using size (a,b) and direction (c,d) as visual encodings. The left side shows segmented encodings, while the right side shows continuous encodings.

colors classify regions into four different zones, ranging from light orange (low wind speed) to red (high wind speed). The four categories of intensity can be distinguished easily, whereas the underlying data values are abstracted away.

If there is a need to grasp more details, a trace back to the individual data values can be necessary. With regard to this example, individual data values can be communicated using a continuous color scale, where each data value is associated with an individual color. [Figure 4.14b](#) shows the individual values of measured wind speed, which is the quantitative measure behind the previous qualitative categorization of the intensity of wind gusts.

Beside color, data values can be mapped on different visual variables, such as size, shape, or orientation, using intrinsic and extrinsic visualization techniques. Accordingly, the strategy of segmentation can be applied to these encodings as well. For example, [Figure 4.15a](#) and [4.15b](#) shows this strategy for visualizing thunderstorm cells, using height to encode intensity. [Figure 4.15c](#) and [4.15d](#) shows wind arrows which encode wind direction into the orientation of the glyphs. The number of categories used for the segmentation

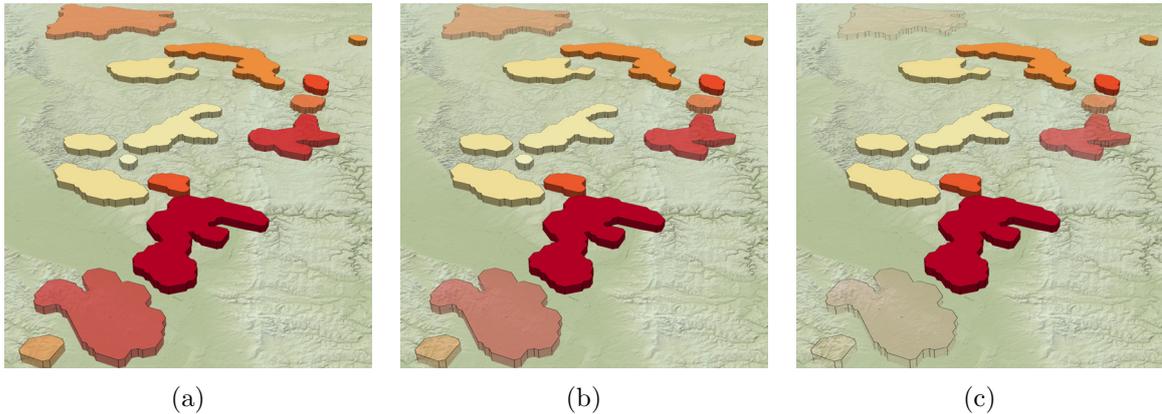


Figure 4.16: Intrinsic uncertainty visualizations using different levels of opacity: (a) 45% transparency, (b) 65% transparency, (c) 85% transparency.

eventually determines the degree of abstraction of the visual design. Using more categories provides more details, while using less categories facilitates a higher level of abstraction.

To sum up, the amount of distinct visualization techniques is huge.

- In general, we can distinguish intrinsic and extrinsic techniques that use different visual variables to encode the attributes.
- In order to achieve individually abstraction levels, most vis techniques can be adjusted to show either individual data values or different levels of abstraction.
- The number of representatives in the segmented encoding steers the actual degree of detail and thus the level of abstraction.

In principle, we encode a certain number of representatives, for example, through a segmented color scale, whereby an increasing amount of representatives determines an decreasing degree of abstraction to the point where all individual data values are shown. Using this strategy for geometrical and numerical data we can provide different visualization options to facilitate different levels of abstraction. This is consistent with the objective of scalability on presentation side.

On processing side, data visualization has commonly not such a major impact on render time and resource consumption as terrain rendering has. Nevertheless, the general rule applies that the fewer individual data values are depicted the less effort is required for the visualization. However, finding an adequate aggregation can be intricate by itself.

This concludes the visualization options for spatial data within terrain. In the following, we will address two further aspects in order to improve the quality of the presentation. For one thing, we want to increase confidence and acceptance of such data visualizations by incorporating information about the data quality into the presentation. For another thing, we want to increase the spatial awareness within 3D presentations and thus, facilitate the conveyance of spatial features.

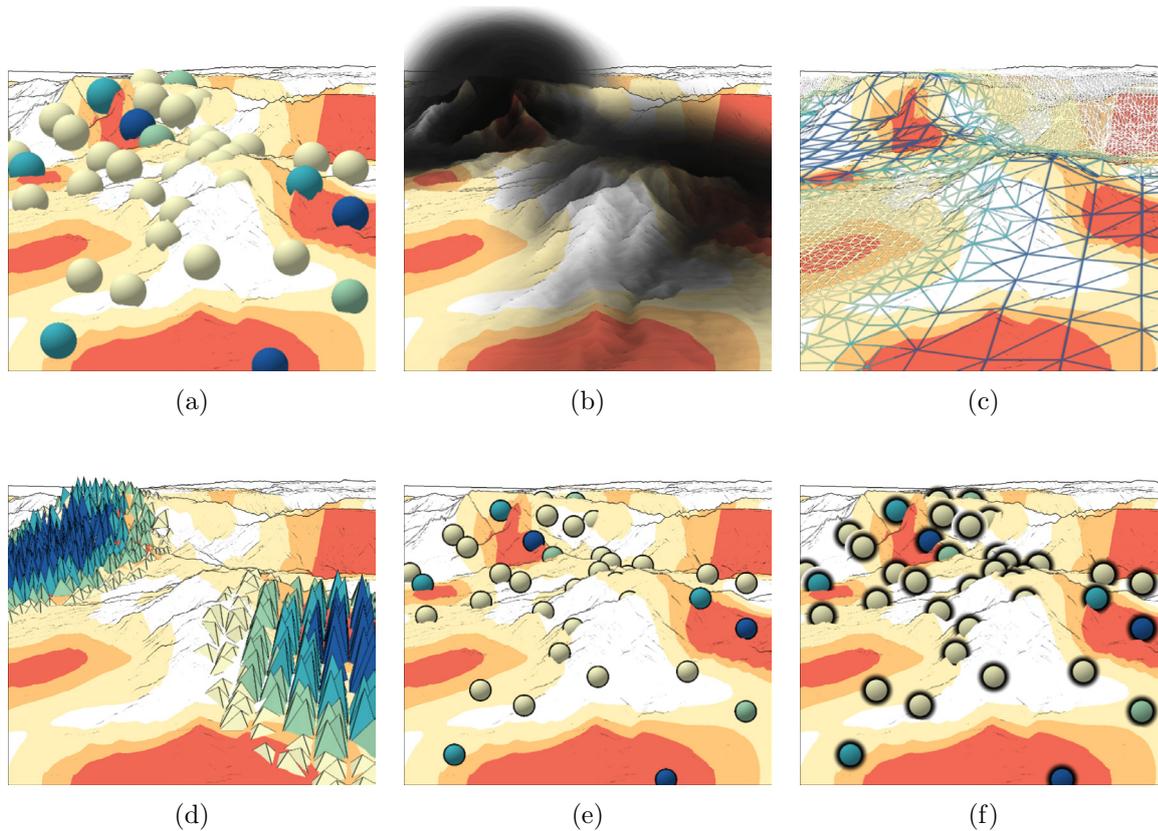


Figure 4.17: Options for visualizing uncertainty. (a) circular glyphs, (b) fog, (c) grid overlay, (d) tetrahedral glyphs with redundant encoding by height and color, (e) outline stylization, and (f) halo stylization.

### 4.2.3 Data Quality

When referring to information visualizing, most methods are concerned with valid data representations. Yet at the same time, data themselves are rarely free of uncertainty. Hence, addressing the issue of representing data validity is also an important subject (Goodchild et al. (1994)). Uncertainty visualization in general already has a long history, though showing uncertainty in spatial data embedded into a 3D presentation of the terrain has rarely been considered in literature. Thus, in order to communicate the data quality in our context, we need to develop novel approaches based on original 2D concepts. To this end, we will consider both intrinsic and the extrinsic visualization methods, which respectively provide different degrees of detail.

Since the idea of intrinsic techniques is to modify the existing visual elements of the data, commonly only a few number of uncertainty gradations can be encoded. Hence, intrinsic techniques typically represent a higher level of abstraction. Specifically, we propose visualization options that vary transparency, saturation, noise, strokes, or bump mapping to modify the data representation. A suitable balance of data and uncertainty visualization can be found by adjusting the strength of the modulation. For example, Figure 4.16 illustrate the modulation of transparency, where uncertain data are represented by high transparency. From Figure 4.16a to 4.16c, the modulation increases in strength. Using moderate transparency (b) allows to identify where uncertainty is located and which data

values are affected. In contrast, when increasing the strength of the modulation (c), the actual data values in uncertain areas become barely visible. This can be useful to draw the attention to data that are certain.

Extrinsic techniques, on the other hand, are less dependent on the data representation and thus provide us with a wide variety of visual encodings. Accordingly, we can communicate a higher degree of detail on the data quality. For instance, we are able to add 2D and 3D glyphs into the representation, including circles, triangles, or tetrahedrons. The former are illustrated in [Figure 4.17a](#).

Other design options for uncertainty presentations is the appliance of fog, which occludes uncertain data values ([Figure 4.17b](#)), and grid overlays ([Figure 4.17c](#)). For grid overlays, the resolution of the depicted triangular mesh varies with the underlying uncertainty. The mesh is subdivided coarsely for uncertain areas to indicate vagueness. A finely subdivided mesh indicates crispness in parts where the data are certain.

The visual appearance of extrinsic techniques is obviously subject to fine-tuning, for example, by adjusting size, density, and color, depending on the desired level of detail. The standard way is to map uncertainty to one selected property. However, we also use redundant encodings. For example, uncertainty can be mapped to both glyph size and color. When applied with care, this can increase the visual prominence of highly uncertain parts of the data. Exemplarily, [Figure 4.17d](#) shows tetrahedral glyphs with varying height and color. Additional stylization via outlines ([Figure 4.17e](#)) or halos ([Figure 4.17f](#)) can be applied further to better separate glyphs and terrain.

To conclude, uncertainty representation is essential for data visualizations. Using intrinsic and extrinsic techniques, we can provide two fundamental options for distinct abstraction levels.

- Intrinsic techniques can typically convey only few gradations of uncertainty.
- Extrinsic techniques allow for more detail due to a higher number of possible encodings.
- Moreover, extrinsic techniques are typically more prominent and hence are more likely to attract attention in the presentation.

However, the particular abstraction level is again subject to the number of uncertainty values that should be communicated. Therefore, we can fine-tune the desired degree of detail by encoding more or less gradations within the technique's scope of possibilities and thus achieve a certain presentation scalability. Regarding processing on the other hand, there is little difference between the various uncertainty visualization techniques, making only the decision whether to integrate uncertainty information into the presentation or not a relevant factor.

### 4.2.4 Spatial Awareness

When visualizing spatial data, we need to relate the data to the spatial context in order to communicate their location and distribution. The same holds true with the structure and topography of the 3D terrain. To this end, we need presentations that convey the position of the individual elements to one another and together with the 3D depth of the scene. This is called spatial awareness.

In [Section 3.1](#) we pointed out that spatial awareness is crucial for cockpit environments. However, the conveyance of spatial awareness is a common issue in 3D computer graphics. In our context, this problem becomes particularly evident in cases of the data elements and the terrain being watched at a flat viewing angle. [Figure 4.18a](#) shows such a situation for a visualization of trajectories above terrain. For one part, this issue could be engaged using stereoscopic images. However, this is not always applicable. Thus, we propose the deployment of specific, tailored depth cues to improve distance perception.

One reason for the difficult depth perception in terrain environments is the fact that the viewer does not know about the size of data elements. Therefore, the perspective distortion—the most common depth cue—does not provide many hints on distances. To engage this issue, we place new elements into the scene, to which size the viewer can relate. For instance, in [Figure 4.18b](#) we lay a regular grid onto the terrain surface. Since the perspective distortion of the grid cells is easily interpretable, the viewer gets a better sense of the scene’s depth. Distances between the trajectories can also be better estimated if cell size is known. As a side-effect, the application of grids can further improve the communication of the terrain curvature.

A similar strategy is illustrated in [Figure 4.19a](#). Instead of a grid on the terrain surface, we place vertical lines onto the trajectory representation equidistant spacing. To further improve this technique, these lines can be animated, so that they travel away from the viewer or along the trajectory and thus facilitate the communication of the shape. In order to ease the association between parts of the terrain and the corresponding parts of the trajectory regarding distance to the viewer, the lines can also be extended onto the terrain surface as shown in [Figure 4.19b](#). This approach has been developed in a bachelor’s thesis ([Judzinsky \(2016\)](#)), which the author of this dissertation supervised.

A further issue is the difficult perception of the spacing between data elements and terrain. Consequentially, the viewer can hardly comprehend above which part of the terrain the represented data are located. In order to deal with this problem, we use either shadow-like orthogonal projections that show the exact position on the terrain, or we add graphical primitives, including lines or polygons, that link the data elements to the surface. The application of projections is illustrated in [Figure 4.20a](#). Similarly, [Figure 4.20b](#) shows a semi-transparent surface that connect the trajectory with the terrain position. The visual links can also be combined, for example, with grids in order to enhance the conveyance of the exact position of the trajectory.

A disadvantage of using additional elements as visual cues is the danger of cluttering the image. Furthermore, when considering the perception of landscapes in real-life, we do not need additional elements as visual cues are still able to relate to distances in the environment. This is mainly due to certain other visual effects that are both subtle but effective.

One of them is the depth of field, which results from the fact that a lens (of the eye or a camera) can only focus on one distance at once. We exploit this natural effect and apply it to our visualizations by defining a focus area in which everything is rendered sharply, while on the outside the elements are blurred. [Figure 4.21\(a\)](#) shows our approach on a visualization of wind glyphs. Elements far away (or very near) to the viewer are blurred, which highlights only data glyphs at a specified distance interval. Besides the achieved depth sense, this technique also relieves the visual budget because elements far away are intuitively considered less relevant. To this end, the size and position of the focus area can be adapted either with regard to the camera position or to the data location.

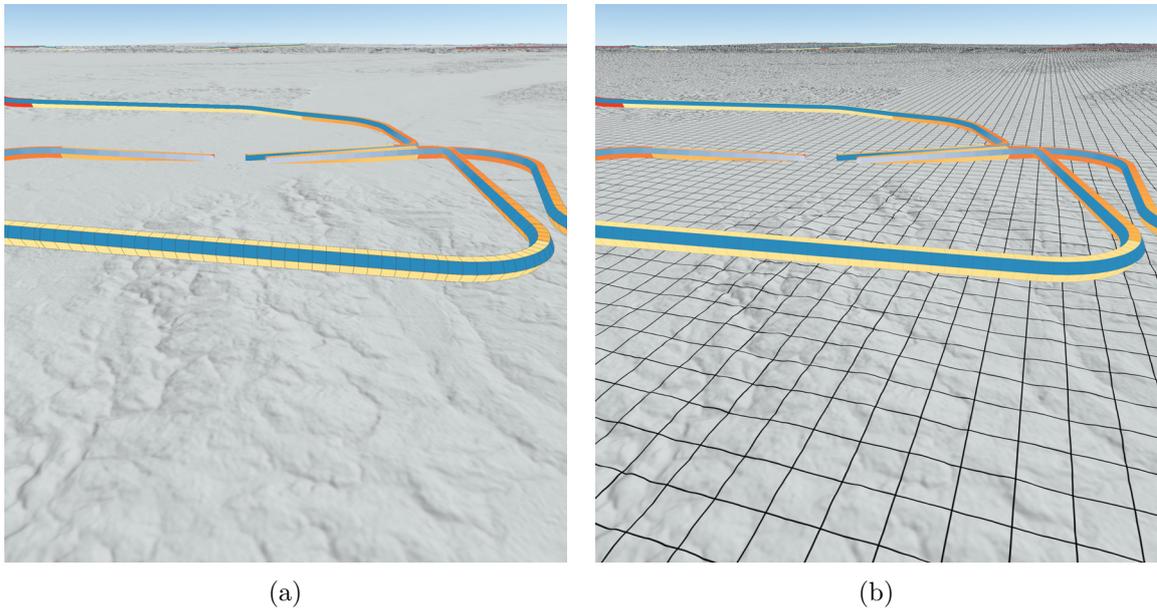


Figure 4.18: Spatial awareness regarding trajectory visualization: (a) without visual cues and (b) using grids on terrain surface

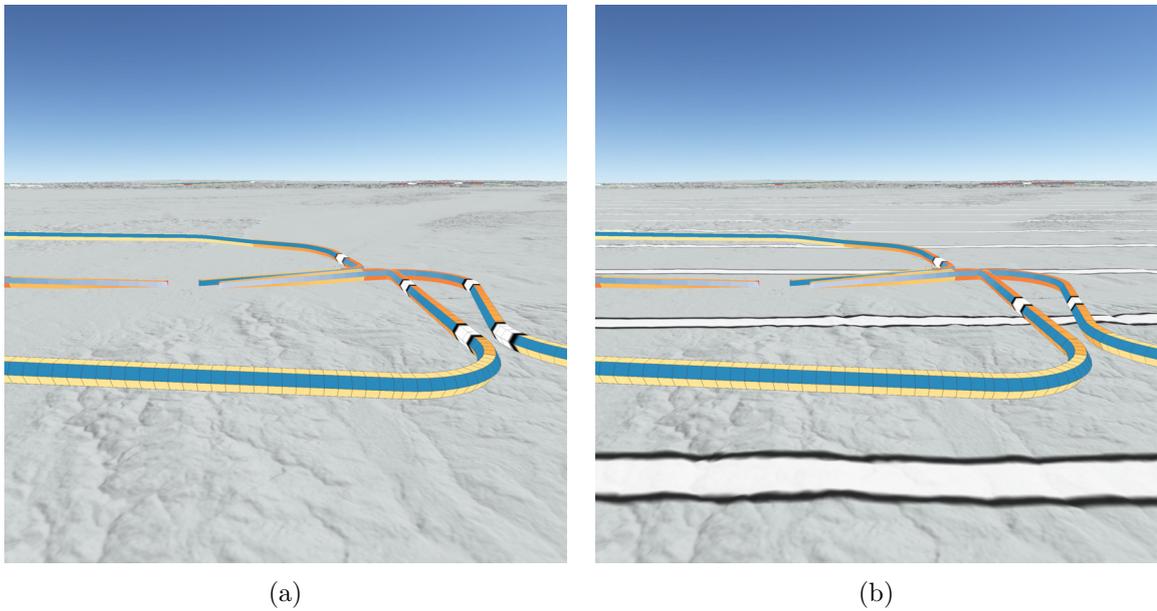


Figure 4.19: Improving spatial awareness regarding trajectory visualization using animated lines (a) on the trajectory only and (b) additionally on the terrain the surface.

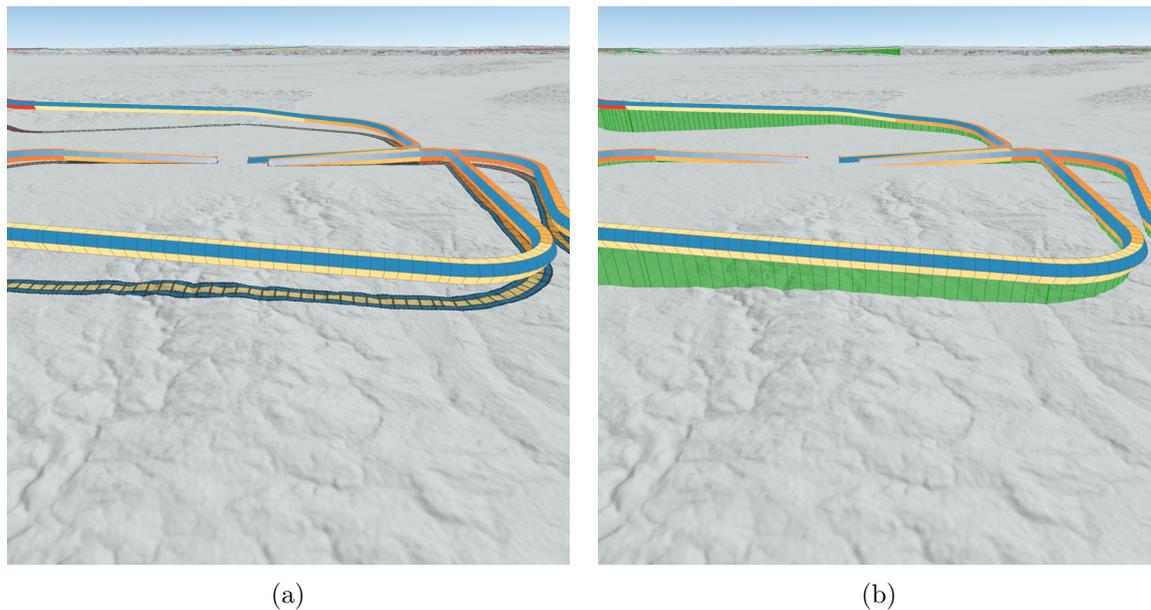


Figure 4.20: Improving spatial awareness regarding trajectory visualization using grids on terrain together with (a) orthogonal projections onto the surface and (b) connecting polygons between trajectories and the terrain surface.

Another capable and realistic visual effect is the application of atmosphere rendering. Since the atmosphere absorbs and scatters the light while it travels to the viewer, contrast and saturation decreases with increasing distance. A fast and simple rendering method to feign such an effect is fogging. Fogging fades the color of the scene to the color of the background with increasing distance, creating the impression of thick mist. However, such an effect seems often unnatural and obtrusive. Thus, we apply outdoor light scattering, also called aerial perspective (Hoffman et al. (2002)), that simulates the interrelation of light and atmosphere realistically, taking scattering on water and oxygen molecules into account. In contrast to fogging, this method is way more sophisticated regarding to the image quality—though on the downside also challenging with respect to computational cost. Figure 4.21(b) illustrates that aerial perspective blends nicely with data visualization in terrain, making the representation look more natural while giving the viewer a hint of object distances.

An alternative cue for improving spatial awareness is the application of soft shadows, which is well suited to indicate 3D positions. However, if elements are wide apart, they cannot cast shadows onto each other. For such situations, we propose a method that is still unconventional for data visualization: volumetric lighting. Using this subtle effect, we can let objects cast shadows into the air, which is illustrated in Figure 4.22. This technique originates from Photorealistic Rendering (PR) and is typically used to achieve a sense of atmospheric density. By applying it to data visualization depth perception can be improved.

Both, atmosphere rendering and volumetric lighting depends on various parameters, such as composition and density of the atmosphere as well as properties of the light source. These parameters control the unobtrusiveness of these effect and must be adjusted with care. Otherwise, too intense effects could confuse or distract the user.

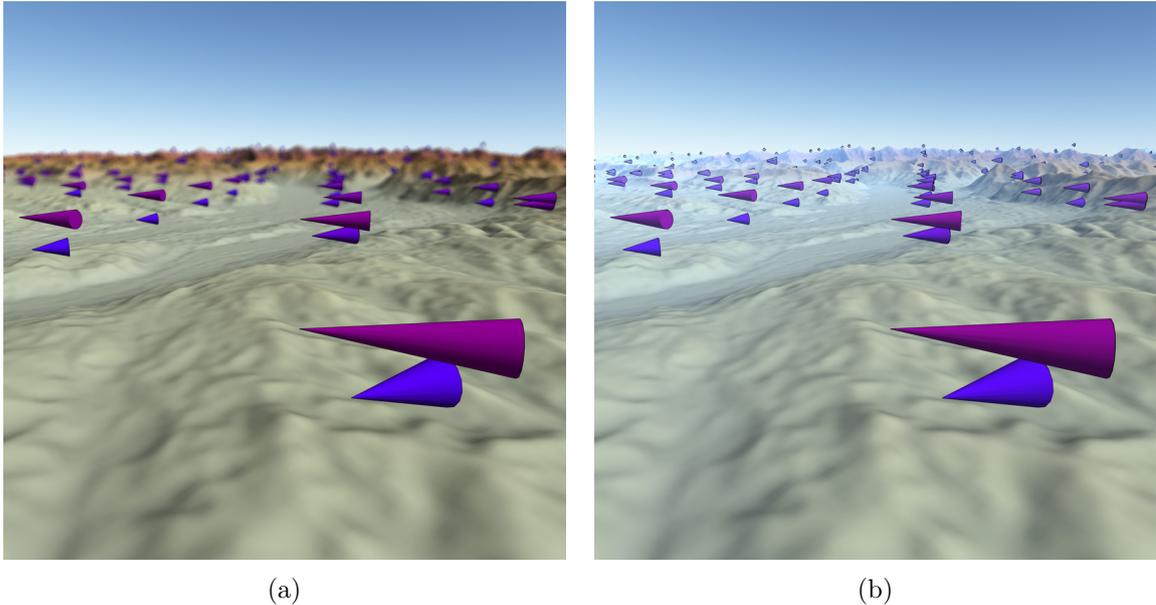


Figure 4.21: Enhancing the perception of the spatial distribution of objects within terrain. The images show wind glyphs that encode wind direction (arrow head) and strength using length and a continuous color scale, ranging from blue (light wind) to red (high wind). In (a), depth of field is applied to improve depth perception. (b) shows the application of aerial perspective.

To briefly recap the visualization options that augment spatial awareness, we can distinguish between adding graphical elements and using visual effects.

- Graphical elements, including grids and links, improve the communication of the data elements' location. This holds particularly true in cases where there is a certain distance between the terrain and the data representations, though these methods are prone to visual clutter if too many of them are applied.
- Visual effects are more subtle, often changing the overall impression of the visualization and thus must be applied with care to ensure the readability of the data.

Since the application of visual cues can improve the quality of the visualization, a certain scalability on presentation side can be facilitated. However, a concrete quantification is difficult if not impossible to assess. Regarding scalability for processing, it can be stated that generally visual effects are more sophisticated than rendering additional elements. Furthermore, the effort on processing side commonly determines the quality of the effect. In this sense, aerial perspective requires more render time and resources than fogging, though the image quality is also unequalled.

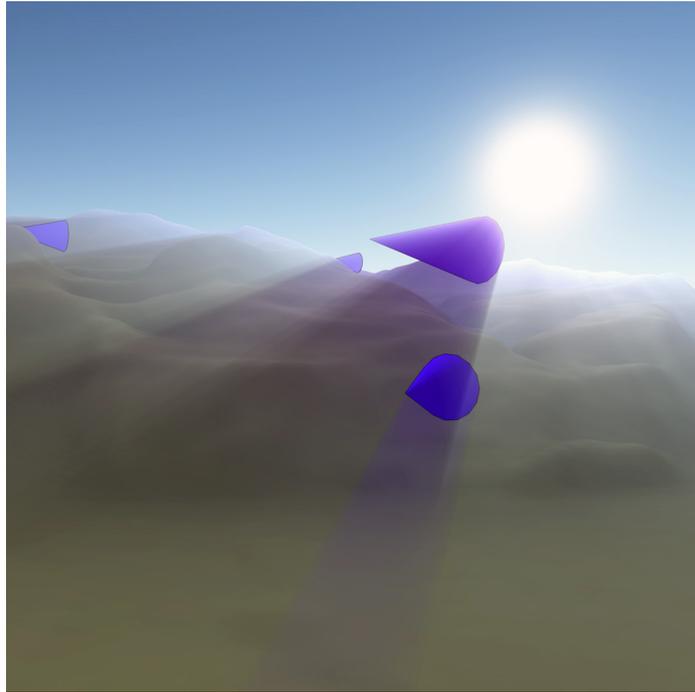


Figure 4.22: Application of volumetric light to improve spatial awareness. Here, the shadowing effect is stressed for illustrative purpose.

### 4.3 Summary

In this chapter we characterized the various visualization strategies for spatial data in terrain. To this end, we first examined the question, whether to use 2D or 3D presentations and proposed a novel systematization that advances the discussion by distinguishing between 2D and 3D presentations of both, the terrain and the data. Through this, we outlined advantages, drawbacks, and implications of visualization methods of either category. This facilitates a better choice of the presentation’s dimensionality with respect to different application scenarios and scaling between different categories with changing requirements. Based on our observation, we reasoned the further investigation of terrain representation in 3D and data representations in both 2D and 3D. However, 2D data visualizations in 3D terrain are typically restricted to billboard presentations and thus primarily suitable for selected numerical data types only.

Afterwards, we proposed a wide range of visualization methods to communicate the terrain and the spatial data. To this end, we presented various visualization techniques. In order to facilitate scalability on presentation side, we investigated how to show different levels of abstraction for each aspect. For *terrain*, we distinguished *sketching*, *shading*, and *texturing*, while for the numeric *data* we differentiated between depicting *aggregated values* and *individual values*. For geometric data, we similarly provided different visualization strategies to facilitate multiple levels of abstraction. Though we demonstrated them using the example of trajectory representations, many of the strategies also apply to other geometric data, such as buildings, which we examined in a student master’s thesis (Trappe (2014)), which the author of this dissertation supervised.

We then discussed methods to improve the presentation quality. In this context we

addressed the conveyance of uncertainty and the augmentation of spatial awareness. For showing *uncertainty* in the data, we argued that *intrinsic and extrinsic techniques* provide different degrees of detail. *Spatial awareness*, in turn, can hardly be divided into techniques for different abstraction levels. But with the distinction between *additional graphical primitives* and *visual effects* we are able to provide methods with different complexities on the level of computation.

Having all these different visualization options at hand, we can now focus on scalable presentations regarding the given task. For this purpose, we will in the next chapter propose strategies that address the question how to combine the individual data aspect into one scalable presentation.



## Chapter 5

# Advanced Visualization Strategies

In this chapter, we will address the question how to scale the visual design of data presentations in terrain with changing tasks. To this end, we need to combine the visualization methods introduced in the previous chapter to one holistic view. Since hardly all the data can be shown simultaneously and in detail, we need to focus on the relevant information. For this purpose, we pursue two strategies.

Firstly, we scale the presentation in conformity with the task at hand on a global level. We will do so by prioritizing one data aspect, for example, the terrain or the data, and showing this aspect with more detail, while reducing emphasis on the others. Accordingly, the first part of this chapter is dedicated to a design strategy called Prioritization. In this context, we will discuss how different visualization options allow us to communicate correspondingly prioritized data aspects.

Secondly, we adjust the presentation based on the given task on a local level. Hereby, we focus on regions of interest—either with regard to the spatial frame-of-reference or the data space—and adapt the visualization to primarily communicate these particular data. To this end, we utilize the concept of Focus & Context to scale the visualization to predominantly communicate the relevant part of the data, while still giving context information. By applying different strategies, the focus region can be specified with regard to both, the terrain and the data. Respectively, an adequate representation of the focus, the context, and an intermediate area is derived.

The last part of this chapter is dedicated to the issue of evaluating our visualization design towards given constraints. To this effect, we propose a novel perception-based measure that facilitates the identification of areas attracting attention and the detection of visual clutter. To incorporate these measure into the design process, we propose a method that derives them automatically and in real-time. Using these measures, we will afterwards assess various proposed visualization designs from [Section 5.1](#) and [5.2](#).

## Contents

---

5.1	Prioritization . . . . .	<b>70</b>
5.1.1	Motivation . . . . .	70
5.1.2	General Approach . . . . .	71
5.1.3	Design Strategies . . . . .	72
5.1.4	Discussion . . . . .	77
5.2	Focus & Context . . . . .	<b>78</b>
5.2.1	Motivation . . . . .	78
5.2.2	General Approach . . . . .	78
5.2.3	Procedure and Design Strategies . . . . .	79
5.2.4	Discussion . . . . .	85
5.3	Evaluation using Saliency Information . . . . .	<b>88</b>
5.3.1	Motivation . . . . .	88
5.3.2	Visual Clutter . . . . .	89
5.3.3	Saliency . . . . .	91
5.3.4	Evaluation of Visualizations . . . . .	94
5.3.5	Discussion . . . . .	98
5.4	Summary . . . . .	<b>100</b>

---

---

Major parts of [Section 5.1](#) are based on the aforementioned journal article:

[Dübel et al. \(2017\)](#): Steve Dübel, Martin Röhlig, Christian Tominski, and Heidrun Schumann. *Visualizing 3D Terrain, Geo-spatial Data, and Uncertainty* Informatics, 2017, 4

The author of this thesis conceived larger parts of the approach and wrote major parts of the paper. The implementation of the uncertainty presentations is based on the master’s thesis *Flexible Visualization of Weather Data in a Spatial Context with Regard to Uncertainty* by Martin Röhlig ([Röhlig \(2014\)](#)), which the author substantially supervised.

---

Major parts of [Section 5.2](#) chapter were released in the following publication:

[Richter et al. \(2017\)](#): Christian Richter, Steve Dübel, and Heidrun Schumann. *Visual Analysis of Geo-spatial Data in 3D Terrain Environments using Focus+Context* EuroVis Workshop on Visual Analytics (EuroVA), The Eurographics Association, 2017

The approach is based on the master’s thesis *Application of Focus & Context techniques for visualization of weather data embedded into 3D terrain* by Christian Richter ([Richter \(2016\)](#)), which the author of this thesis supervised. The author conceived and implemented larger parts of the approach and also wrote major parts of the paper. For this thesis, the article was rearranged and the figures have been replaced. Moreover, most parts were reworded to fit the concept of this thesis.

Moreover, the labeling approach, which will briefly be discussed in this section, was submitted as the following patent:

[Richter et al. \(2019\)](#): Christian Richter, Martin Luboschik, Steve Dübel, and Stefan Müller-Divéky *A method for generating a 2D image of a 3D surface* Pending Patent Application (since 2017).

The approach is based on the idea developed by [Luboschik et al. \(2008\)](#). The approach described in the application was conceived by Christian Richter and was designed and implemented through collaborative work with the author.

---

The saliency-based evaluation strategy in [Section 5.3](#) is a novel approach that was conceived and implemented in collaboration with Christine Ripken within the scope of a master’s thesis *Using Saliency Information for priority-based information visualization in 3D Terrain* ([Ripken \(2019\)](#)) that the author of this thesis supervised. This approach is not published yet.

---

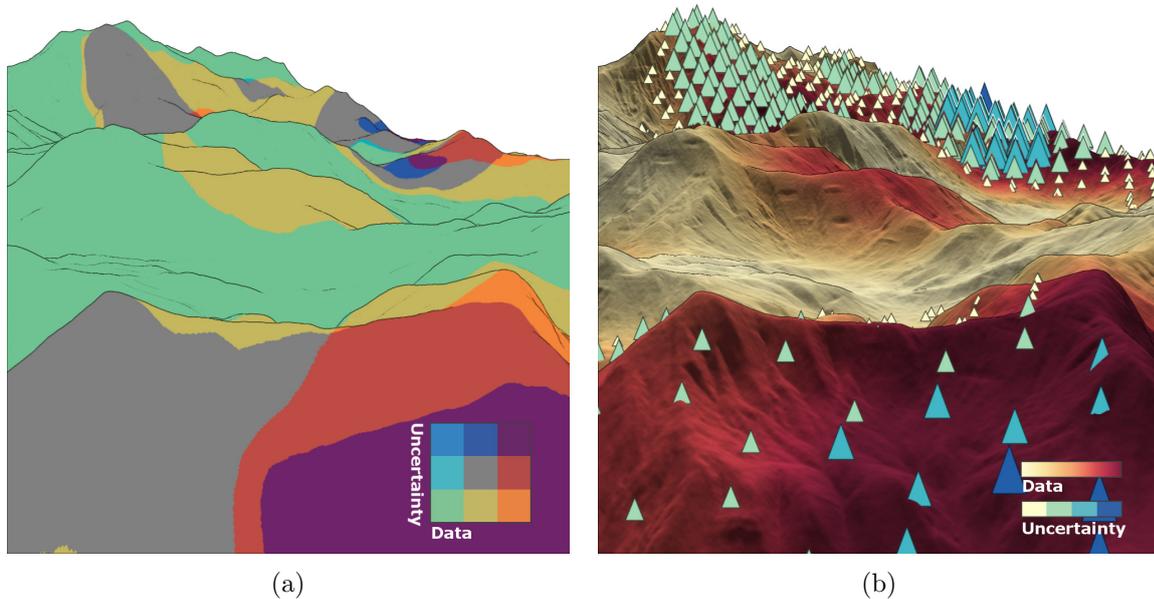


Figure 5.1: Visualization of 3D terrain, wind speed forecast data, and forecast uncertainty. (a) The terrain is depicted abstractly using contour lines, and a bivariate color scale with few discrete colors is used to encode the data (green to red) and the uncertainty (green to blue); (b) The terrain is depicted faithfully by shading its surface, a continuous color scale (yellow to red) is used for the data, and uncertainty is represented by color-coded (yellow to blue) triangular glyphs.

## 5.1 Prioritization

We have shown in the previous chapters that terrain rendering alone is already challenging with regard to real-time processing and image complexity. Showing the data along with the terrain aggravates the problem. Besides these two aspects, the interpretation of data strongly depends on knowledge about the data quality. Hence, visualizing the data's associated uncertainty is also an important objective. In order to communicate spatial data together with uncertainty within the terrain, we need to choose suitable visualization techniques for each aspect. Existing approaches usually consider only two out of the three aspects: either data in their spatial frame-of-reference or data together with their uncertainty (cf. Chapter 2). Visualizing all three aspects in a single comprehensible image remains an open question and will be addressed in this section. The choice, which visualization technique is suitable, however, depends on the scenario at hand. The scenario in turn determines the given task towards which we need to scale our visual design. By scaling, we eventually mean the choice and change of suitable design options proposed in Chapter 4.

### 5.1.1 Motivation

If terrain, data, and uncertainty are depicted together, the resulting visual representations can be quite complex. This is particularly true if all three aspects are communicated with a high degree of detail. A simple example shall illustrate this problem.

Figure 5.1 shows a 3D terrain with color-coded weather forecast data and associated uncertainty information. According to our discussion, how to depict terrain, data, and

uncertainty with low detail, [Figure 5.1a](#) shows an exemplary, combined visualization. The structure of the terrain is communicated by drawing silhouettes only, which provides a general overview but neglects local features of the terrain. Data and uncertainty are encoded by a discrete bivariate color scale. Its few different colors support a simultaneous identification of data and uncertainty values, but only coarsely.

In contrast, [Figure 5.1b](#) shows the terrain, the data, and uncertainty at high fidelity. Shading is added to show more details of the terrain, and a continuous color scale is applied to encode data values more precisely. Additionally, color-coded glyphs represent uncertainty in a prominent form.

While theoretically being superior, this detailed visual representation actually has some drawbacks. Comprehending data values accurately involves additional mental effort, as it requires frequent lookups of continuous colors in the color legend. Moreover, shading the terrain leads to different levels of brightness, which in turn distorts the perception of colors. Parts of the terrain that are less illuminated (e.g., in shadows) might not show any color at all. This could lead to wrong interpretations of color-coded data and uncertainty, and eventually to false conclusions. On top of this, uncertainty is encoded using additional graphical primitives, which theoretically better expose the degree of uncertainty. However, these graphical primitives occlude parts of both the terrain and the color-coded data.

This simple example demonstrates that representing 3D terrain, data, and uncertainty at a sufficiently high level of detail is challenging. Apparently, it is not possible to simply show each individual aspect with full detail. It is rather necessary to scale the interdependencies of the involved visual encodings.

We address this issue by an elementary approach: Prioritization. We prioritize a selected aspect of particular interest (terrain, data, or uncertainty), depending on the given task. The aspect of interest is in detail, while the remaining two aspects are represented at a lower degree of detail. We hereby utilize in particular the design options proposed in [Section 4.2](#).

### 5.1.2 General Approach

The concept of Prioritization is a widely applied strategy in general and in the context of visualization in particular. For a long time, emphasis ([Hall et al. \(2016\)](#)) has been used to accentuate regions of interest and to dim less-relevant ones. For another example, [Beecham et al. \(2016\)](#) investigated the idea of creating different levels of detail to prioritize the visualization of spatial, temporal, and descriptive aspects of spatio-temporal data. Their study suggests that Prioritization is a feasible approach to deal with complex and potentially conflicting visualization requirements.

Our approach is conceptually similar to their work. We pursue Prioritization by considering different levels of detail that are to be combined into a single holistic visualization. However, instead of using a 2D map, we consider—with respect to our application scenario—3D terrain to visualize the spatial frame-of-reference. Furthermore, instead of the temporal aspect we consider the uncertainty of data.

To make our case clear, we will first briefly revisit our flight scenarios from [Section 3.1](#). Together with domain experts we have derived the actual tasks that need to be facilitated by visualization for each scenario. Thus, we will identify the requirement of prioritizing either the 3D terrain, the spatial data, or the uncertainty. For the purpose of a better readability, we will abbreviate these aspects by the letters *T* (Terrain), *D* (Data), and *U* (Uncertainty), respectively.

In accordance to the previous chapter, we will introduce two fundamental strategies for each aspect: One generates a detailed representation, the other a more abstract depiction with less detail.

**Tasks for Flight Scenarios** The visualization goal for our flight scenarios are derived from the pilot's task at the respective situation. In all scenarios, spatial awareness is relevant, and thus the geometry of  $T$ . For  $D$ , we consider weather data, including hazardous weather zones and weather forecasts. The forecasts are naturally associated with some uncertainty  $U$ . In summary, all three aspects  $T$ ,  $D$ , and  $U$  have to be considered, yet we need to scale the visual design with respect to the importance that varies with each different scenario. Typically, a selected aspect is of specific interest (denoted as  $^+$ ), while the others communicate context information (denoted as  $^-$ ).

- *Take-off and landing approach*  $T^+D^-U^-$ : During take-off and approach the flight altitude is low and natural obstacles in the terrain (e.g., mountains) constitute a principal threat to an aircraft. Therefore, the topography of  $T$  should be communicated accurately and as detailed as possible. Weather conditions and forecast reliability have to be considered as well, but they are not as relevant as the terrain. Therefore, it is sufficient to provide a qualitative representation of  $D$  and  $U$ .
- *Cruise Flight*  $T^-D^+U^-$ : In overflight scenarios, the flight altitude is quite high ( $> 10\text{km}$ ). Hence, fine-grained features of the terrain are not essential. The visual representation of  $T$  merely serves as an overview to facilitate orientation. Weather forecast data, on the other hand, can be of primary interest, in particular the spatial distribution and characteristics of hazardous weather zones. Therefore,  $D$  should be represented at high detail. Information about potential uncertainty are needed as well. But providing a coarse overview of  $U$  to estimate the forecast quality is usually sufficient.
- *Re-planning*  $T^-D^-U^+$ : Safety-related issues, such as dangerous weather conditions, can make an abort of pursuing the current route or of a landing approach necessary. Subsequently, the pilot needs to re-plan the next steps of the flight. This re-planning requires the considerations of the hazardous weather zones to adapt the flight route. Before planning a new route, the reliability of the corresponding forecast has to be verified. Hence, the presentation of  $U$  is important and should be prioritized. Weather forecast data are needed as well, yet arguably, individual quantitative data values might not be needed. A qualitative distinction of safe zones and hazardous zones is sufficient. Similarly, local terrain features are not as relevant. Therefore,  $D$  and  $T$  can be shown at less detail.

Next, we will illustrate the basic rendering strategies to account for the different degrees of relevance of  $T$ ,  $D$ , and  $U$ .

### 5.1.3 Design Strategies

In Section 4.2 we discussed several design options to communicate the individual data aspects. In this section, we determine two fundamental design strategies for each aspect. Depending on whether an aspect is categorized as  $^+$  or  $^-$ , the strategies will aim for a higher or lower level of abstraction, which we discussed in the previous chapter. This allows us to

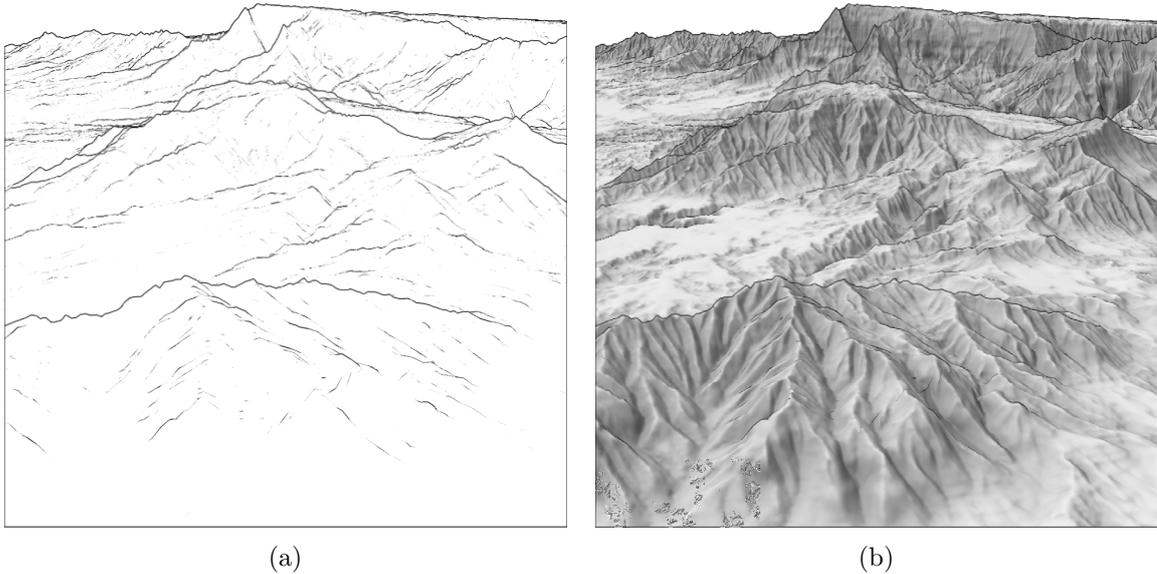


Figure 5.2: Strategies for terrain: (a) Showing contours through line drawing provides an overview of basic terrain features and (b) shading together with emphasized silhouettes represents the terrain at high fidelity.

adjust the visual representation on a fundamental level. Obviously, a subsequent fine-tuning of the individual strategies is obligatory, however this will be considered later on.

**Representing Terrain  $T$ :** We have looked at a large variety of terrain rendering techniques. For our two levels of abstraction, we make a rather abstract distinction between representing 3D geometry of terrain either by sketching or by shading.

- *Sketching for  $T^-$ :* Our fundamental strategy for less-detailed representations of the terrain is to use sketching. Silhouettes, contours, or characteristic curves communicate the most significant geometric features and emphasize object boundaries or provide shape details. Specifically, we draw lines based on both the curvature of the terrain model and variations in depth when viewed from the camera's position. [Figure 5.2a](#) shows our approach.
- *Shading for  $T^+$ :* We specify shading as the fundamental strategy to visualize terrain at high fidelity. With respect to the different illumination methods that can be applied to shade a terrain surface, we choose an approximative global illumination, that is, ambient occlusion. This represents a good balance between image quality and render time and local terrain features become emphasized. Additionally, we add lines for the silhouette to highlight the coarser structures of the model and thus strengthen the spatial impression. This rendering style is demonstrated in [Figure 5.2b](#).

**Representing Data  $D$ :** In accordance to our definition of distinct levels of abstraction for data, we distinguish between showing data abstractions and individual data values.

- *Abstractions for  $D^-$ :* We already demonstrated that showing data through segmented encodings, rather than depicting individual values through continuous encodings, is a

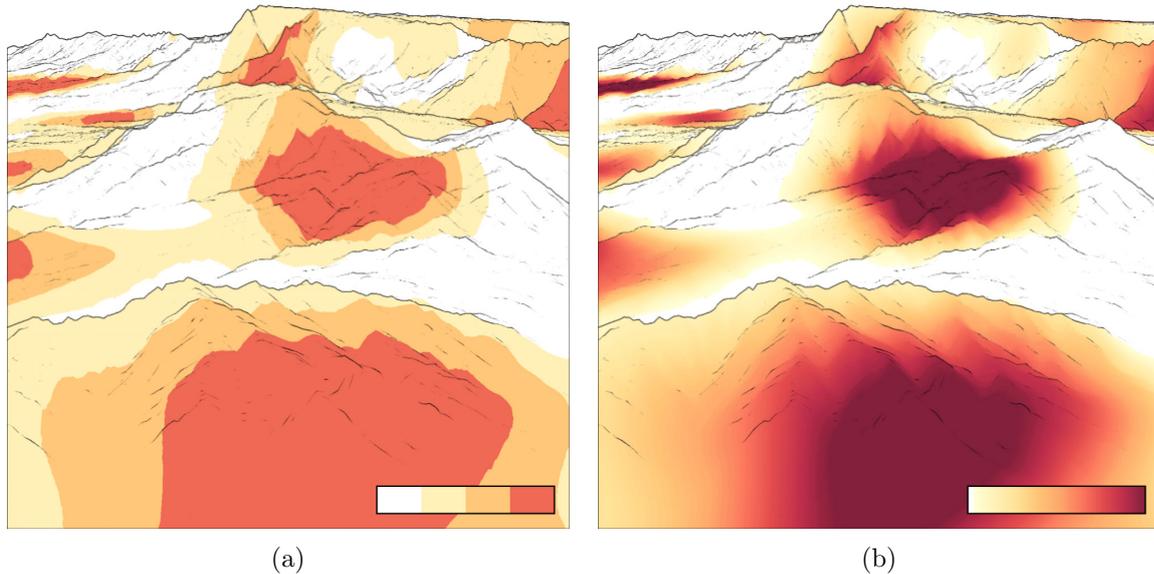


Figure 5.3: Strategies for data: (a) Aggregation of data showed by a segmented color scale and (b) individual data values showed by a continuous color scale.

fitting approach to achieve higher levels of abstractions. To this end, individual data values are composed into larger units or groups. These units capture basic characteristics of the data and provide a good overview, yet they lack details. Figure 5.3a shows an example where data abstraction is achieved by using a color scale with four segments. The figure shows the severity of thunderstorm cells mapped onto the terrain surface. The different colors classify regions into safe zones (white), potentially safe zones (light orange), hazardous zones (orange), and no-go zones (red). The four categories of zones can be distinguished easily, whereas the underlying data values are abstracted away.

- *Individual data values for  $D^+$* : If there is a need to grasp more details, a trace back to the individual data values can be necessary. For example, detailed data could be required to figure out whether a hazardous weather zone rather tends to be a no-go zone. To continue our previous example, individual data values are communicated using a continuous color scale, where each data value is associated with an individual color. Figure 5.3b shows the individual values of radar reflectivity (dBZ), which is the quantitative measure behind the previous qualitative categorization of the intensity of thunderstorm cells. Now, the individual data values allow for a better recognition, and a smoother transition between the four groups is apparent.

**Representing Uncertainty  $U$ :** In order to provide two levels of abstraction for visualizing uncertainty, we use either intrinsic and extrinsic techniques.

- *Intrinsic encoding for  $U^-$* : Intrinsic encodings support qualitative assessments with about three or four distinguishable levels Carpendale (2003). To this effect, we apply either transparency or noise. The degree of transparency and noise in such visualizations should be within reasonable limits to ensure that colors, and hence, data values

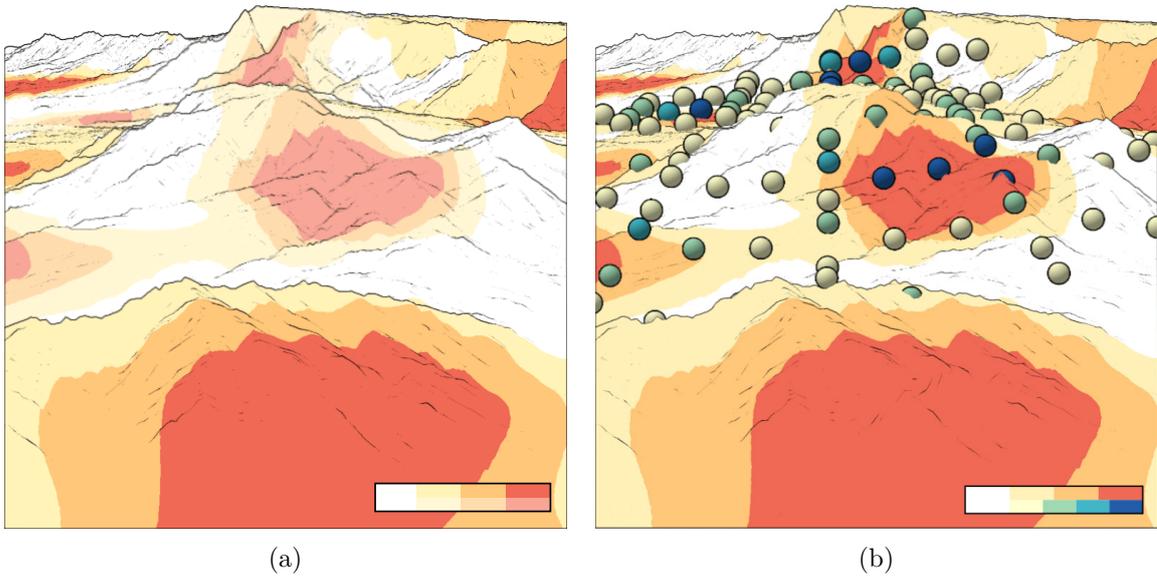


Figure 5.4: Strategies for uncertainty: (a) Uncertainty is encoded intrinsically by mapping it to transparency (0 and 50 %) and (b) uncertainty is encoded extrinsically by mapping it to the color of additional circles embedded into the terrain.

are still identifiable, even if they are affected by higher uncertainty. Nevertheless, a rough estimation of data values, which validity is very low, is typically sufficient.

Exemplarily, [Figure 5.4a](#) shows uncertainty intrinsically encoded by modifying transparency in two steps. Transparency (50 %) modulates the perception of color for uncertain data values, whereas colors in certain regions remain unchanged. Thus, we are able to communicate the fact that data values are uncertain, but the particular degree of uncertainty can hardly be recognized.

- *Extrinsic encoding for  $U^+$* : Extrinsic techniques incorporate additional graphical elements, which offers a greater flexibility, better visual emphasis, and the potential to encode more details. [Figure 5.4b](#) illustrates our extrinsic encoding via color-coded glyphs embedded into the scene. In contrast to the intrinsic example, the glyph's color coding conveys a few different levels of uncertainty. The glyphs clearly outline the areas that are afflicted with uncertainty. This allows to communicate exact locations of uncertainty, which is difficult with intrinsic encodings, and even facilitates the presentation of missing data values.

Having now two fundamental rendering strategies for each aspect (sketching and shading for the terrain  $T$ , segmented and individual data values for the representation of data  $D$ , as well as intrinsic and extrinsic encoding for the uncertainty  $U$ ), we can now combine these strategies to generate differently prioritized visual representations of  $T$ ,  $D$ , and  $U$ .

[Table 5.1](#) summarizes the combinations of rendering strategies when prioritizing one aspect. The prioritized aspect ( $^+$ ) is visualized by the strategy that provides details. The respective other two aspects ( $^-$ ) are presented with the strategies that satisfy the need for contextual overviews. The resulting visualization designs will be explained by means of examples in the following.

Table 5.1: Prioritized combination of terrain, data, and uncertainty.

Prioritization	Strategy for T	Strategy for D	Strategy for U
$T^+D^-U^-$	<b>Shading</b>	Segmented encoding	Intrinsic
$T^-D^+U^-$	Sketching	<b>Continuous Encoding</b>	Intrinsic
$T^-D^-U^+$	Sketching	Segmented encoding	<b>Extrinsic</b>

**Prioritizing Terrain:** As previously described, take-off and approach scenarios require a detailed representation of the terrain, while data and uncertainty are less significant. Thus, we need a detailed presentation of the terrain, which we achieve by visualizing its surface by shading. An adequate illumination model is applied to expose important detailed features. Moreover, interference with or over-plotting of the terrain features should be reduced to a minimum. Hence, data and uncertainty are presented with less detail. According to our render strategies, this means showing data abstractions instead of individual data values. By using a segmented color scale with only a few distinct colors, an overview of the data is warranted. For uncertainty, intrinsic techniques should be used, because they work without additional graphical elements, and thus hardly introduce occlusion. As a typical intrinsic approach, we vary the transparency of the data-encoding colors.

Figure 5.5a shows our combined visualization with terrain, data, and uncertainty, where the terrain is prioritized. The fine variations in color in this figure particularly communicate the details of the terrain topography. The data, on the other hand, are conveyed by a few color gradations only. This reduces the interrelation of these two encodings and still makes the important features of the data easily identifiable, even in shadowed areas of the surface. The lack of opacity for uncertain data, in turn, allows to quickly grasp the existence of uncertainty, though no further details.

**Prioritizing Data:** We mentioned that the tasks of the pilot in overflight scenarios require prioritized data representations. As the focus lies on the data, individual data values are visualized. In our case, this is done by using a continuous color scale. To prevent that the data presentation gets impaired, terrain and uncertainty are shown with less detail. A shaded terrain surface would considerably distort the perception of the color-coded data. Therefore, the fidelity of the terrain rendering is reduced to sketching of significant edges. Uncertainties are presented by intrinsic techniques to avoid occlusions of the data values. For another intrinsic example, we now superimpose the data presentation with noise. Similar to what has been said before, a certain maximum density of the noise should not be exceeded to maintain the legibility of data values.

Figure 5.5b shows our combined visualization where the data are prioritized. The variation in color now communicate only the details in data, while the terrain features are outlined through contours. Certain data are clearly interpretable. Only uncertain data are made noisy, which allows to recognize the data quality.

**Prioritizing Uncertainty:** The task of our last scenario involves re-planning for alternative flight routes and thus uncertainty plays a significant role. As uncertainty is the most relevant aspect now, we visualize it using an extrinsic technique. We embedded glyphs into the terrain to prominently represent uncertainty. The locations of the glyphs encode where

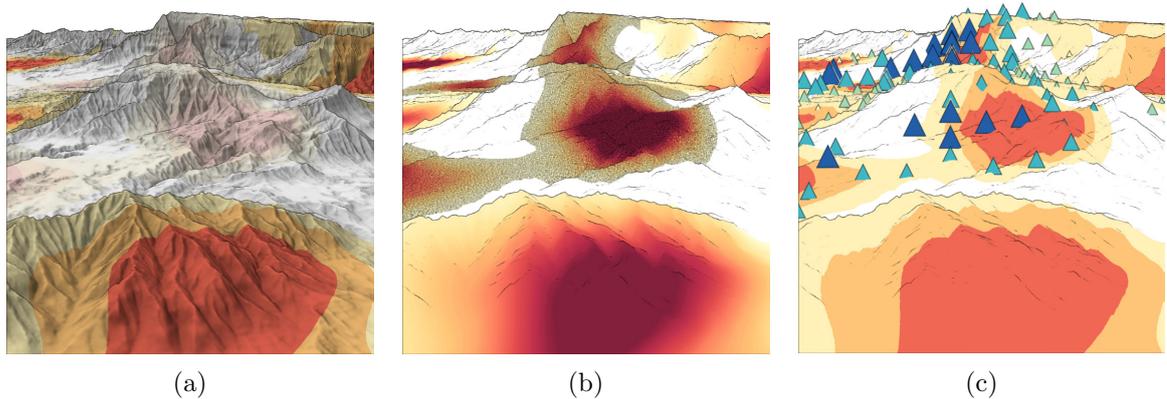


Figure 5.5: Scalability of the visualization design with respect to the given priority: (a) terrain first, (b) data first, and (c) uncertainty first.

uncertainty occurs, and even the degree of uncertainty can be mapped to a certain extent to properties of the glyph, such as color or size.

As terrain and data are deemed less important in this case, the occlusion caused by the glyphs is acceptable, and terrain and data can be depicted at lower detail. Again, the terrain is represented by contours, and the data are shown in an abstracted fashion using a segmented color scale. This prevents too much interrelations between the encoding of the terrain and data with the encoding of uncertainty and relieves the visual budget.

Figure 5.5c illustrates our design where the focus is set on the data’s uncertainty. As the figure shows, uncertain regions are easily identifiable. We use triangular glyphs whose color and height vary depending on the underlying uncertainty. Since height is difficult to interpret in 3D presentations, we use this only for redundant encoding and thus, support a quick conveyance of locations where high uncertainty occur.

Comparing the three images in Figure 5.5, we can see that details of the terrain topography are conveyed only in the first one, while they recede into the background for the other two. Similarly, individual data values can only be seen in the middle figure and uncertainty is less prominently encoded in the first two images, while being prominently represented in the third one. Accordingly, these images can facilitate their respective purpose to prioritize one data aspect in the presentation. We will provide a more detailed analysis of these visual designs in Section 5.3.

#### 5.1.4 Discussion

In conclusion, the results of our prioritization strategy show that we can scale the visualization of terrain, data, and uncertainty to different tasks at hand. We demonstrated that already the simple combination of our fundamental rendering strategies, collected in Table 5.1 and shown in Figure 5.5, leads to quite different visual representations suited for different scenarios.

Within each of the fundamental strategies, however, there are numerous design options to fine-tune the visualization. For example, which shading technique should be applied and how should it be parameterized? How many segments and which hues should a color scale have? What shapes should be used for encoding uncertainty extrinsically? The design space

is considerably large. On the one hand, this allows us to better attune the visualization to application-dependent requirements. Yet, on the other hand, it is not clear beforehand how certain design decisions will affect the overall visualization outcome. Thus, we need both, ways to evaluate our visual design and tools to test potentially useful visualization variants. For the former, we can pursue two strategies, using feedback from users and domain experts on the one hand and applying technical, perception-driven evaluation methods on the other. Both will be discussed later in this thesis, in [Section 5.3](#) and [Section 7.2](#), respectively. For the latter, we will introduce our visualization framework TEDAVIS in [Section 6.1](#).

## 5.2 Focus & Context

The previous section addressed the question how to emphasize the relevant parts of information in a visualization using the concept of Prioritization. This concept focused on scalability of visualization designs on level of one data aspect, that are, the terrain, the data, and the uncertainty. This section takes a different approach to this question by scaling the visualization more locally, on level of the data elements. In particular, we utilize the concept of Focus & Context, which is widely used for information visualization. We, however, enhance this concept for the use in presentations of spatial data in 3D terrain.

### 5.2.1 Motivation

For a long time, the Focus & Context approach has been subject of extensive research. In [Section 2.4](#) we already summarized related work. Focus & Context allows to concentrate on the most relevant parts of the data (focus) while simultaneously providing an overview on related information (context). Typical approaches consider a focus region either in the data space (e.g., [Doleisch et al. \(2003\)](#)) or in the reference space (e.g., [Trapp \(2013\)](#)). Since we aim to communicate the characteristics of data as well as geometric details of the terrain, our approach addresses both aspects. Thus, we propose the use of two types of foci—the spatial focus regarding the reference space and the data focus regarding attribute space. Both foci depend on each other. Accordingly, the spatial focus can be derived from the data focus and vice versa.

### 5.2.2 General Approach

Our approach can be described as follows: We distinguish between a *spatial focus* and a *data focus*. The spatial focus defines a spatial region of interest and can be located at a point (e.g., a landmark), along a path (e.g., a trajectory of an aircraft) or within a volume (e.g., the airspace around an airport). The data focus defines certain aspects of interest and can be either a single attribute value (e.g., condensation point) or an interval of attribute values (e.g., hazardous weather conditions).

By specifying one type of focus, the other one can be determined automatically. Accordingly, if a spatial focus is specified, the corresponding data focus consists of all data points within the selected region. Likewise, if a data focus is specified, the spatial focus consists of all regions where these data values occur. Through this, we can scale our visual design according to the elementary tasks of exploratory analysis—direct and inverse lookup, as described by [Andrienko and Andrienko \(2006\)](#). For direct lookup, the user is interested in *what* data values are located at a certain spatial region. Consequentially, one selects this region to specify the spatial focus. For inverse lookup, the user wants to know *where*

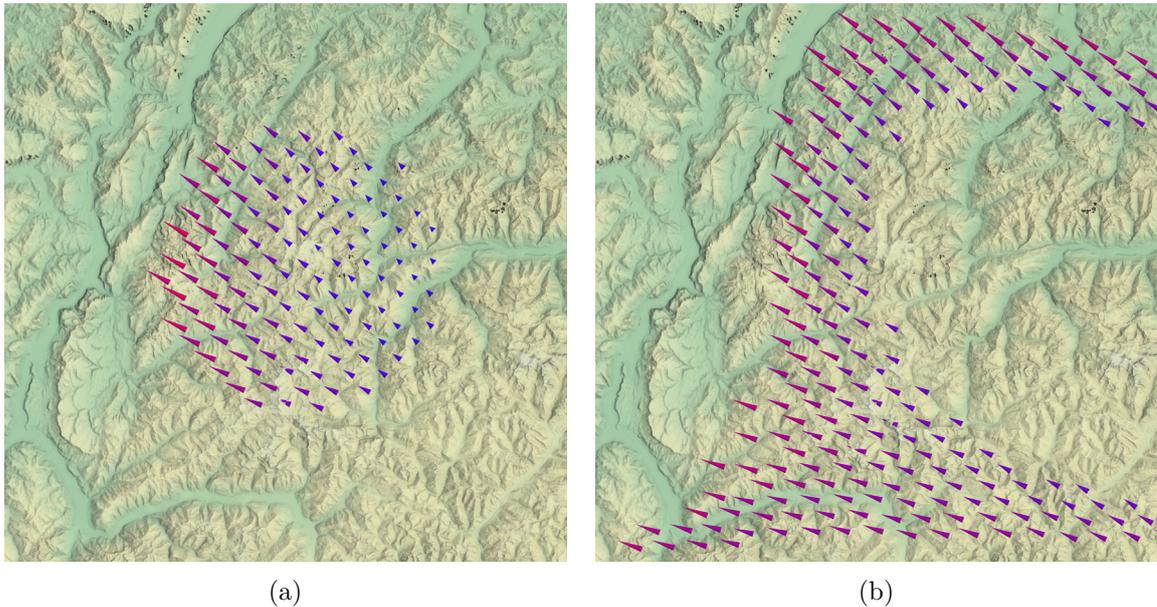


Figure 5.6: Distinction between a spatial and a data focus. (a) The spatial focus shows what data are located at a given position. (b) The data focus shows where data that possess certain characteristics are located.

certain data values are located and thus selects the data focus. In Figure 5.6 this procedure is illustrated for terrain-embedded wind glyphs. The result after specifying a spatial focus region is shown in 5.6a, where the data glyphs are presented for a certain area. On the other hand, the result of specifying a data focus is depicted in 5.6b, where the data glyphs are presented if the data values, in this case wind speed, lie within a certain interval.

Since there are two types of foci, there are also two types of context: the *spatial context* and the *data context*. Both types of context contain abstracted information. However, just reducing details might not be sufficient to keep visual clutter under control, for instance due to a rather complex terrain or because of too many data elements. Therefore, it is reasonable to show data only for a sub-region, the *intermediate context*, whereas the remaining context, the spatial context, depicts solely the terrain. In this way, the intermediate context forms a transition between the focus and the spatial context. This strategy matches the midground concept of Hall et al. (2016). The schematic diagram in Figure 5.7 again clarifies our strategy to divide the visualization into focus, intermediate context, and context regions.

### 5.2.3 Procedure and Design Strategies

In order to specify the different types of foci and contexts and to create tailored visual representations for both regions, we developed the following procedure that consists of three fundamental steps:

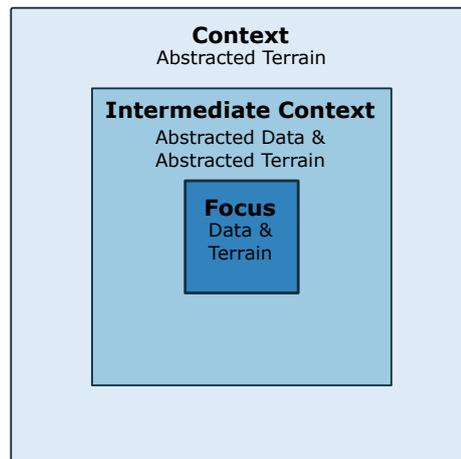


Figure 5.7: Division of the visualization into focus, showing detailed data and terrain, intermediate context, showing abstracted information, and context region, showing abstracted terrain only.

**1. Selection of foci:** Initially, either a spatial focus or a data focus needs to be selected. There are two ways to select the focus:

- *Predefined:* The spatial focus can be computed based on given constraints, such as geographical characteristics. Similarly, the data focus can be calculated from predefined data constraints.
- *Interactive:* The focus is selected interactively by the user. Hence, the spatial focus can be selected by positioning marker or lens, whereas the data focus can be specified by selecting relevant attribute values.

While defining one focus—either spatial or data focus—the respective other focus is automatically determined.

**2. Computation of contexts:** Based on the focus, the intermediate context is computed in either of the two following ways:

- *Based on spatial relationships:* The intermediate context is determined depending on the distance to the spatial focus.
- *Based on data relationships:* A similarity measure is computed. Regions that show similar data values are assigned to the intermediate context.

The remaining regions then form the spatial context.

**3. Adjustment of the visual representation:** The final step is the adjustment of the terrain's and the data's representation within focus and context, respectively. Focus areas are emphasized and represented in detail, while context areas are de-accentuated and show less detail in order to provide an overview. To this end, we utilize and extend our previously described visualization options to facilitate the following strategies:

- *Levels of abstraction* allows to simplify the visual representation, specifically for the context regions.
- *Highlighting* emphasizes important regions, particularly the focus region.
- *Distortion* alters the presentation space to achieve a better spatial proportion between focus and context regions.

We realized our described Focus & Context concept by designing a pipeline to match the aforementioned three-step procedure: focus selection, computation of the context, and adjustment of the representations. In the following, we will describe our approach in more details and propose appropriate design strategies. For illustrating purpose, we exemplarily utilize wind data. In particular, we use arrow glyphs that encode the wind direction by their arrow tip and the wind speed by their length and color (e.g., [Figure 5.8](#)). In general though, our approach can be applied to many different data types to facilitate visual scalability with regard to important regions in attribute space or reference space.

**Focus Selection** Our concept provides multiple options to specify the focus. Based on the task at hand, we must first choose whether we want to set a spatial focus or a data focus. For instance, for a landing scenario, the pilot needs to know wind condition at the destination airport. Thus, a spatial focus would be set at the desired location. During a planning scenario, the pilot would need to identify areas of bad weather conditions and thus specify data characteristics of interest.

According to the scenario, both type of foci can be chosen either manually, for instance by selecting a region of interest or relevant data characteristics, or computed automatically, for instance to be always in the vicinity of the user or to show always the most critical wind conditions.

In case of a spatial focus, we can choose various shapes to select the region of interest. In particular, we support shapes of different dimensionality, that are, a point (0D), a path (1D), an area (2D) and a volume (3D). [Figure 5.8](#) illustrates these options. In [5.8a](#) the focus is set on a point of interest, here at the location of a mountain. Accordingly, wind glyphs for this position are depicted. [Figure 5.8b](#) shows the focus region along a path, in this example a flight route. For both cases, point and path, the user can specify a certain distance threshold so the focus region includes data within a certain proximity, which is exemplarily shown for the path. [Figure 5.8c](#) and [5.8d](#) show the focus for a 2D area and a 3D volume, respectively. We provide either circular or square-shaped focus regions for 2D and sphere or cuboid shapes for 3D. Two-dimensional areas can be useful if wind conditions need to be detected for certain regions, while three-dimensional foci include data within a certain altitude. The latter one, of course, makes only sense if 3D data are available.

In case of the data focus, we can chose either a certain data characteristic or an interval of data characteristics. For instance, [Figure 5.9a](#) shows the data with the highest wind speed in the current reading and [Figure 5.9b](#) shows wind data that lies within an interval, here the average of the current reading.

**Computation of the intermediate context** The computation step addresses two tasks: calculating the intermediate context and gathering the corresponding data. The intermediate context hereby supports the comprehension of correlation, either on a spatial or on

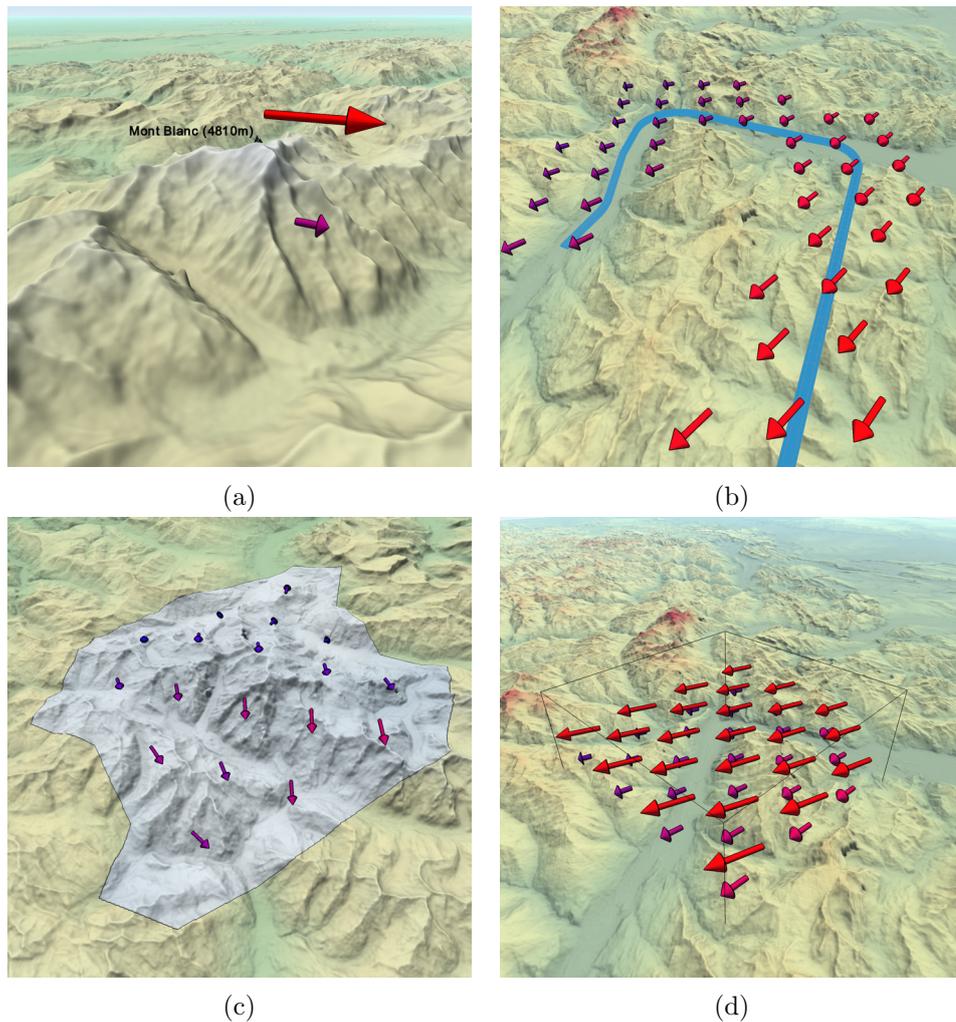


Figure 5.8: Strategies for spatial focus selection: (a) for a point of interest, (b) for a path of interest, (c) for an area of interest, and (d) for a volume of interest.

attributive level, based on the focus selection before. Hence, we utilize the spatial relationship if previously a spatial focus has been selected. Accordingly, the space in a specified distance is assigned to the intermediate context. The distance threshold can be interactively adjusted and directly affects the size of the intermediate context. Since we facilitate foci of different dimensionality, different strategies to determine the distance are possible. We base our computation always on the 2D projection of the spatial focus onto the terrain. From this, we calculate the spatial intermediate context either as a circular (Euclidean distance) or as a rectangular shape (Chebyshev distance), depending on the shape of the focus. Though it would be possible to base the distance computation on the real 3D position, our approach is sufficient and easy to retrace with respect to our typical scenarios.

If a data focus was selected, we utilize data relationships. In general, different similarity measures are applicable to this effect. We apply the following simple distance measure: Two data values are considered similar if the difference of these values is smaller than a given threshold. If a data interval was specified, the same measurement can be applied to the endpoints of the interval.

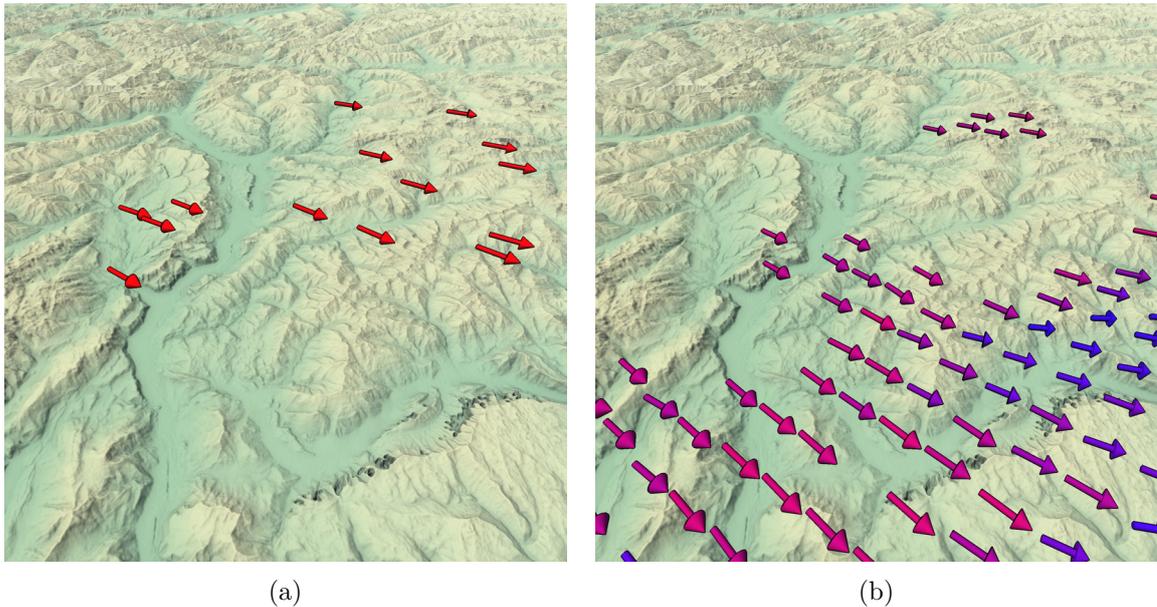


Figure 5.9: Strategies for data focus selection: (a) showing the data for a certain value and (b) showing the data within a certain interval.

Figure 5.10 shows the intermediate context calculated either by a spatial (a) or an attributive relationship (b). The specified maximum distance directly affect the size of the intermediate context and thus provide means to scale the presentation to the current scenario. After gathering the data of the intermediate context, the adjustment of the presentation is performed.

**Adjustment of representations** Finally, appropriate representation styles need to be chosen in order to accentuate terrain and data in the focus, while attenuate them in the intermediate context. In contrast to our concept of Prioritization in Section 5.1, we do not emphasize a whole data aspect, but only the particular data subset in focus. For that purpose, we utilize our previous strategy of multiple levels of abstractions, but additionally exploit highlighting and distortion.

**Abstraction** Using our strategy of abstraction, we can simplify the visualization with regard to both the information amount (what to show) and their representation (how to show). To reduce the amount of represented information, we abstract the data, for example, through segmentation or aggregation. To simplify their depiction, we abstract the encoding, for example, by using less complex visual primitives. Both concepts can be applied to terrain and data.

For abstracting the terrain representation, we use again sketching, rather than shaded surfaces. What this means for Focus & Context is illustrated in Figure 5.11a. The terrain in focus is depicted prominently with all the details, while in context regions, the surface is only outlined by contours.

For data abstraction, we pursue two strategies. For one thing, we aggregate data values in spatial proximity, such as by averaging. After aggregation, one wind glyph represents multiple data points. In this way, the number of data elements is reduced

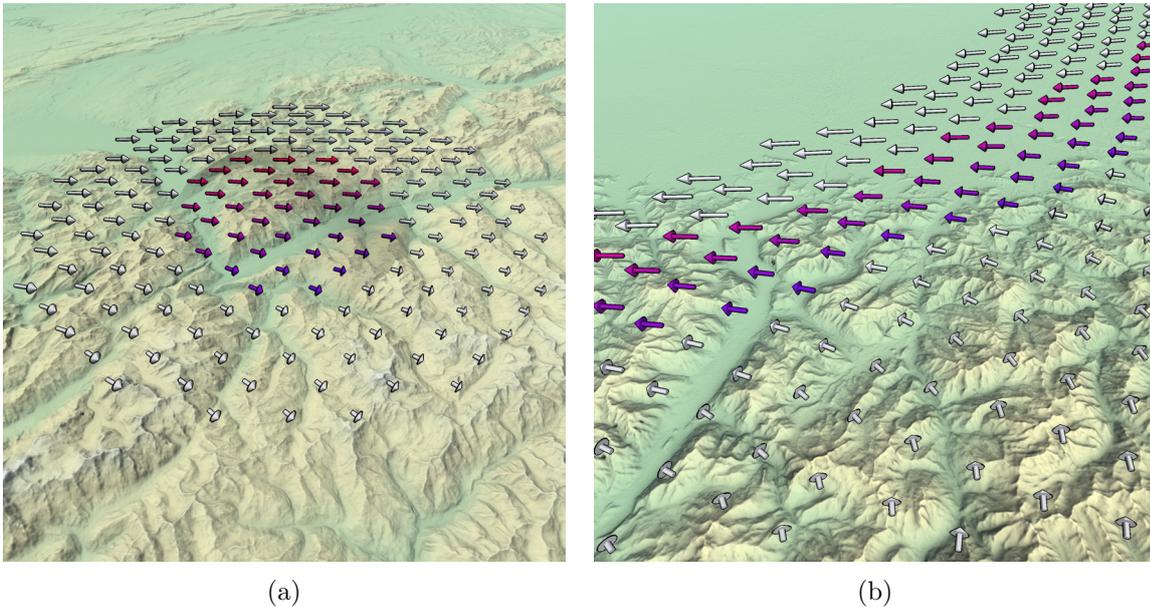


Figure 5.10: The intermediate context (white arrow glyphs) can be computed, using (a) spatial relationships or (b) data relationships.

in order to provide a better overview. On top of that, we can visually simplify the glyph. For instance, instead of encoding wind direction and wind speed, a simplified glyph would only encode directions through its orientation. Figure 5.11b illustrates both strategies. The reduced number of wind glyphs and the absence of encodings for wind strength relieves the visual complexity and de-emphasize of the intermediate context's region.

**Highlighting** Highlighting is utilized to accentuate information in the focus in order to guide the user's attention (Robinson (2011)). This can be achieved by emphasizing the focus or by de-emphasizing the context. We apply to this end two highlighting methods: adjustment of color and blur. To keep the visual presentation in the focus consistent, we show the focus always full-saturated, bright and crisply, but reduce either for the context.

The colors are adjusted by decreasing brightness or saturation. In case of the terrain, we reduce the brightness by reducing the amount of light that is shed on the surface. To reduce the prominence of data glyphs, we desaturated the color of the arrow. However, this highlighting method has to be applied carefully, so it does not interfere with visually encoded information. Figure 5.12a illustrates our approach. While the terrain is shown in unaltered colors for the focus and the intermediate context but with reduced brightness in the remaining context, the data are shown in unaltered colors only in the focus. The brightness of the data glyphs in the intermediate context is reduced and the remaining context shows no data at all. This design emphasizes the selected, relevant data and de-emphasizes the rest.

The second approach is the usage of blur. Through this, we can simulate the focal area of a real-world camera and thus depict areas in focus crisply and blur the regions outside. This strategy is similar to depth-of-field described in Section 4.2.4.

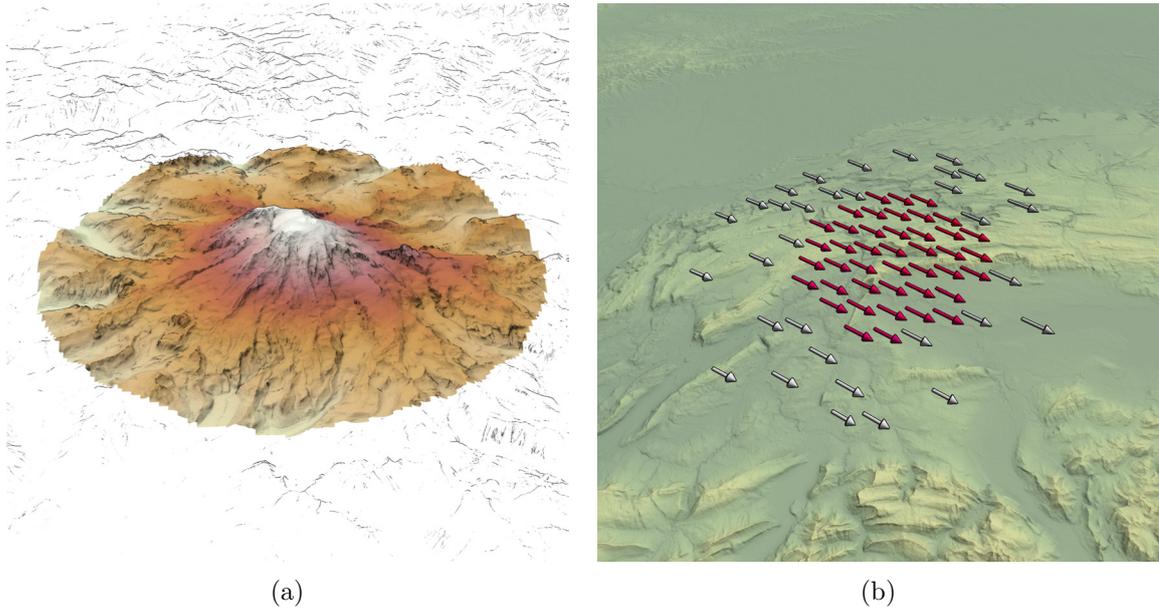


Figure 5.11: Adjustment of the representation using abstraction for (a) the terrain depiction and (b) the data depiction.

Figure 5.12b shows the data and terrain in the focus sharply, which automatically attracts attention, while the context region is blurred with increasing distance from the focus. This method must be applied with caution, since the size of the focus and the strength of the blur have a large impact of the interpretability of the image.

**Distortion** In many cases, the focus region constitutes only a small part of the presentation space. By using a distortion of the presentation space, we aim to enlarge the space for the focus area, while decreasing the space of the context area. In our case, data and terrain are connected via fixed spatial relationships. Therefore, distortion always affects the representation of data and terrain simultaneously.

Figure 5.13a shows the undistorted presentation space, while in 5.13b, the space of the focus area is enlarged and accordingly the context region is compressed to make appropriate room. We apply this concept with caution. To facilitate the interpretation of size and distances, only well-defined scaling steps are possible. More precisely, we use a piece-wise linear scaling in power of two. Moreover, to visually communicate where the boundaries of the enlarged and the compressed regions are located, we depict the borders either through lines (e.g., Figure 5.13) or through halos (e.g., Figure 5.12a).

#### 5.2.4 Discussion

Up to this point, we exemplified our concept by visualizations of wind data above terrain. However, we can apply our concept to other spatial data as well. Regarding terrain visualization in flight scenario, a typical example is the visualization of Points of Interest (POI), together with their labels. Examples of such POIs would be cities, airports, or mountains.

The amount of POIs in a terrain visualization can be very large, making their identification difficult. Showing additional labels aggravates this problem. As depicted in Fig-

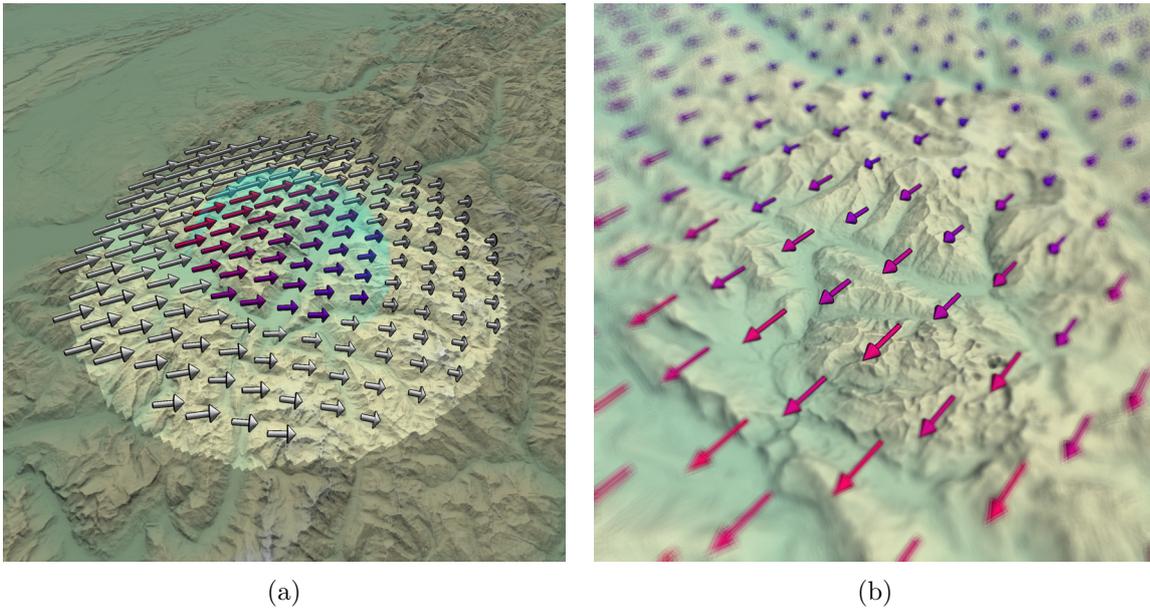


Figure 5.12: Adjustment of the representation using abstraction. (a) The focus region is shown by full saturated and bright colors for the focus, while the context is desaturated and darkened. (b) The focus region (center of the image) is rendered crisply, while the context region is blurred with decreasing distance.

Figure 5.14a, occlusion and clutter easily occurs, making the identification of labels nearly impossible. For one thing, we addressed this issue with an approach of occlusion-free, particle-based 3D labeling that has been filed as a patent application together with our partners from the aviation domain (Richter et al. (2019)). For another thing, we aim to reduce the amount of information by applying our Focus & Context concept.

In our particular case, we picked a specific data focus based on the prominence of the POI. For an exemplary data attribute that represents the prominence, we utilized population for cities and height for mountains. However, the relevance of POIs is also a subject to their distance to the viewer. Hence, we extended the focus selection strategy by this additional criteria. We generalized this strategy further by introducing a Degree of Interest (DOI) function that takes different criteria with different weights into account. The DOI function calculates the individual importance of a data point with respect to its distance to the viewer and other criteria regarding its relevance, such as population or size.

After the focus selection, the intermediate context is computed by collecting those POIs that differ from our DOI by a certain threshold. POIs within focus are depicted by markers (red circles for cities and black icons for mountains) together with their label (name). POIs within the intermediate data context are depicted only by their marker and those outside of the intermediate context are omitted. The result is shown in Figure 5.14b.

In conclusion, Focus & Context allows us to scale our visualization based on the task at hand. We demonstrated our approach by example of terrain-embedded wind glyphs and labels for POIs. Our three-step procedure covers the focus selection and derivation, the context computation and the adjustment of the representation. We provide various options for setting up the focus and accordingly determined the context. In order to emphasize the relevant parts of the visualization and de-accentuate the context, we scale the respective

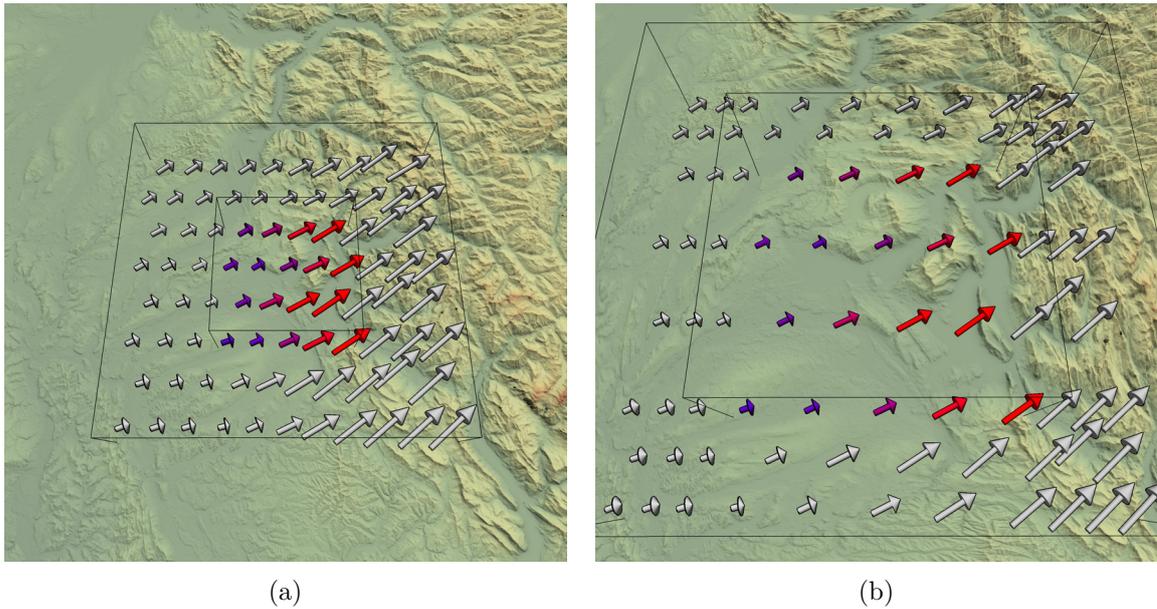


Figure 5.13: Adjustment of the representation using distortion. (a) Normal view on focus and context area and (b) the same view after magnifying the focus area.

Table 5.2: Options for parameterizing Focus & Context in 3D terrain.

Focus Selection	Context Computation	Adjustment
Data Focus/Spatial Focus, Predefined/Interactive, 0D/1D/2D/3D, Circular/Square Shaped Value/Interval Position Size	Data/Spatial Relationship Distance/Similarity Circular/Square Shaped	Data/Presentation Abstraction Coloring/Blur Distortion Halo/Grid/No Boundary

data representation using either different levels of abstractions, highlighting, or distortion. For this purpose, we utilized different visualization strategies from [Chapter 4](#), but also make use of novel approaches, such as magnification. Each step of focus selection, context computation, and adjustment provides a large range of alternatives to customize the visualization to the given scenario. [Table 5.2](#) summarizes once more the principal design options.

However, all these different options at hand also complicates the design process. Thus, the question how to parametrize the focus or how large to set the context can be difficult to decide and must be tailored to the current task. Similarly, the eventual emphasizing of the data representation in the focus area is enabled by various alternatives that can produce quite different results. Thus, in order to identify feasible designs we also need options to evaluate them. We will discuss this issue in the following section.

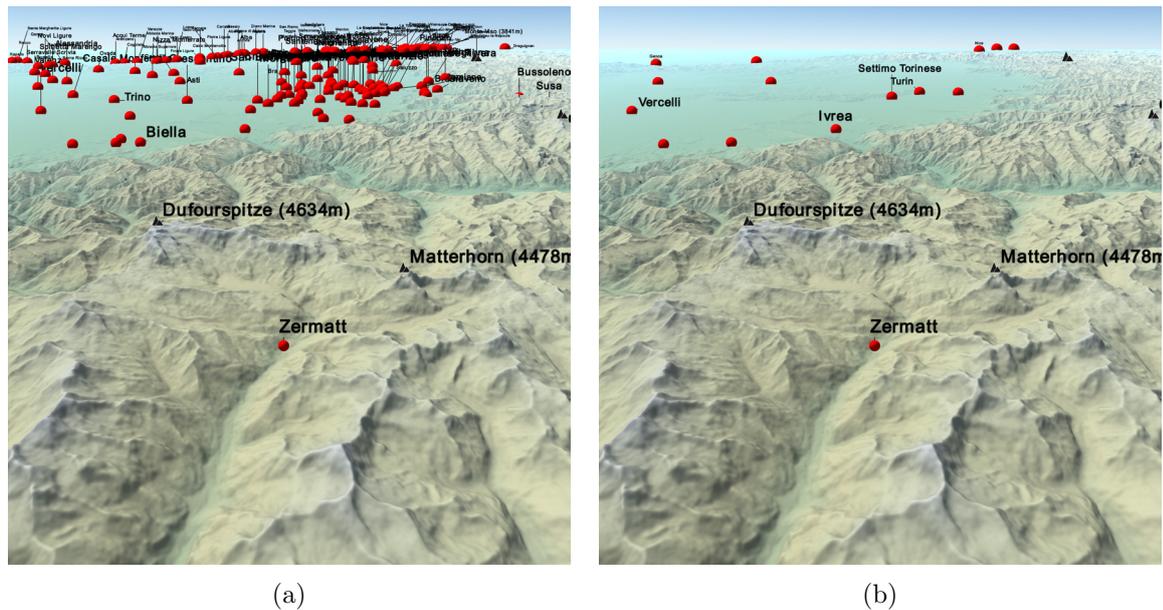


Figure 5.14: Strategies for reducing labels and POIs, using Focus & Context. (a) Showing all the available data for cities and mountains clutters the image. (b) Using Focus & Context via a DOI function allows to show only the most important POIs and thus un-clutters the image.

### 5.3 Evaluation using Saliency Information

The key idea of our concepts Prioritization and Focus & Context is to determine and to emphasize the relevant parts of the data. What is relevant is defined with respect to the given flight scenario. Emphasis, in turn, is realized by using different abstraction levels for visualizing the data. For this purpose, many different presentation methods exist and accordingly the design space is extremely large. Therefore, finding appropriate visual designs is intricate. Moreover, even after the decision for one presentation method it is typically difficult to determine whether the design really focuses on communicating the relevant data. For this reason, we need means to evaluate our visualizations.

One approach to engage this issue is to analyze the visual design with respect to how a presentation is perceived. Today, models are available that can automatically and reliably predict how an image would be perceived by considering the human perceptual system. While in some areas (e.g., for designing Graphical User Interfaces (GUIs) (Rosenholtz et al. (2011))) first approaches in this direction exist, such attempts for data visualizations are still rare. From there, this section is dedicated to the development of such a computational analysis approach in our context of spatial data visualization in 3D terrain.

#### 5.3.1 Motivation

In order to assess whether a visualization clearly communicates the relevant data, we need to evaluate the presentation towards two aspects, in particular. On the one hand, we need to know whether the data presentation is lucid, meaning that visual clutter does not prevent the perception of the data. Visual clutter, in this context, is the observed information that degrades the user's completion of a certain task. Hence, visual clutter is closely related to

the complexity of an image as it is more difficult for the human cognitive system to segment an unorganized scene. In order to measure visual clutter, we need methods that are capable of conveying which image areas cause clutter and how large its effect is.

On the other hand, we are interested in whether the visualization really emphasizes the relevant data. One way to analyze this is to examine which parts of the presentation are salient, meaning that they attract attention. It can be assumed that the parts that are likely to be looked at particularly facilitate the communication of information. Accordingly, these parts should match with the semantic relevance of the data. In order to detect the areas of a visualization that attract attention, we need methods that are able to measure the saliency within an image.

In the scientific field of cognitive science, strategies to measure visual clutter and saliency already exist, though they have primarily been developed for and applied to 2D static images, leaving interactive 3D data visualizations aside. Therefore, our aim is to apply them in this context to make a first step towards an qualitative evaluation of 3D visual designs. To this end, we will use a measure for visual clutter and saliency respectively in order to assess whether a visualization overloads the visual budget and whether the salient regions of the presentation communicate the relevant information.

In the following, we will briefly summarize related work in order to provide the backgrounds for our research and describe our concrete realization.

### 5.3.2 Visual Clutter

Rosenholtz et al. (2007) described visual clutter as the state in which the representation or organization of entities leads to a degrading performance of some task. They summarized clutter as excess or disordered display that causes crowding, reduced object recognition, or decreased visual search performance. In literature several approaches exist that measure visual clutter, such as Woodruff et al. (1998), Frank and Timpf (1994), Mack and Oliva (2004), Rosenholtz et al. (2005), Rosenholtz et al. (2007), and Oliva et al. (2004). In the following, we will focus on the following three approaches: Feature Congestion Measure, Subband Entropy Measure, and Edge Density Measure that have shown good results in previous evaluations.

The *Feature Congestion Measure*, described by Rosenholtz et al. (2005), makes extensive use of the Statistical Saliency Model (Rosenholtz (1999)). The model itself is intended to detect prominent features. Features hereby refers in particular to the primary perceptual phenomena color, contrast, and orientation (Wolfe (1998)). Accordingly, the Statistical Saliency Model calculates a score for the degree to which a certain feature vector  $F$  is an outlier to the local distribution of other feature vectors. To this end, the local mean  $\mu$  and the covariance  $\Sigma$  are used to determine the local saliency  $S$  by the following equation:

$$S = \sqrt{(F - \mu)^T \Sigma^{-1} (F - \mu)}$$

For one thing, this model states which features in the image are prominent and how large the difference to the surrounding features is. For another thing, the model also indicates the difficulty to add new entities with prominent features if the local area is already congested. This is the case, if the covariance  $\Sigma$  of a feature vector  $F$  is large in relation to the possible features. This observation is the main idea for measuring visual clutter by feature congestion. In sum, the measure detects visual clutter where the average saliency is relatively low. The drawback of this method is the requirement of a target feature to

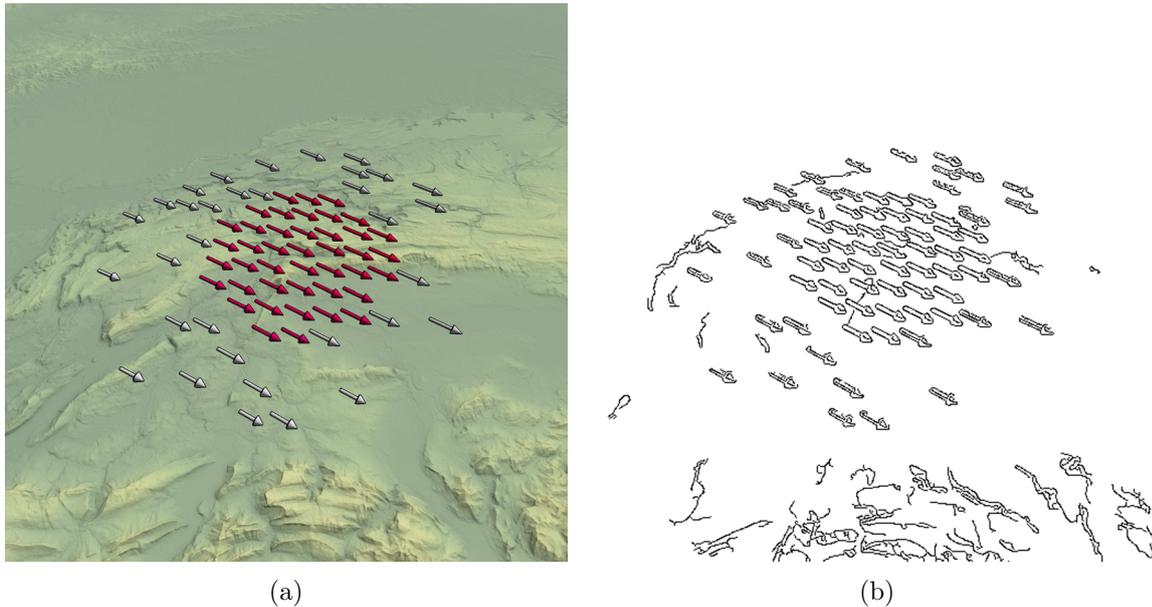


Figure 5.15: Application of the Canny Edge Detection on the visualization from Figure 5.11b. (a) is the original image and (b) shows the detected edge.

narrow down the search of clutter. Otherwise, the measure amounts to the inverse of the average saliency per unit image area, which would give misleading results.

The *Subband Entropy Measure* pursues a different approach, where visual clutter is described as the inverse proportional to the grade of organization in an image. Organization hereby refers to the grouping of similar entities of the image together and its degree can be seen as the predictability of a scene's content. Treisman and Gelade (1980) have shown, that the organization directly affects visual search performance. In order to compute the degree of organization, Rosenholtz et al. (2007) suggest to determine the encoding efficiency by current subband image coding methods, such as wavelet image coding (e.g., JPEG2000). To determine the entropy of the encoded image, the Shannon entropy  $H$  is used as follow:

$$H = - \sum_i p_i \log_2(p_i)$$

Here,  $p$  refers to the probability distribution of coefficients in each subband  $i$ . The entropy  $H$  can now be used to determine the degree of predictability, or in other terms, the degree of redundancy of an image. Accordingly, the entropy would be directly proportional to the amount of visual clutter.

The underlying idea of the *Edge Density Measure* is the fact that receptive fields of the human cognitive system naturally extract edges from the visual scene. Accordingly, it can be supposed that many edges within a scene cause a certain stress on the cognitive system and hence trigger visual clutter. Therefore, this measure computes the density of the edges. Mack and Oliva (2004) used in their work a Canny edge detector to measure the density of edge pixels. Low and high thresholds determine how weak or strong the edges found are and weak edges are only kept if they are connected to strong ones. According to the authors, the computed density is directly proportional to the visual clutter in the image.

Rosenholtz et al. (2007) evaluated and compared the three measures Feature Congestion, Subband Entropy, and Edge Density. They stated that all these measures perform well and

that with respect to the predicted reaction time of users no significant differences were found. However, Feature Congestion performed better at predicting mean contrast thresholds and was also more sensitive to visual clutter caused by increased color variability. However, this algorithm uses component analysis, which is complex to compute. In particular, the complexity for computation is  $O(n^3)$ . The Subband Entropy Measure is based on wavelets, which computation complexity is  $O(n \log n)$ . By parallelization and further optimization, real-time computation might still be possible. With respect to our goal, an automatic, rapid analysis of the visualization Edge Density Measure is particularly suited. Computing the edges, for example, through the Canny edge detection, is a simple, well known, and fast algorithm with a complexity similar to Subband Entropy's complexity, but using a more straightforward and GPU-friendly implementation. Thus, we decided to base our visual clutter measure on Edge Detection.

Rosenholtz et al. (2007) showed in their evaluation that the Edge Density Measure performed reasonable, though for images with a high color variability this measure was not as good as the other two. This is because the Canny edge detector does not account for color at all but works on intensity only. We try to account for this fact by applying the edge detection on all three color channels separately and subsequently combining the results by looking for the maximum in either channel.

Figure 5.15 shows exemplarily the application of the Canny edge algorithm on a visualization. Areas with a higher edge density represent regions that tend to clutter. Based on the extracted edges, we can also compute the average edge density and thus provide an overall value that represents the degree of visual clutter in the visualization. We normalize this value between 0.0 and 1.0 and call it visual clutter degree, though strictly speaking this value states a ratio, namely between the detected edges and the available image space. We can utilize this value to point out visual clutter, for instance when evaluating dynamic scenes, which we will describe later on.

### 5.3.3 Saliency

Goal of a saliency measure is to generate a map that marks the conspicuous features of an image. In literature, this map is called *saliency map*. A saliency map depicts for each location the probability of attracting attention, a so-called visual fixations. Such a probability can be estimated by empiric studies that record the visual fixations of multiple test persons for a given image. Another approach is to find a computable function that predicts these visual fixation.

But, there is a large amount of factors that contribute to the saliency of features and extensive research has led to various models focusing on different factors. A comprehensive overview of this research is given by Borji and Itti (2013).

**Backgrounds** In general, saliency measures can be categorized in two ways—based on their biologic plausibility or their mathematical concept. Biologic plausibility refers to aspects of the human visual system that the models incorporate. Hereby, we can differentiate between top-down and bottom-up manner. Expectation, knowledge, and goals are considered to influence the visual search process in a top-down manner. For example, a search for a certain shape in a scene would trigger different visual fixation points than a search for a certain color. According to Itti and Koch (2000), top-down attention is generally slow and task-driven.

Bottom-up attention is guided by the characteristics of the scene. For instance, a red sign in an image is likely to attract attention. According to Egeth and Yantis (1997), bottom-up attention is relatively fast. Examples of models that primarily utilize top-down attention cues are Mccallum (1996), Paletta et al. (2005), and Verma and McOwan (2009). Approaches that mostly utilize bottom-up attention cues are Itti et al. (1998), Zhang et al. (2008), and Li et al. (2010).

The mathematical categorization was proposed by Borji and Itti (2013) and differentiate between cognitive models (e.g., *Itti-CIO2* (Itti et al. (1998))), bayesian models (e.g., Itti and Baldi (2009)), decision-theoretic models (e.g., Gao and Vasconcelos (2005)), information-theoretic models (e.g., *AIM* (Bruce and Tsotsos (2005))), graphical models (e.g., *GBVS* (Harel et al. (2006))), spectral analysis models (e.g., Hou and Zhang (2007)), and pattern classification models (e.g., Judd et al. (2009)). In general, many models can be categorized in multiple mathematical categories and a few do not match any of them.

There exist many approaches to evaluate and compare the different saliency models. For instance, Borji et al. (2013) tested 35 different models regarding their quality and time complexity, using various data sets and different types of scores. Depending on the data set and score, they determined different models working particular well. Generally speaking, *AWS*<sup>1</sup> (Garcia-Diaz et al. (2012)), *GBVS*, *AIM*, and *Itti-CIO2* performed well, though there is still a certain gap between current models and real human performance. The fastest models for a small data set (511 × 681 px image) were *HouCVPR* (Hou and Zhang (2007)) with 300 ms, *VOCUS* (Frintrop (2006)) with 25 ms, and *Itti-CIO* (Itti and Koch (2000)) with 17 ms. Furthermore, there exist GPU-based parallel implementations of certain models that also perform particularly fast, such as Han and Zhou (2007), who based their work primarily on *Itti-CIO2*.

Saliency models have typically applied to static 2D images. Hence, Lang et al. (2012) tested the influence of 3D to the quality of saliency maps, but found no significant changes between 2D and 3D. Moreover, saliency maps are typically developed for static scenes, only some incorporate a motion channel. Borji et al. (2013) also tested saliency models against dynamic scenes (i.e., video data), but did not adjudge models with motion channels better than those without. On the contrary, the best performing saliency maps for static scenes were also the best performing models for dynamic scenes.

These findings indicate that saliency maps are also applicable in our context, that is, data visualization using 3D dynamic scenes. Demands on such a saliency map are, in particular, quality with respect to the prediction of visual fixations and speed with respect to computation time. Regarding quality, we already mentioned *AWS*, *GBVS*, *AIM*, and *Itti-CIO* that performed particularly well. There are also 12 further models that performed nearly as good, including *Itti-CIO2*. Thus, we have a wide range to choose from. Concerning the categorization with respect to biological factors, bottom-up cues are much faster, involuntary and more universal than top-down cues. Accordingly, models that incorporate bottom-up cues will be preferable. Concerning the mathematical categorization, we prefer cognitive models, since they are inspired by cognitive concepts and findings from the scientific fields of psychology and neurophysiology.

Regarding speed, we particularly focus on models that are capable for interactive analysis. Through this, we facilitate the evaluation of presentations during the design process and thus allow an immediate adaption. The currently fastest implemented model on CPU is *Itti-CIO*, followed by *VOCUS* and *HouCVPR*. The fastest GPU implemented model is

---

<sup>1</sup>Abbreviations for the saliency models have been adopted from Borji et al. (2013).

*Itti-CIO2*. Due to this fact and because *Itti-CIO2* performed with an acceptable quality, we choose this model for our evaluation.

**Computation** The computational model of *Itti-CIO2* consists of an image decomposition into multiple *conspicuity maps* and a subsequent weighted fusion of these maps into the final saliency map. Features of the conspicuity maps are the individual color channels  $C$ , the intensity  $I$ , and the orientations  $O$ . Accordingly, the saliency map  $S$  is computed as follows:

$$S = w(I)I + w(C)C + w(O)O$$

To better approximate the human perception, the color channels and the intensity are derived from the CIE-LAB color space, which consists of a lightness component ( $L$ ) and two color differences ( $A = \text{red} - \text{green}$  and  $B = \text{blue} - \text{yellow}$ ). With regard to orientation, we consider  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , and  $135^\circ$ .

To compute the conspicuity maps, we first extract the features from the image. Since receptive fields of the human are sensitive in their center, whereas the antagonistic surround inhibits their response, the cognitive system is particular responsive for local discontinuities. This fact is modeled by computing differences of the features between fine and coarse scale versions of the image into so-called *feature maps*  $M$  using across-scale addition.

To derive the conspicuity maps from the feature maps, we need to consider the respective degree of uniqueness, which represents the different contributions to the final map. Since being salient means that an entity is very different to its surroundings, the contribution of feature maps to the saliency map is inverse proportional to the amount of available features. Accordingly, a feature map with only a few strong features contributes stronger to the saliency map than the feature maps that have many features. To model this inverse proportionality [Han and Zhou \(2007\)](#) utilize the aforementioned Shannon entropy to calculate the respective weights for each feature map:

$$H(M) = -\log_2(p(M))$$

where  $p(M)$  is the probability of a feature within the map. We use the uniqueness and additionally the amplitude given by the global maximum to weight each feature map as follows:

$$\bar{w}(M) = H(M) \max_{v(i,j) \in M} |v(i,j)|$$

Using  $\bar{w}$  we can generate the conspicuity maps as follows:

$$\begin{aligned} I &= \sum_{\sigma} \bar{w}(M_{I_{\sigma}}) |M_{I_{\sigma}}| \\ C &= \sum_{\tau} \sum_{\sigma} \bar{w}(M_{\tau_{\sigma}}) |M_{\tau_{\sigma}}| \\ O &= \sum_v \bar{w}(M_{O_v}) \times |M_{O_v}| \end{aligned}$$

where  $\sigma \in \{+, -\}$ ,  $\tau \in \{A, B\}$ , and  $v \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ . To obtain the weights of the final saliency maps, we again utilize the Shannon entropy to take account for the individual uniqueness of each conspicuity map, though this time without considering their amplitude:

$$w(M) = H(M)$$

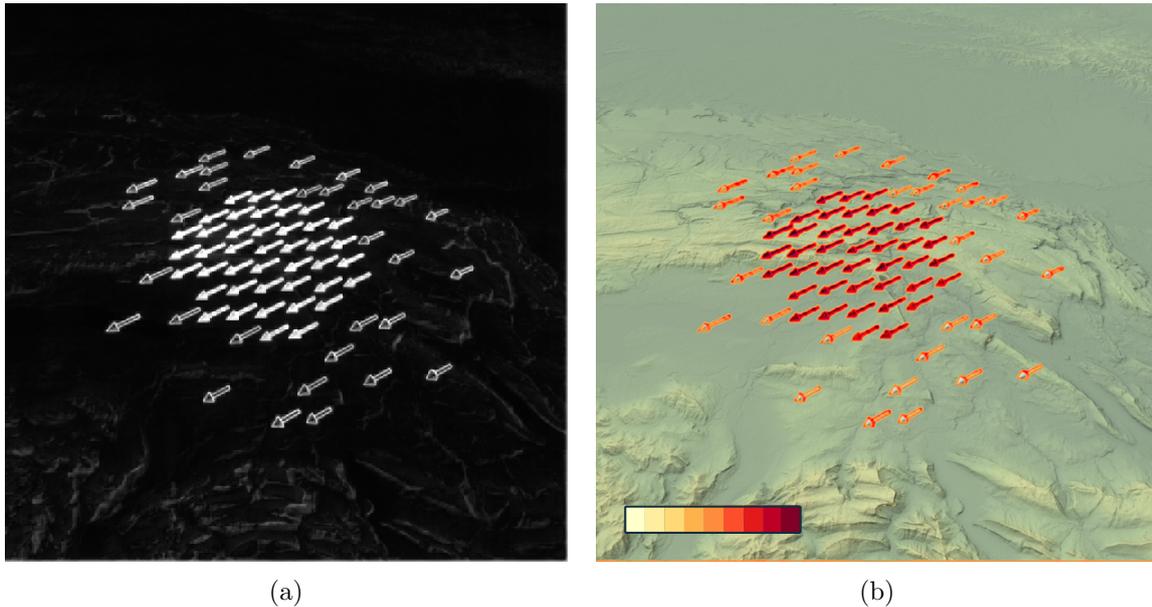


Figure 5.16: Application of the Saliency Map on the visualization from Figure 5.13a. (a) shows the computed saliency map and (b) the corresponding heat-map, blended with the original visualization.

An exemplary saliency map is depicted in Figure 5.16. The result can be displayed as an intensity image, where bright regions mark salient areas. Such a saliency map is shown in Figure 5.16a. Alternatively, a heat map can be computed from the saliency map and blended with the visualization to communicate which entities in the image are salient (Figure 5.16b).

The full computation of the saliency map can be done in parallel on the GPU, which facilitates results in real-time. Together with our visual clutter measure, we can now evaluate our visual designs as we will describe in the following section.

### 5.3.4 Evaluation of Visualizations

Using the visual clutter measure and the saliency map, we can now verify two things: how cluttered is the current image and where are the regions within the visualization that attract attention. This information can aid the design of presentations that show the relevant data prominently. To this effect, the corresponding saliency map would show high values for places where the prioritized or focused data are located and the respective visual clutter measure is low. With this in mind, we will examine in the following our proposed visual designs derived from our concepts Prioritization and Focus & Context.

**Prioritization** In Figure 5.17, we show our three exemplarily proposed visualizations from Section 5.1 and the corresponding saliency map.

The *prioritization of terrain* is shown in Figure 5.17a. The respective saliency map in Figure 5.17b indicates for one thing very high values for the contours, but also high values for the shaded parts of the surface. Accordingly, the details of the terrain are attracting attention in this kind of visualization. Regions that show data values are also marked

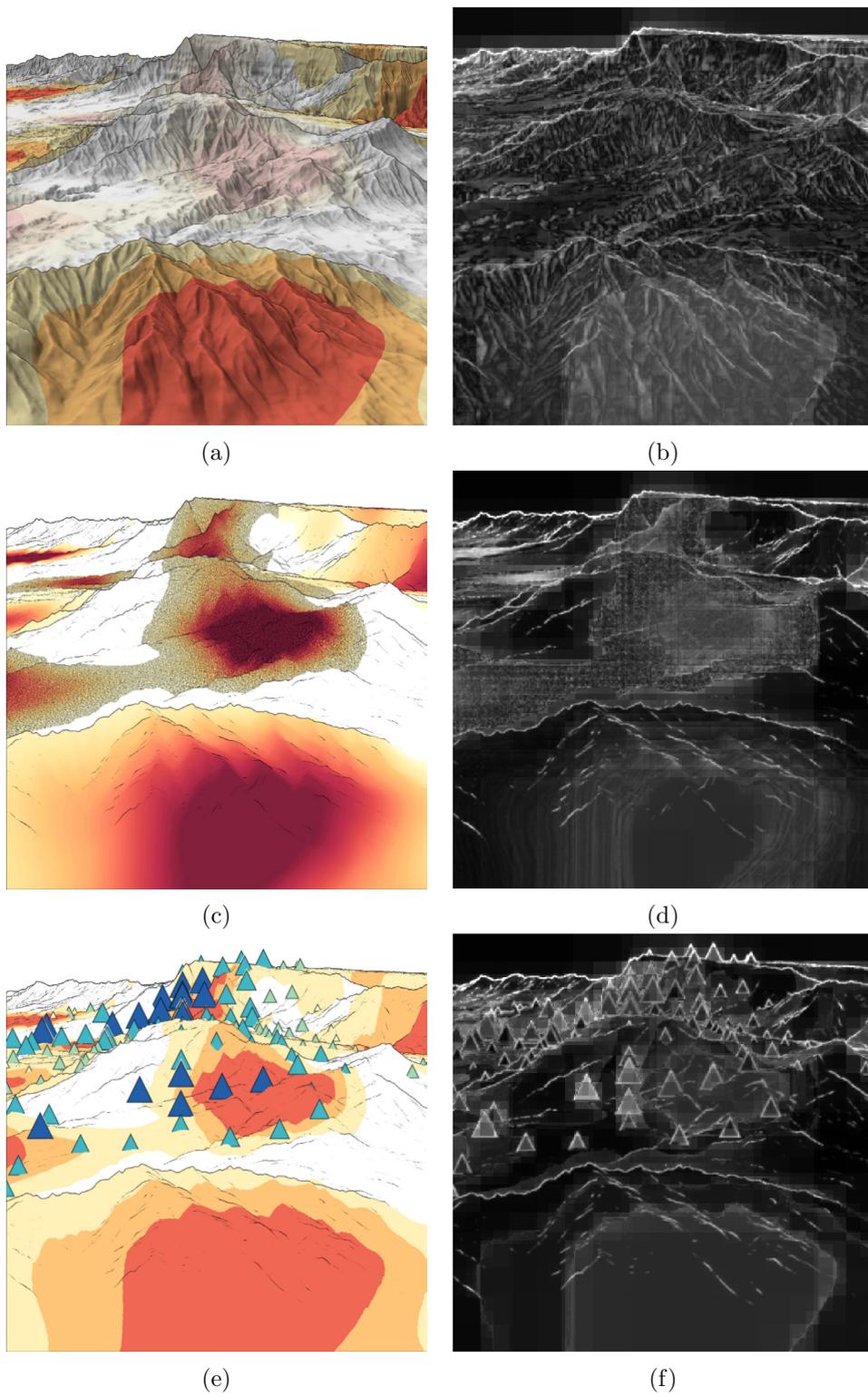


Figure 5.17: Saliency Maps (right column) for each visualization (left column) described in Section 5.1. (a), (b) terrain first, (c), (d) data first, and (e), (f) uncertainty first.

prominent, if uncertainty is low. Otherwise, regions that communicate uncertain data are not identifiable in the saliency map. This is consistent to our goal to let uncertain regions attract less attention than certain ones. The degree of visual clutter in this exemplary visualization was 0.25, which indicates moderate clutter, mainly due to the shaded surface of the terrain.

Our next option, *prioritization of data*, is shown in Figure 5.17c and 5.17d. In contrast to the previous example, the terrain is now represented prominently only in the contours, whereas the surface (without data) does not attract attention. But the saliency map also shows that regions that should communicate the data attract only average attention for yellow areas and minor to no attention for dark red areas. There are multiple reasons for this.

First, the model for our saliency particularly searches for discontinuities in the image. But, our design was primarily intended to communicate detailed data values using a continuous color scale and prominence was not our primary goal for data first. The continuous transition, however, contradicts the operating principle of the model.

Second, the saliency map indicates that the colors used to encode the data, in particular the dark red colors, do not induce visual fixations. Hence, a change of the color scale might reduce this issue.

Third, the model determines salient regions based on the number of present features in nearer surroundings (cf. Statistic Saliency Model in Section 5.3.2). Accordingly, large uniform areas naturally induce less visual fixations. This also correlates with the Shannon theorem that would state a higher predictability in this part of the image.

On the other hand, the uncertainty representation in this example increases the prominence of the respective areas. This is due to the noise and thus to the reduced predictability of these image parts. This is an interesting observation, which can be useful when deciding whether uncertain data should be depicted prominently or not.

The visual clutter measure indicates for this visualization a lower degree of clutter (0.18) than the previous example, probably due to the decreased detail of the terrain surface.

The strategy *prioritization of uncertainty* is finally illustrated in Figure 5.17e and 5.17f. Similar to the previous example, the terrain is shown less prominently. However, the glyphs that represent uncertainty attract much attention, which is consistent with our intention to primarily communicate uncertainty. Interestingly, the model predicts a high probability of visual fixations for solitary glyphs that are not surrounded by many other glyphs, which again correlates with the Shannon theorem. The data representation, on the other hand, is determined less prominent by the saliency map, which is plausible.

The degree of visual clutter in this example (0.18) is similarly low as the previous one. However, it is important to note that the measured values only indicate clutter on a global level. To identify local regions, one value is not sufficient. Instead, the value could be determined for sub-regions of the image or the result of the edge detection can be visually examined.

**Focus & Context** In order to evaluate our proposed strategies for Focus & Context, we again applied the visual clutter measure and the saliency map. In our concept, we pursued two strategies to emphasize information in focus. On data level, we reduced or omitted information outside the focus. On presentation level we utilized abstraction, highlighting, and distortion. With respect to the data level, the visual clutter measure obviously shows lower values if the number of depicted graphical primitives decreases. At the same time,

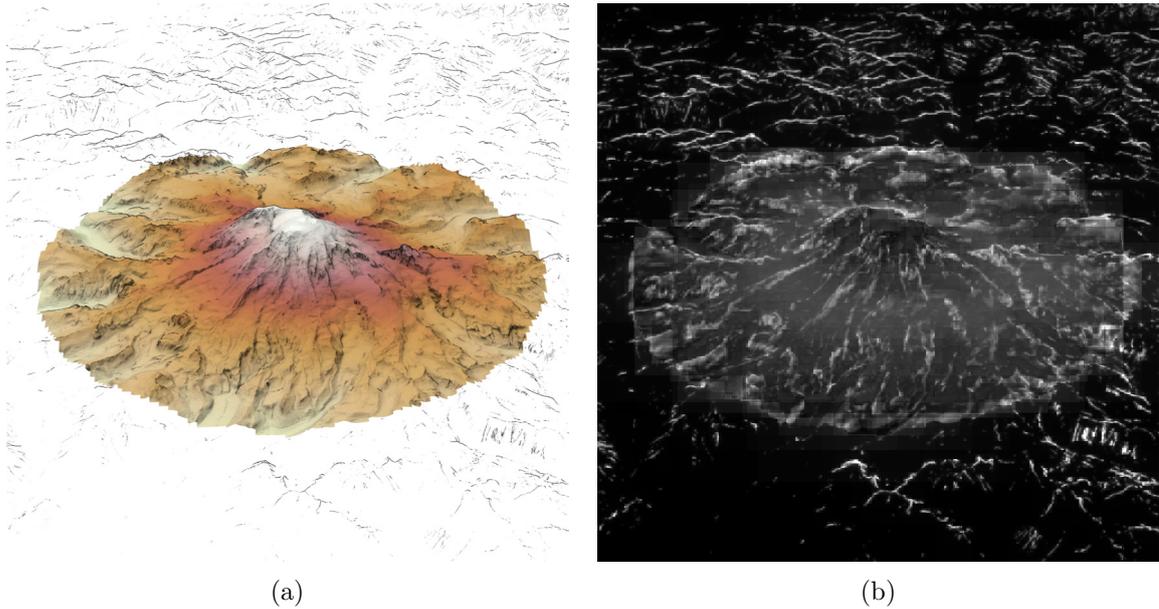


Figure 5.18: Saliency Map for the Focus & Context example using abstraction. (a) shows the original visualization and (b) the corresponding saliency map.

the maximum amplitude of the feature maps decreases, which results into higher values for available features in the saliency map.

With respect to the presentation level, we can assess our methods for abstraction, highlighting, and distortion. Accordingly, we applied the saliency computation to our exemplary visualizations from Section 5.2.

*Abstraction* of the terrain and the corresponding saliency map is exemplarily shown in Figure 5.18. Salient regions are clearly identifiable within the focus region, where the terrain is colored and shaded. The transitions between focus and context region are particularly prominent due to the discontinuity of the color. Moreover, details of the surface represented with shading result also into higher values in the saliency map. The same is true for the contours in the context region. Hence, the coarser structure of the terrain is still communicated. The degree of visual clutter in this visualization is measured with 0.25 in opposition to a value of 0.35 for a full shaded terrain surface, without Focus & Context techniques.

*Highlighting through color adjustment* is shown in Figure 5.19. With respect to the data, wind glyphs within the focus are clearly more salient than those in the context, particularly those exhibiting red color. Saliency decreases for glyphs exhibiting blue colors, which indicates that the color coding must be chosen carefully, if all glyphs in the focus must be highly prominent. For instance, using a color scale ranging from yellow to orange would increase the saliency in this particular case.

The saliency of the terrain in the focus and intermediate context though is hardly higher than in the context. There might be two reasons for this. Either the desaturation of the terrain in the context is not enough to increase the saliency in the focus or the saliency of the wind glyphs in the focus prevent further visual fixations onto the terrain.

Further tests revealed that the latter is the case. Due to the high feature density in the focus, the saliency of the terrain is decreased. Countermeasures would be a further

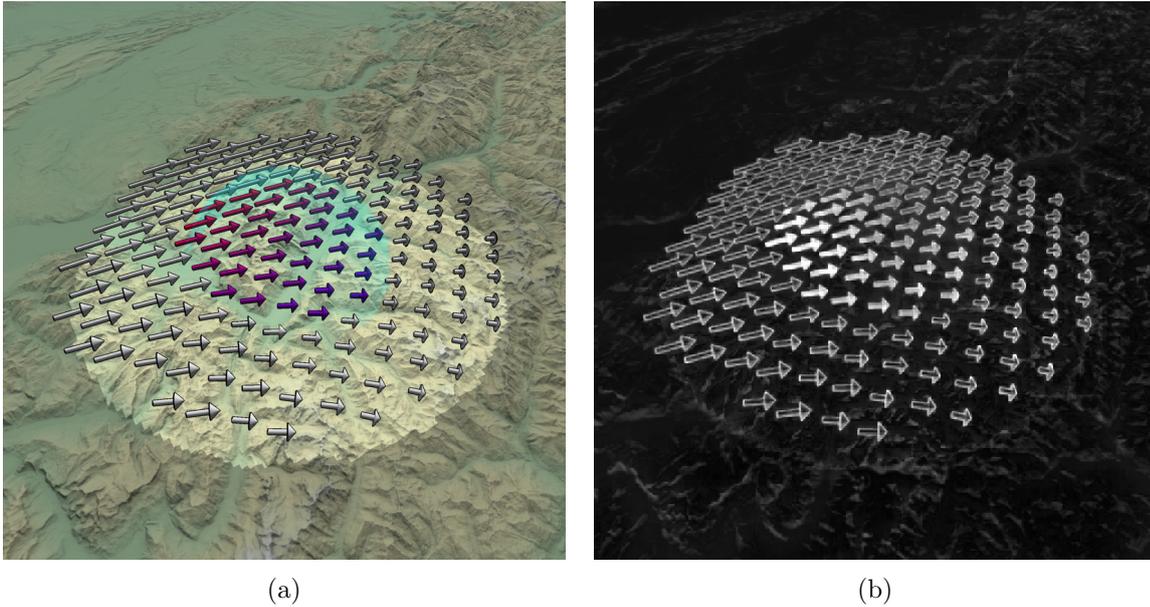


Figure 5.19: Saliency Map for the Focus & Context example using highlighting through color adjustment. (a) shows the original visualization and (b) the corresponding saliency map.

desaturation of the terrain within the context. Nevertheless, the transition between context and intermediate context is still clearly visible.

Regarding visual clutter, this exemplary visualization shows a relatively high value of 0.32, which indicates potential clutter, though the degree of clutter without Focus & Context (0.43) is still higher.

*Highlighting through blurring* was tested as a second example, which is shown in Figure 5.20. We anticipated that this technique certainly facilitates Focus & Context, but that the saliency map would not show much different. We thought so, because the changes in the image are only minor, considering the high amplitudes in color discontinuities. Interestingly, the saliency map clearly shows that regions in the focus (image center) are more salient than those in the context (image sides), regardless of the wind glyph's color. The same holds true for the terrain surface. Our visual clutter measure also shows a decrease from 0.34 to 0.18. Thus, it can be concluded that blurring can actually be a very effective Focus & Context technique.

Finally, the *distortion technique* was tested. The technique magnifies the focus area and scales down the context. Figure 5.21 shows the saliency maps for the original visualization (5.21a) and for the magnification (5.21b). The figures show that there is hardly any change in the saliency when applying distortion. In fact, a change in saliency is only expectable, if the dense of features in the focus area without distortion affects the prominence of the individual features. On the other hand, the map also shows that there is no decrease in prominence, if the wind glyphs are pulled apart. The visual clutter measure also indicates no significant decrease (0.30 decreased to 0.28) in clutter for this example. This is plausible, since the measure is based on global edge detection and does not consider local cluttered areas.

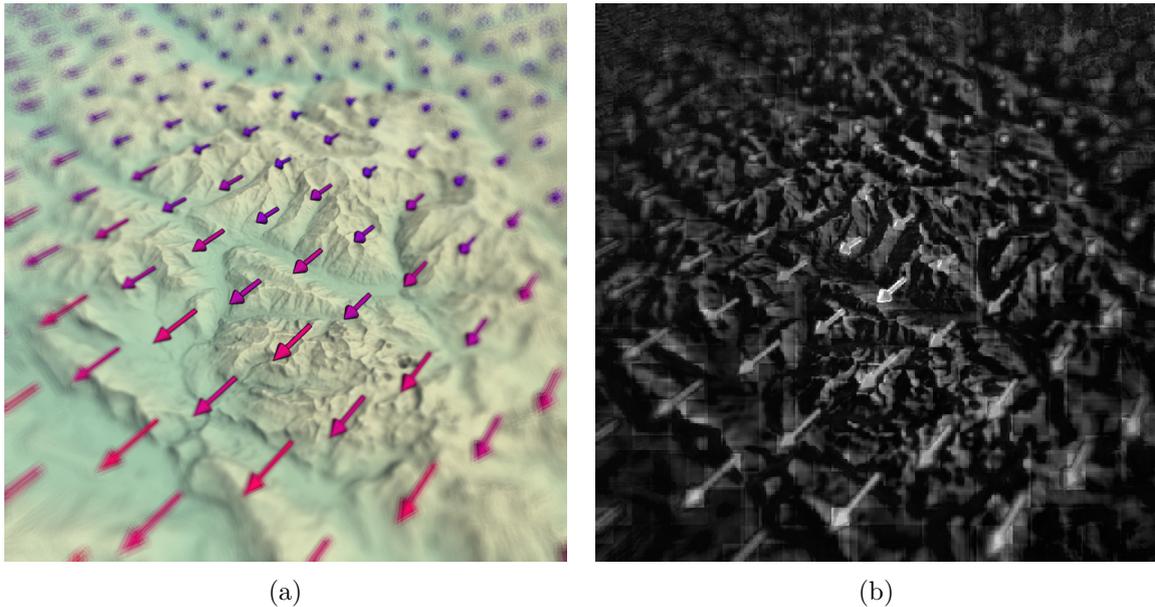


Figure 5.20: Saliency Map for the Focus & Context example using highlighting through blurring. (a) shows the original visualization and (b) the corresponding saliency map.

### 5.3.5 Discussion

In this section, we discussed ways to assess visualization with respect to a clear communication of relevant data. To this effect, we applied a measure for visual clutter and a perception-based model for generating a saliency map. Our selected visual clutter measure is based on edge detection, a simple but fast and satisfactorily performing method. For generating the saliency map we applied the *Itti-CIO2* model.

By using the saliency map, we can identify regions of the image that potentially attract attention. Ideally, these would be the same regions that communicate the relevant data. The perceptive model that produces the saliency map hereby normalizes the map with respect to the amplitude, meaning the strength of the features and with respect to the uniqueness of the features. Therefore, the resulting values represent a probability of visual fixations relatively to the whole image. This also means that high values in the saliency map do not necessarily equate with a high prominence in general. Accordingly, a feature within an image that otherwise exhibits only few features can have a high saliency, though the same feature in another image that shows numerous strong features would have a lower saliency.

Knowing this, it is obvious that evaluating visual designs of dynamic scenes cannot be done by examining single images. Instead, many possible or reasonable viewing angles must be considered. For this purpose, we facilitate two evaluation strategies. For one thing, we support the analysis of the visual design after a performed exploration of the scene or after carrying out a certain task. To this end, either the visual clutter values or the saliency maps can be examined. A possible procedure is illustrated in [Figure 5.22](#). Here, we exemplarily analyzed a landing approach through the terrain. The measured visual clutter values are presented as a line graph (bottom of the figure). The peaks of the line graph imply a sudden increase of visual clutter and thus indicate the need for further examination. Thus, the respective visualizations (middle of the figure) can be investigated. Moreover, we can

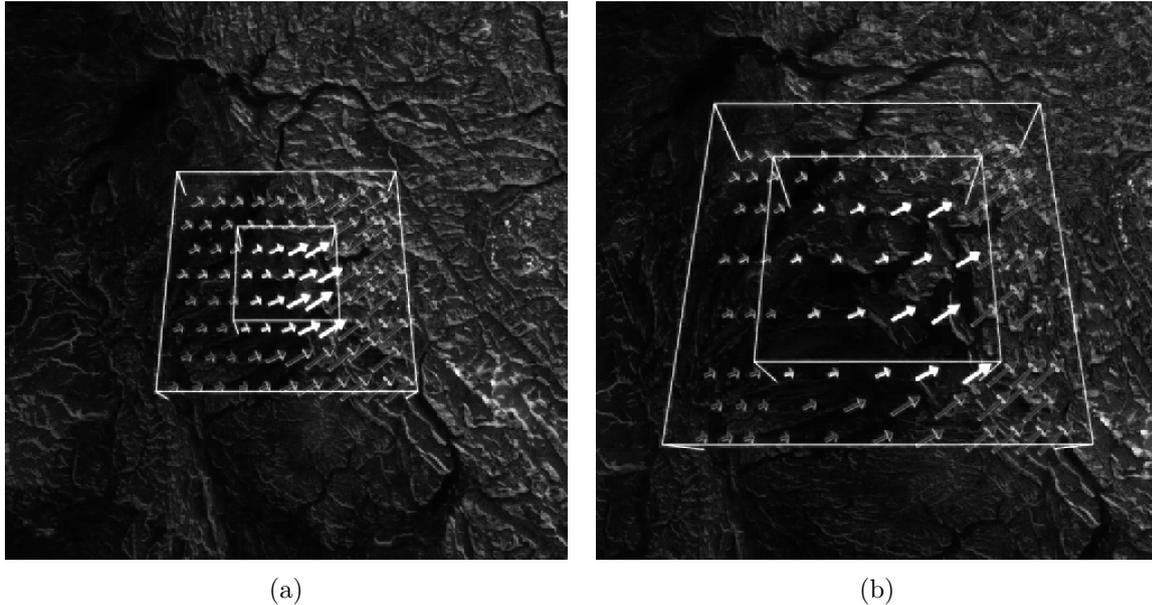


Figure 5.21: Saliency Map for the Focus & Context examples using distortion. (a) shows the saliency map of the visualization without and (c) with distortion.

look at the corresponding saliency maps to ensure that the clutter does not impede the communication of the relevant data. Accordingly, the visual design can be reconfigured and the test can be iteratively repeated.

The second evaluation strategy would be an on-the-fly analysis of the visual design. Since we support a real-time computation of both visual clutter and saliency, the appropriate maps can be looked at either side-by-side or blended with the visualization, for instance, as a heat-map (Figure 5.16). Through this, the design can be adjusted and the results can be evaluated instantly. However, to facilitate such rapid prototyping a sophisticated visualization framework is required. To this effect, we will propose such a tool, named TEDAVIS, in the following chapter.

## 5.4 Summary

In this chapter we proposed two fundamental concepts to task-dependently scale visual designs, that are, Prioritization and Focus & Context. The core idea of *Prioritization* is to concentrate on the communication of one particular data aspect and attenuate the others. We applied our concept to visualizations from aviation-related scenarios that incorporate multiple data aspects, in particular terrain, data, and uncertainty. We illustrated prioritized visual designs that can be useful during take-off and approach, during overflight, or for re-planning of flight routes. Though we have considered so far only terrain, data, and uncertainty, we are confident that the idea of Prioritization can also help us to integrate other information, such as streets or obstacles. Prioritized rendering strategies could be developed in analogy to the ones we described here.

Our second approach, *Focus & Context*, facilitates scalability of visualizations on a more local level. While Prioritization place emphasis on one whole data aspect, Focus & Context allows to interactively select subregions of the data and the terrain. We hereby extend the

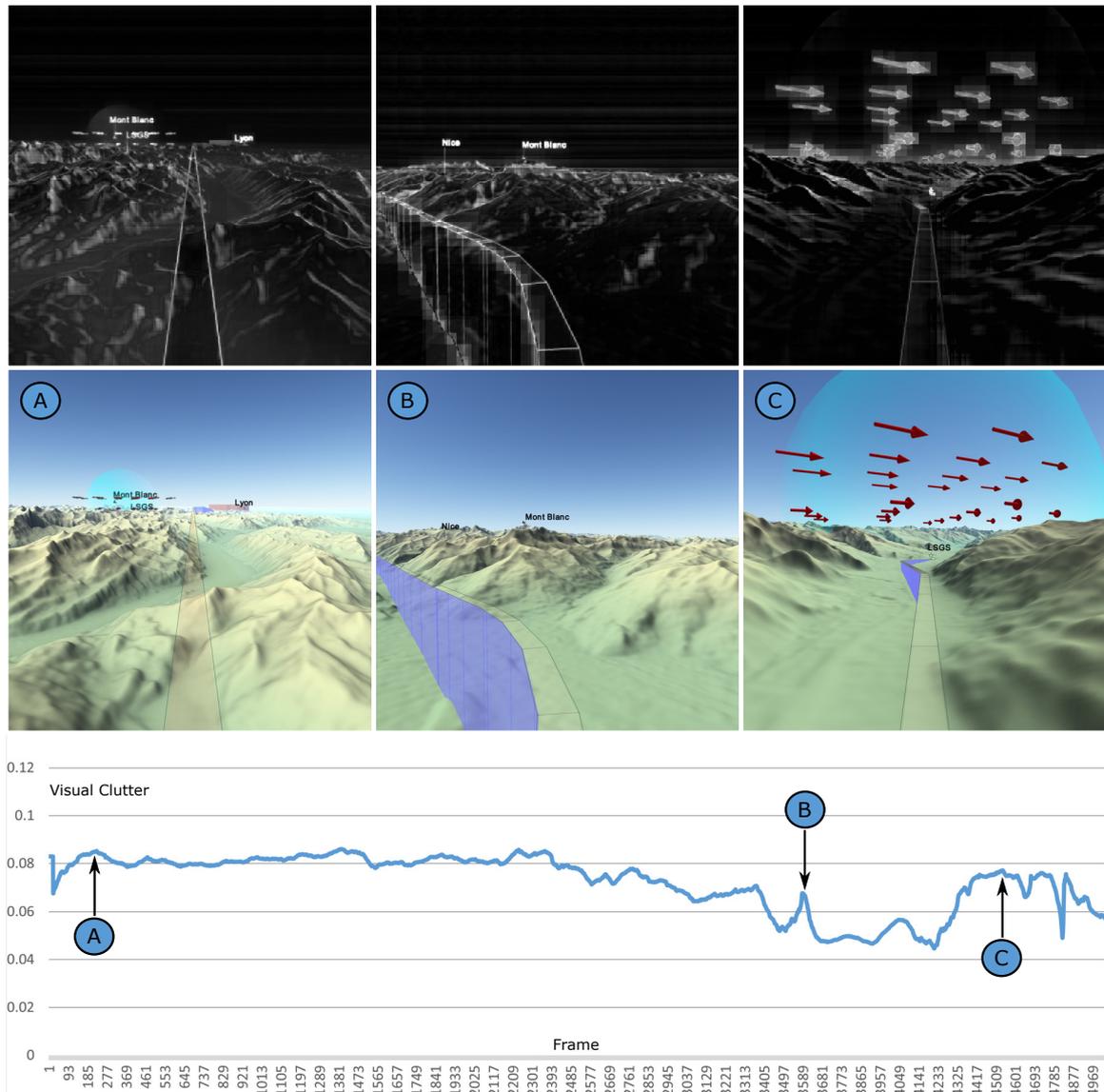


Figure 5.22: Visual clutter analysis during a simulated flight towards a destination airport. The line graph shows the degree of visual clutter during flight. Exemplary peaks (A, B, C) are shown in the respective screenshots together with their corresponding saliency map.

classical Focus & Context concept to our context, the visualization of spatial data in 3D terrain. Accordingly, we distinguish between a spatial focus where an interesting point or region in space is selected and a data focus where certain interesting data characteristics are selected. Through numerous visualization methods, we provide special emphasis on the focus, while reducing accentuation for the context.

Nevertheless, there are still open questions regarding the representation of the focus. Though the data focus specifies the most important information, it cannot be guaranteed that these data are visible in the user's view. This would be particularly the case if a data focus is specified though the currently viewed region does not contain any of the specified data values. One way to engage this problem might be adding visual cues that lead to hidden information. Moreover, visual clutter can still appear in the context due to the detailed data representation. Thus, an adjustable degree of abstraction for the focus region and likewise for the intermediate context would be desirable.

We developed both approaches in tight cooperation with our domain experts (cf. [Section 7.2](#)), and thus we can state that it matches their requirements. However, formal evaluations are still missing. As a consequence, we presented in our last section of this chapter an approach to facilitate assessment of visualization by automatic means. To this end, we proposed the use of a *visual clutter measure* and a *saliency map*. We have shown that these two strategies can already indicate some hints regarding the quality of the visualization, in particular the global degree of clutter and the regions of an image that attract attention.

However, a currently still open point is the assessment of this method using concrete user studies and comparison of our results to these studies. Additionally, we would like to integrate other models for both, visual clutter and saliency, to validate the results.

In sum, we dedicated this chapter to scenario, respectively task dependent scalability of visual designs. In the following chapter, we will focus on ways to implement our concepts, considering different hardware capabilities.

# Chapter 6

## Implementation Strategies

In the previous chapters, we have presented different concepts for scalability with respect to the presentation of spatial data in terrain. This chapter is dedicated to the implementation of these concepts and addresses scalability on processing the data. For the rendering of the data two fundamental strategies exist: rasterization and ray tracing. We will investigate both strategies. Regarding rasterization, we will propose our visualization framework TEDAVIS that facilitates the design and application of comprehensive spatial data visualizations in 3D terrain. Afterwards, we will present a prototypical, scalable architecture for terrain rendering based on ray tracing.

### Contents

---

6.1	TEDAVIS: A Scalable Terrain-based Visualization Framework . . . . .	<b>105</b>
6.1.1	General Approach . . . . .	105
6.1.2	Requirements . . . . .	105
6.1.3	Architecture . . . . .	107
6.1.4	Application of TEDAVIS . . . . .	110
6.1.5	Discussion . . . . .	111
6.2	Scalable Architecture for Ray Tracing 3D Terrain . . . . .	<b>112</b>
6.2.1	General Approach . . . . .	112
6.2.2	Architecture . . . . .	114
6.2.2.1	Dynamic Task Management . . . . .	114
6.2.2.2	Scalable Ray Tracer . . . . .	116
6.2.2.3	Functional Model . . . . .	119
6.2.3	Implementation . . . . .	120
6.2.4	Discussion . . . . .	123
6.3	Summary . . . . .	<b>124</b>

---

---

The scalable visualization framework TEDAVIS that is introduced in [Section 6.1](#) was presented in the second part of the aforementioned journal article:

[Dübel and Schumann \(2017\)](#): Steve Dübel and Heidrun Schumann. *Visualization of Features in 3D Terrain* ISPRS International Journal of Geo-Information, 2017, 6, 357:1-357:20

The author conceived, designed, and implemented the approach and wrote most parts of the article. For this chapter, parts of the article were extracted, rearranged, and rewritten to fit the concept and wording of this thesis.

---

The approach of the scalable architecture for ray tracing in [Section 6.2](#) is based on the following publication:

[Dübel et al. \(2015\)](#): Steve Dübel, Lars Middendorf, Christian Haubelt, and Heidrun Schumann. *A Flexible Architecture for Ray Tracing Terrain Heightfields* Proceedings of the International Summerschool on Visual Computing, 2015

The author conceived, designed, and implemented the scalable ray tracing approach. The model for dynamic task scheduling is based on the work of Lars Middendorf ([Middendorf \(2014\)](#)). The functional model was designed in collaboration with Lars Middendorf. Finally, the paper was also written in collaboration. For this chapter, parts of the article were extracted, rearranged, and rewritten to fit the concept and wording of this thesis.

---

## 6.1 TEDAVIS: A Scalable Terrain-based Visualization Framework

In Chapter 4, we proposed a wealth of strategies to facilitate scalability with respect to different data aspects. Chapter 5 described various methods to support scalability with respect to the given tasks. For both types of scalability, we presented a large amount of design options. This allows refining and adapting the visualization to the given data and scenario. However, it is often not clear what impact the different methods and parameters have and how they affect each other. In order to support the design process, we developed a comprehensive visualization framework called TEDAVIS (short for Terrain and Data Visualizer). It enables testing different render strategies and configurations to create suitable visualizations.

### 6.1.1 General Approach

TEDAVIS is a sophisticated visualization framework. Figure 6.1 shows an overview screenshot of the framework, demonstrating various features. The software enables presenting large terrain models along with different types of spatial data. It allows for high-quality rendering of 2D, 3D, and stereoscopic images in real-time. TEDAVIS enables rapid prototyping and seamlessly switching between different visualization strategies with only a few clicks. Adjustments are incorporated into the render process immediately. Thus, the software supports experimenting with different setups in order to adapt the visualization to the given requirements. To this end, TEDAVIS supports all visualization strategies, design options, and parameterizations for spatial data visualization that we discussed in the previous chapters.

TEDAVIS is build upon a highly efficient rasterization kernel that enables extremely fast computation of visual representations. Moreover, our tool is based on a scalable architecture that facilitates the application of the software in different hardware environments. By utilizing so-called *Config Files*, exchanging various visual design configurations between different systems is assisted.

The following sections will first conclude the challenges that are engaged by our framework before going into detail of the fundamental architecture. Afterwards, we will explain how TEDAVIS facilitates scalable data visualizations in 3D terrain.

### 6.1.2 Requirements

While we have developed various strategies to enable scalable data visualizations, implementing them into one unified framework comes along with challenges on its own. In particular, we need to support the processing of the data together with the interactive design of the visualizations and their dynamic presentation, that means, showing the spatial data in terrain as a three-dimensional, dynamic scene. Moreover, we need to facilitate the application of the visualization not only on standard hardware, for example, during the visual design but also on specialized embedded hardware in cockpit environments during runtime.

**Data Handling** The starting point for any visualization is the data. Thus, a tool that facilitates the visualization of spatial data needs to support loading, processing, and storing various data aspects, including terrain models, geometrical data, and

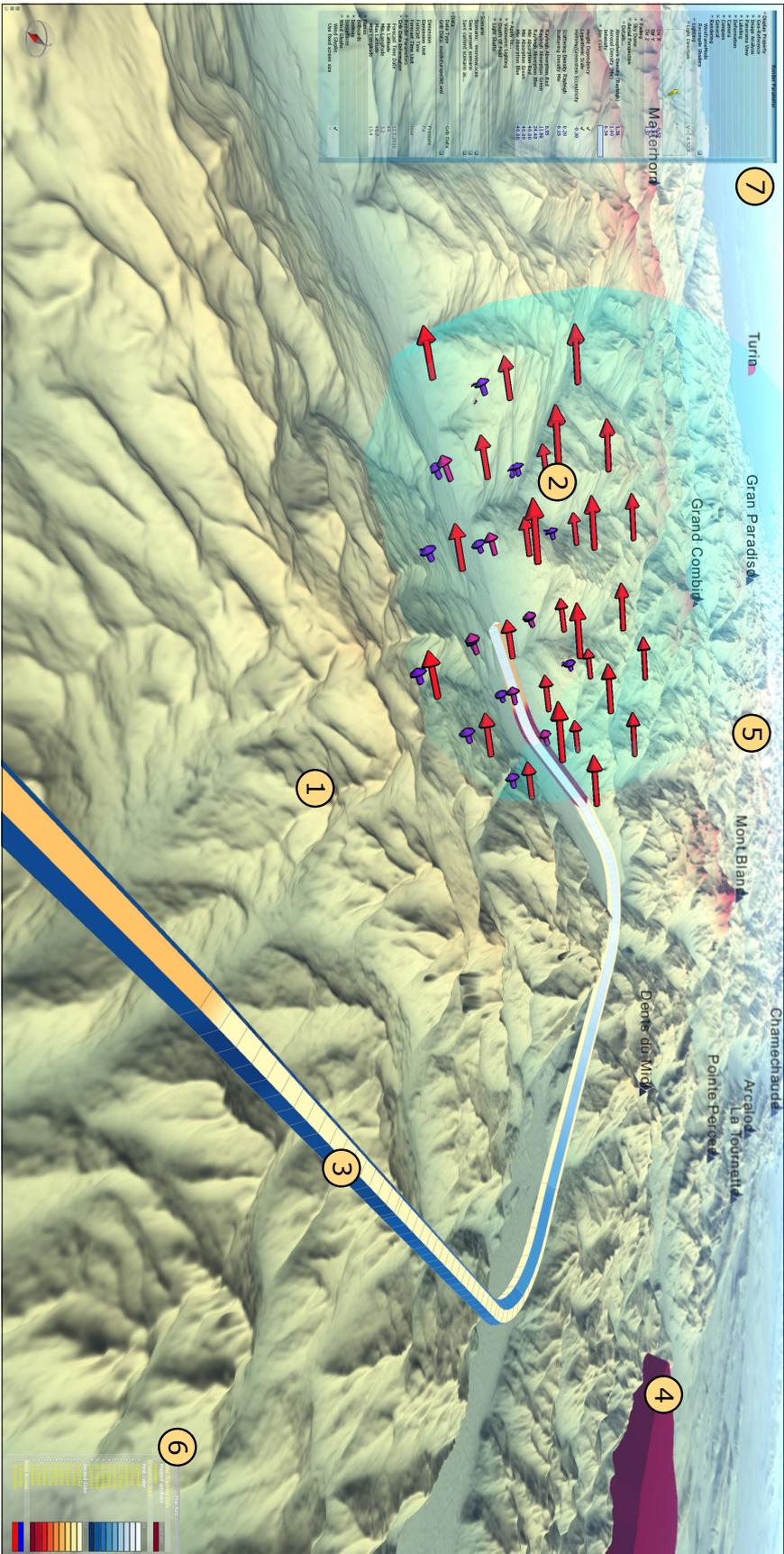


Figure 6.1: Overview screenshot of TEDAVIS that shows an elevation-colored terrain model of Central Europe ①, embedded wind data around the Sion airport ②, an approach trajectory of an aircraft ③, no-go-areas ④, and labels for points of interest ⑤. The applied color scales are specified in a legend shown in the lower right ⑥. The visualization can be interactively parameterized using the graphical user interface ⑦

numerical data. As we already mentioned, spatial data can be considerably large. Hence, the efficient handling of the data is an important requirement. For instance, data exchange between CPU and GPU must be kept to a minimum in order to prevent bandwidth overloading. The tool also needs to provide an interface for loading and configuring the data on demand to support the design of suitable visualizations.

**Interactive Design** To support the process of creation of scalable visualizations, the visual design must be configurable interactively. To this end, for each data aspect a suitable representation technique must be selectable. Thereby, not only the fundamental rendering strategies are of interest, for example using shading or contours for the terrain representation, but also the very details, such as the choice of the illumination model or the light direction. To this end, a responsive User Interface (UI) is required. In order to support the application of visualization concepts such as Focus & Context, we also need an appropriate mechanism that conveys the possible and to a certain extent the useful combinations of options. Eventually, an immediate reconfiguration of the visualization must be facilitated. Furthermore, to help with the design process, possibilities to assess the visualization are needed, for instance by providing indications about visual clutter and saliency as described in [Section 5.3](#). After finding a suitable design, possibilities to save the configuration are desired to enable continuing the design process at a later time or to transfer the design to other systems.

**Dynamic Visualization** Another requirement of such a tool is the visualization itself. Both the designer and the domain expert need to use and view the visualization during their respective tasks. In order to dynamically view the various data representation and at the same time sustain responsiveness, the visualization needs to be rendered fast. Preferably, the vis designer needs at least interactive frame rates ( $> 10$  Frames per Second (FPS)), while the domain expert typically requires real-time frame rates ( $> 30$  FPS). Considering the large amount of data available and the sophisticated rendering techniques as well as the variety of different hardware capabilities, this is challenging.

As we have mentioned, our visualizations are particularly, though not exclusively, aimed for spatial data presentations in cockpit environments. Hardware configurations in cockpits typically prioritize reliability over capability. Hence, typically only low-performance graphics systems are installed. Therefore, portability is another important requirement for our tool. In the following, we will describe the architecture of our framework.

### 6.1.3 Architecture

Our framework is designed to support an interactive design process even if handling large and complex data. To this end, TEDAVIS is developed as an efficient, modular and scalable framework. [Figure 6.2](#) shows the fundamental architecture. In the back-end, the Data Interface creates and utilizes efficient data structures and loading strategies. The User Interface in the front-end allows for interactive exploration and configuration of the visualization. The Rendering component finally applies sophisticated render techniques to create suitable outputs.

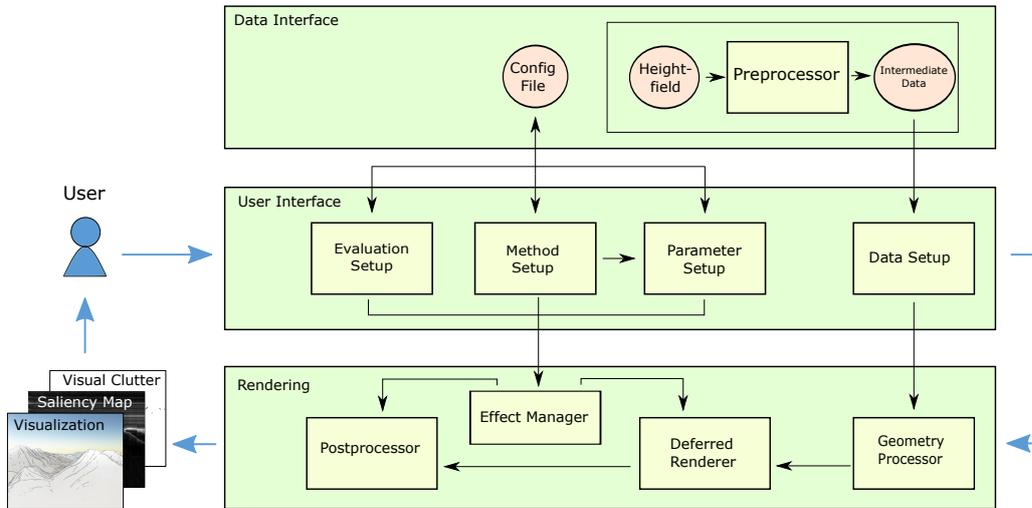


Figure 6.2: Overview of the architecture of TEDAVIS.

**Data Interface** The data sources of our framework are managed by the Data Interface (top of Figure 6.2). It supports various common *heightfield* formats (e.g., *hf2*, *ter*, *bt*, and *mmf*) and can process very large terrain models. For example, the terrain model shown in Figure 6.1 ① has a resolution of  $1.5 \text{ asec}$  and spans over a virtual area of  $600 \text{ km}^2$ , comprising approximately  $12 \text{ GB}$  of data. These massive data can only be rendered by using sophisticated **LOD** concepts, which we described in Section 2.1. In particular, TEDAVIS utilizes a quadtree-based data structure similar to the work of Ulrich (2002).

Additionally, we use a refined error metric that allows a pixel accurate model representation. This means that even though **LOD** is used, no perceivable geometrical errors occur, which is a strict requirement in cockpit environments. Generating this data structure from the heightfield can take considerable time. Therefore, a *Preprocessor* carries out this task beforehand, allowing TEDAVIS to resort to prepared *Intermediate Data*. During the preprocess, various attribute maps, including curvature or light maps, are computed as well, which further speeds up the render process. As for numerical data, TEDAVIS supports common data types for scalar fields, for example, weather data values given on a 3D grid (Figure 6.1 ②). Regarding geometrical data, the framework facilitates loading attributed trajectories ③, radar measured objects ④, and predefined obstacles, such as cable car posts. In addition, the tool can load and depict point-based, geo-referenced information, such as cities, airports, and mountain names ⑤.

Another functionality of the Data Interface is loading and serializing a snapshot of the application’s configuration, that are, selected render techniques, parameterizations, and used data sources. This information is serialized into a single *Config File*. This file is written in a human readable text format, making the configuration adjustable by external tools and hence allows for duplicating and scripting multiple scenarios outside of our software. By loading a Config File all respective data is loaded and the settings are restored, which allows for testing and application of preconfigured visualizations on different instances of TEDAVIS running on varied systems. The configuration can be adjusted afterwards and then be saved again.

**User Interface** The interface for the user (middle of Figure 6.2) is primarily implemented as a straightforward GUI (Figure 6.1Ⓓ). Basically, it supports a three-step procedure to design suitable visualizations. The first step is the *Data Setup*, which means choosing the desired data. This might also imply loading only parts of the data, for instance through filtering. Having specified the data that should be presented, the next step involves the *Method Setup*, where an appropriate render method is selected for each data type. For example, when a terrain model is loaded, the user can decide between different levels of abstraction, including shading by means of a specific illumination model or an appropriate NPR technique. Regarding numerical data the user can choose, for instance, between different intrinsic and extrinsic techniques. Considering strategies such as Prioritization or Focus & Context appropriate levels of abstractions can be utilized.

The *Parameter Setup* eventually enables configuring the selected render method. The user might parameterize the lighting situation, choose appropriate color scales, or apply specific depth cues. The *Evaluation Setup* can be used to deploy the visual clutter measure and the saliency map if needed. Through this, the user can assess the current visualization and reconfigure the design when appropriate. Whenever the configuration is changed, data, methods, or parameters are immediately updated in the rendering component, which is described next.

**Rendering** The Rendering component of our tool (bottom of Figure 6.2) has direct access to the configuration set of the User Interface. Thus, all changes are incorporated immediately. For this purpose, the *Geometry Processor* is checking for modifications in the data each render cycle. If the data have been changed, the associated objects are faded out and the geometry is constructed anew. From the data, the Geometry Processor builds up graphical primitives, processes them to efficient rendering resources, and stores them on the graphics hardware. Subsequently, the *Deferred Renderer*, deployed on the GPU, renders the graphical primitives, using appropriate render methods and effects. Effects are multi-pass shader programs maintained by the Effect Manager.

The *Effect Manager*, in turn, is using the configuration from the User Interface to adjust each shader pass and render method, resulting in the desired rendering outcome. The Deferred Renderer creates multiple image layers (k-Buffers), each containing different types of information about the rendered object, such as color, normal, or depth. Moreover, each data type is rendered into a separate render target.

Lastly, the *Postprocessor* creates the final image. Using the k-Buffers, it computes the final color of each object, taking into account the environment information such as lighting. Since all data types are rendered into separate render targets, otherwise global effects, such as aerial perspective or depth of field, can be applied locally only on a subset of objects. The Postprocessor also utilizes effects from the Effect Manager and is thus parameterizable by the User Interface. As an example, the calculation of the saliency map and the visual clutter measure induced by the Evaluation Setup are computed by the Postprocessor. The multiple render targets are finally blended and post-edited to ensure a proper image quality. If stereoscopic vision is applied, the render process is traversed a second time to create an image for the left and the right eye, using slightly different camera positions. Altogether, the Rendering component supports the deployment of all described design options and still enables a responsive design.

The architecture of TEDAVIS is build as a modular design. All components are interchangeable. We utilize this design in order to meet with the requirement of scalability. Based on

the available technology of the target platforms, that is, a cockpit system we developed distinct tailored modules to exploit and push the limits of the system's capability. In this process, TEDAVIS is primarily developed on a high performance platform. Afterwards, individual modules are ported and deployed to specific target systems by adjusting them towards the given hardware requirements. Due to the unified Application Programming Interface (API) of all components, newly developed components can be easily integrated into either system.

Regardless of the application system, Config Files are always supported. Thus, we are able to port created visual designs from one system to another. If specific rendering techniques are not supported by a certain system, alternative replacements are used instead. These replacements can be defined either by the system or by the visualization designer.

#### 6.1.4 Application of TEDAVIS

To design a presentation, firstly, the data are selected, secondly, an appropriate visualization method is chosen for each data aspect, and, thirdly, the presentation is parameterized to fit the requirements at hand. Afterwards, the visualization might be evaluated, using information about visual clutter and saliency and reconfigured afterwards.

Accordingly, a typical design work-flow would start with loading the data, for instance a terrain model for the area of interest. This data is initially displayed by a default visualization method. For instance, the terrain is by default shown by using edge enhancement combined with elevation-coloring. The designer is then be able to choose another visualization method that facilitates a suitable level of abstraction. Accordingly, shading or a combined method is used if many details of the terrain surface must be communicated. Finally, the parameters of the selected method needs to be adapt to the current demands. For instance, an appropriate illumination model, such as global illumination, needs to be chosen and certain hypsometric tints for the surface can be applied.

If the designer is satisfied with the terrain representation, another data source can be added and the depiction of this data type is then configured using the same work-flow. For example, either intrinsic or extrinsic visualization techniques can be chosen for any numerical data type. To prioritize a certain data aspect in the visualization, a suitable detail level needs to be chosen. By using appropriate color scales or other highlighting techniques, emphasis on the respective data aspect is facilitated. Alternatively, Focus & Context can be applied to either terrain, data, or both.

Eventually, the visualization needs to be fine-tuned, for instance to improve spatial awareness by using atmospheric effects or grids. The designer is able to apply such effects either to all data types or only to individual ones. This can be useful if data values are encoded by color and should not be affected by any effects placed on the terrain, or vice versa.

Since information on visual clutter or the saliency map are computed on the fly, the designer is able to evaluate the current visualization and refine it if required. The presentation can be adjusted by reviewing any step of the configuration in arbitrary order, thus, permitting the designer to view the changes at once. After the parameters are adjusted, the entire configuration can be saved along with the data. This configuration can then be loaded in any instance of the application. This allows, for example, to port the design to the target system and discuss it with the domain expert or test it with actual pilots.

The end-user such as the domain expert is able to apply TEDAVIS to test and use a visual design in their respective environment. The dynamic visualization facilitates the exploration

Table 6.1: Benchmark tests of TEDAVIS under different visualization scenarios. The table shows the minimum and maximum render FPS and the resulting average render time for Full HD images as well as the number of rendered triangles.

Scenario	Max. # Triangles	FPS	∅ Render Time
Kauai ( $\approx 0.5 GB$ )	$\approx 3.2 Mio$	342 ~ 563	2.2 ms
Central Europe ( $\approx 12 GB$ )	$\approx 6.0 Mio$	267 ~ 488	2.6 ms
+numerical data	$\approx 7.5 Mio$	168 ~ 254	4.7 ms
+geometrical data	$\approx 7.6 Mio$	128 ~ 199	6.1 ms
+atmospheric effects	$\approx 7.6 Mio$	64 ~ 78	14.1 ms
+volumetric lighting	$\approx 7.6 Mio$	58 ~ 72	15.4 ms
+saliency map	$\approx 7.6 Mio$	42 ~ 61	19.4 ms

of the representation either by interactive means, using mouse and keyboard, or by being toured automatically, for instance, along a predefined path. To ease the interpretability of data values, a legend (Figure 6.1 ⑥) is shown in the visualization. Additionally, the user can interact with the visualization. For instance, Focus & Context can be adapted by selecting a proper focus region or changing the size of the context.

### 6.1.5 Discussion

TEDAVIS is a state-of-the-art terrain visualizer implemented using OpenGL, GLSL, OpenCL and C++. For performance benchmarks, we tested our framework with different visualization scenarios to render Full HD images ( $1920 \times 1080 px$ ) and listed the resulting minimum and maximum frame rates and render times in Table 6.1. Among others, we tested our tool using a standard PC with an Intel Core i7-3770K CPU and 32 GB RAM as well as an NVIDIA Geforce GTX 780 GPU with 6 GB RAM. We inspected the performance of our tool under different scenarios, also listed in Table 6.1. The tested terrain data sets were a medium-sized model of Kauai ( $\approx 0.5 GB$ ) and a large model of Central Europe ( $\approx 12 GB$ ). All scenarios, except the first one, use Central Europe as terrain model. For numerical data, we used a wind data set consisting of approximately 100,000 wind glyphs and regarding geometrical data, we embedded ten larger trajectories into the terrain. In summary, our framework always sustained real-time frame rates, even when sophisticated effects, such as aerial perspective, volumetric lighting, or the saliency map were applied.

Tests on the target system in the cockpit show that the size of the terrain models must be restricted to facilitate real-time frame rates. Moreover, due to the lack of certain hardware specifics, some effects, such as volumetric lighting, cannot be realized. Nevertheless, TEDAVIS still enables dynamic visualizations of spatial data in 3D terrain on the target hardware.

Using this, the visualization designer has a potentially powerful tool at hand that can be applied to create and scale various presentation designs towards given requirements, in particular the data of interest, the given task, and the available processing capabilities. During this process, the designer is responsible to choose adequate configurations to minimize interferences, which may occur between the different effects. For instance, visualizing numerical data by intrinsic techniques on top of the terrain surface impedes the analysis

of local features of the terrain topography and vice versa. Using strategies such as Prioritization can help to decrease such interdependencies. The application of Focus & Context strategies can further improve the visualization.

Eventually, evaluations of the visual design are still required. For this purpose, TEDAVIS implements computational strategies, including the saliency map. But it also facilitates the application of designed visualizations and thus allows for tests with end users and domain experts. Due to its modular design, TEDAVIS is hereby well suited for deployment in different environments, but also for further upgrades, and for rapid prototyping.

## 6.2 Scalable Architecture for Ray Tracing 3D Terrain

As an alternative to the rasterization-based approach of TEDAVIS, we will now investigate how ray tracing can be used for rendering. We will do so by focusing on the presentation of the terrain. Due to the large size and high resolution of current models along with ever increasing demands on the visualization's quality, rendering 3D terrain is particularly challenging in this context. To facilitate scalability regarding hardware, we will use a flexible architecture to implement our approach.

As shown, rasterization is fast, flexible, and well supported by common GPUs. Ray tracing, on the other hand, has rarely applied in the field of visualizations since this render strategy is considered slow. However, ray tracing has advantages on his own. For one thing, ray tracing does not require a triangulated data structure. Thus, memory consumption for large geometric models, such as the terrain heightfields, is decreased considerably. On the other hand, ray tracing allows for global illumination effects that can significantly improve image quality.

Recently, more and more approaches towards interactive ray tracing of dynamic scenes have been developed. Beside efficient algorithms (cf. [Section 2.1.2.2](#)), interactive ray tracing requires an appropriate hardware support. This is mainly achieved by using multiple parallel processing units, including CPU-clusters or many-core-systems (e.g., [Govindaraju et al. \(2008\)](#) and [Seiler et al. \(2008\)](#)). In particular, the parallel architecture of common GPUs have been exploited to accelerate ray tracing.

However, current ray tracers can only be sufficiently fast if they are optimized towards one specific setup and thus they cannot adapt to changing requirements on image quality, computation time, or memory consumption. In order to facilitate flexibility to varying requirements, we need a more scalable approach.

Implementing a scalable ray tracer, however, comes along with problems for both GPU-based and CPU-based solutions. The recursive nature of ray tracing and the desired scalability of our approach fit not well with the pipeline-based programming model of GPUs. Optix ([Parker et al. \(2010\)](#)), a powerful and easy-to-use, general purpose ray tracing engine for the GPU, grants a better flexibility, but does not allow the user to fully customize acceleration structures, buffer usage, and task scheduling. On the other hand, the CPU permits high flexibility, but lacks high data parallelism. Thus, the performance of such CPU-based approaches is generally lower than GPU-based approaches. Hybrid solutions that run on CPU and GPU mostly suffer from the bottleneck of efficient communication. Hence, there is still need for novel approaches.

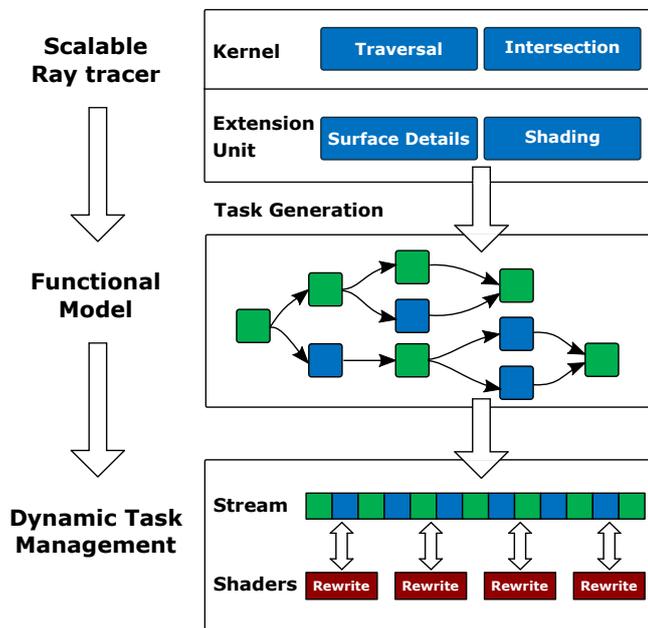


Figure 6.3: Rendering architecture for scalable terrain ray tracing.

### 6.2.1 General Approach

In order to provide more flexibility in customizing ray tracing of terrain heightfields, we propose a new rendering architecture that consists of three stages (illustrated in Figure 6.3).

The first stage (top of the figure) is a *Scalable Ray Tracer* that is made of two parts: First, a fixed, but highly efficient ray tracing kernel for terrain heightfields, and second, a flexible modular extension unit. The ray tracing kernel utilizes a beam tracing based fast start and quick intersection finding techniques to achieve interactive frame rates. The modular extension unit provides a set of enhanced rendering operators that can be activated on demand depending on the current requirements. The generation of surface details through interpolation, micro-structures, anti-aliasing and advanced shading can be generated dynamically by a set of appropriate functions. In this way, a better balance between required quality, computation time, and available resources depending on the hardware capabilities can be achieved.

The second stage (center of the Figure 6.3) provides a *Functional Model*, defined as a network of executable tasks, including parallel and recursive parts. Both the fixed ray tracing kernel, as well as the extension modules are described as individual tasks, that is, as nodes of a functional model. The functional model provides benefits like dynamic composition and recursion. However, it does not fit well into the data parallel execution model of current graphics hardware.

Hence, we utilize with the third stage (bottom of the Figure 6.3) an approach for *Dynamic Task Management* based on the work of Middendorf (2014) in order to provide the necessary scalable, hardware-based architecture. The task management encodes the functional model as a token stream that is iteratively rewritten by several shader cores in parallel. In particular, invocations are represented by specific patterns in the stream that are replaced after computation by the result of the corresponding functions. Therefore, all

types of indirect and recursive functions can be evaluated in parallel.

In sum our approach is based on a novel scalable terrain ray tracer, which application is facilitated by an execution model for scheduling dynamic workloads on varying hardware.

## 6.2.2 Architecture

In the following we will go into detail of our architecture. To this end, we will first describe the Dynamic Task Management of Middendorf (2014) that we will utilize to implement our approach. We will then introduce our scalable ray tracer and finally present the functional model that operates as a link between both stages.

### 6.2.2.1 Dynamic Task Management

Since the key idea of the dynamic task management is based on Middendorf’s work, we will only briefly describe the concepts required for our approach, that are, the execution model, its parallel implementation, and the local rewriting. For the details on the dynamic task scheduling we refer to our collaborated publication Dübel et al. (2015).

**Execution Model** The execution model operates on a token stream that stores the current state of a functional program and a set of *rewriting* rules that are iteratively applied to modify the stream. In particular, it is assumed that the program is given as a set of functions  $F := \{f_1, \dots, f_n\}$  and that the stream contains two different types of tokens to distinguish literal values from invocations.

Formally, a stream  $s \in S$  can be described as a word from a language  $S$  with alphabet  $\Sigma := \mathbb{Z} \cup F$ , while each function  $f_i$  maps a tuple of  $n_i$  integers to  $m_i$  output tokens  $f_i : \mathbb{Z}^{n_i} \rightarrow \Sigma^{m_i}$ . The rewriting step is specified by a function  $rewrite : S \rightarrow S$  that replaces the pattern  $(a_1, \dots, a_{n_i}, f_i)$  with  $a_1, \dots, a_{n_i} \in \mathbb{Z}, f_i \in F$  by the result of the invocation  $\langle r_1, \dots, r_{m_i} \rangle$  with  $(r_1, \dots, r_{m_i}) := f_i(a_1, \dots, a_{n_i})$ .

In particular, an invocation pattern takes a list of literal arguments and a reference to the corresponding function  $f_i \in F$ . If the function token  $f_i$  is preceded by at least  $n_i$  arguments  $(a_1, \dots, a_{n_i})$ , it is evaluated and replaces the original sub-stream. Hence, this scheme is equivalent to the post-order format also used in reverse Polish notations. Most important for a parallel hardware implementation, the rewriting affects only local regions of the stream and can be performed on different segments in parallel.

By starting with an initial stream  $s_0$  and iteratively trying to replace invocations, a sequences of streams  $s_{n+1} := rewrite(s_n)$  can be constructed, which limiting value can be considered as the final result:  $result(s_0) := \lim_{n \rightarrow \infty} s_n$ .

Due to the iterative rewriting, at least one pattern must be replaced by the function  $rewrite$  in order to guarantee the monotony of this sequence. As a result, an implementation is not required to detect every pattern in the stream. Instead, it can be partitioned more freely for parallel rewriting.

Despite the simplicity of this model, which is entirely based on find-and-replace operations, an efficient hardware implementation has to solve several issues that will be discussed in the following.

**Parallelization** For the purpose of efficient hardware implementation, the proposed algorithm can be parallelized by letting each core rewrite a different region of the stream. The partitioning of the stream into regions can be chosen almost arbitrary. Further, there

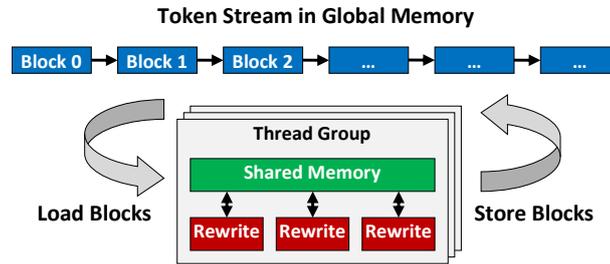


Figure 6.4: Global scheduling of the segmented stream.

is no need to consider the contents of the stream because data dependencies are resolved by pattern matching and the model does not define an explicit execution order. Instead, control dependencies are represented by local data dependencies.

However, an efficient implementation has to respect the architecture of modern graphics hardware, including GPUs that are optimized for data parallel kernels and therefore require a large number of threads to reach optimal occupancy. To this end, threads organized into groups are executed on the same processor and communicate through shared memory. Since the functional model creates and deletes a variable number of tokens in the stream, its length is continuously changing during the rewriting process. Hence, the data structure must be able to provide random access to permit the fast insertion and removal of tokens.

Therefore, the stream is partitioned into blocks of fixed size that are stored as a linked list in global memory, as shown in Figure 6.4. A distinction is made between the global scheduling of blocks at system-level and the local rewriting of individual tokens performed in the shared memory of each thread group. In particular, the concept of persistent threads is utilized, which run an infinite loop executing the following steps:

1. *Loading of blocks:* Depending on their size, one or two consecutive blocks are fetched from the stream and loaded into the shared memory. Due to the coherence of memory access the load operations can be merged to utilize the available bandwidth.
2. *Local Rewriting:* The stream is rewritten locally and the results are stored in the shared memory. This process can be optionally repeated several times and requires multiple passes as well as random memory access.
3. *Storing of Blocks:* The resulting tokens are written back into the global stream and additional blocks are allocated. The occupied memory of these blocks is released afterwards.

Most of these steps can be performed independently by several thread groups in parallel. Hence, especially the local rewriting but also the loading and storing of blocks can benefit from a parallel hardware architecture.

**Local Rewriting** Unlike the global scheduling process that is partitioned across several thread groups, the local rewriting on the stream must exploit the data parallelism of a single multiprocessor using shared memory. Since each function consumes and emits a different number of tokens, the stream must be adjusted accordingly. In particular, for each of the two token types two possible actions can be described:

- *Literal value tokens* are either removed from the stream if the corresponding function is executed or they are copied for the next iteration. Hence, a literal always creates one or zero outputs.
- *Function tokens*  $f_i \in F$  either produce the specified number of  $m_i$  outputs if a sufficient number of literals are available or they are kept on the stream. Thus, a function token needs to be rewritten into one or  $m_i$  output tokens.

Since the local rewriting in shared memory needs to employ coherent control flow and data parallelism, it is structured into three passes:

1. *Decoding the stream*: The stream is scanned for executable patterns, and for each token the number of outputs is computed according to the two cases above.
2. *Allocating outputs*: Depending on the number of outputs, the new position of each token is calculated.
3. *Executing functions*: The functions determined in the first step are executed and their results are stored at the positions computed in the second step. Finally, the remaining tokens that do not participate in an invocation are copied to the output array.

In summary, the dynamic task management allows to solve a token stream by executing functions with literal values in parallel and adjusting the stream by removing executed functions and adding their results. Further details can be found in the dissertation of Middendorf (2014). In the following, we will present our scalable ray tracer.

### 6.2.2.2 Scalable Ray Tracer

The ray tracing stage consists of two major components: For one thing, a fixed ray tracing kernel that traverses the rays through a discrete terrain heightfield, and for another thing, a modular extension unit that controls surface generation and shading. While the ray tracing kernel generates a simple unicolored or textured terrain representation, the extension support flexibility to respond to given requirements.

**Ray Tracing Kernel** To ensure interactive rendering, we apply and combine sophisticated techniques from literature for both ray traversal and intersection tests. We utilize a grid-based bounding volume hierarchy, in particular the Maximum Mipmap (**MM**) (Tevs et al. (2008)). The **MM** is structured similar to a quad tree and stores the maximum of all underlying height values at each node. The root node spans the whole heightfield, while a leaf node stores the maximum of four actual height values of the field. This structure can be constructed very fast and since spatial information is stored implicitly, the increased memory footprint is low ( $\approx 33\%$ ). The nodes of the **MM**-tree operate as bounding volumes for the underlying patches based on the height values. When tracing the rays through the scene, the tree is traversed to find the locations of potential intersection with these patches.

To decrease rendering time, we apply a beam tracing based fast start. Instead of traversing all rays starting from the image plane, that is, starting from the root of the **MM**-tree, more suitable starting points are identified through a priorly performed beam traversal. The beams hereby approximate a bunch of rays for a certain image area (e.g.,  $8 \times 8 px$ ). The rays defined by the corners of the base area are traversed through the **MM**-tree. As long as these four rays take the same path through the tree, it is guaranteed that all rays

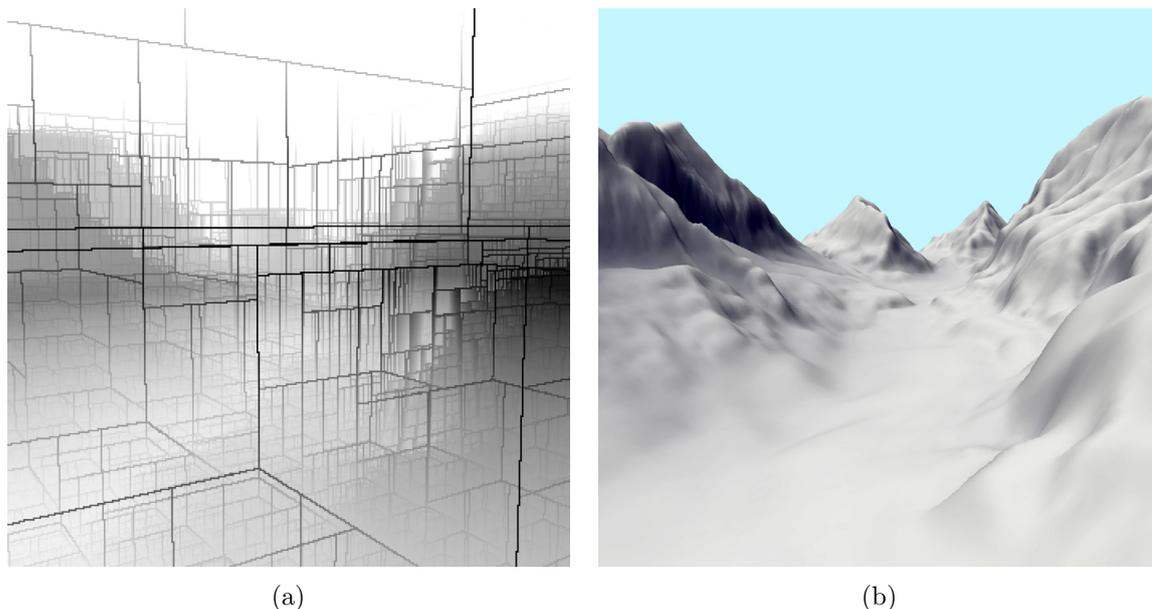


Figure 6.5: Illustration of the beam tracing based fast start. (a) The result of beam tracing is the starting position of individual ray traversal encoded as a depth map, which means that the map encodes the distance of the intersection points from black for near intersection points to white for intersections far away. The grid structure reflects the point of disjunction at the edges of the BVH. (b) The image rendered by means of the beam tracing’s result.

of this beam traverse the tree in the same manner. If the rays visit different nodes or hit different sides of the bounding volume, beam tracing needs to be replaced by traversing individual rays. However, the individual ray traversal can now start at this particular point of disjunction, as illustrated by Figure 6.5.

Since an exact calculation of intersection points between a ray and a surface patch of the heightfield is time consuming, we approximate the intersection using uniform and binary search (Policarpo et al. (2005)). An advantage of this method is the abstraction from the real structure of the patch. Instead, the intersection test is based only on the height values ( $z$ ) at given coordinates ( $x,y$ ). Therefore, the generation of patches themselves can be encapsulated by operators of the modular extension unit.

The ray tracing kernel, as described here, was implemented as a stand-alone application on GPU and resulted in real-time frame rates ( $\approx 100$  FPS at HD resolution for a moderate-sized heightfield). However, this application is missing the scalability to balance between reduced rendering time and improved quality. Therefore, our architecture extends the ray tracing kernel with a flexible modular unit.

**Modular Extension Unit** The Modular Extension Unit consists of a set of operators that allow to improve image quality on demand. In particular, we utilize enhanced operators for surface generation and shading, though further operators can be added easily.

Considering *surface generation*, the ray-patch-intersection computed by the ray tracing kernel is based solely on height values. Thus, the surface patches for this computation can be generated by enhanced operators on the fly. Through this, we can adapt the render process with respect to our requirements. As a first step, we will examine different methods

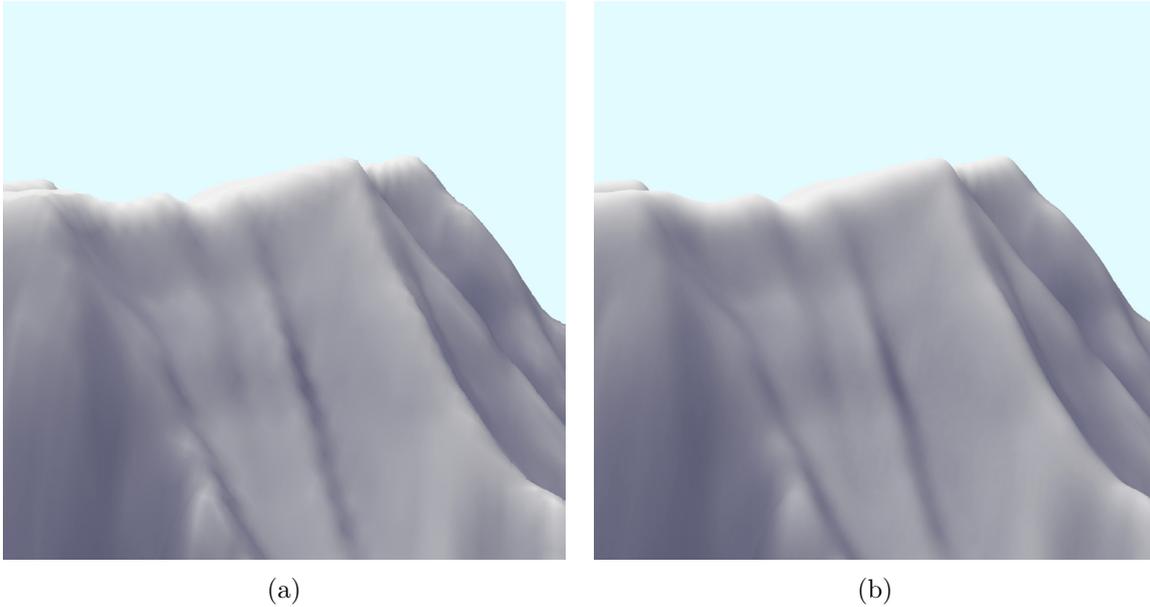


Figure 6.6: Different strategies for surface interpolation. (a) Bilinear interpolation introduces artifacts, such as discontinuation within the silhouette and at shadow edges, whereas bicubic interpolation (b) provides smoother transitions.

for surface generation, that are, interpolation, additional micro-structures, and antialiasing.

With regard to *interpolation*, we generate a surface patch prior to the computation of intersections by a specific operator of the extension unit. As a default, the patch is generated by bilinear interpolation between four neighbored height values. However, bilinear interpolation can introduce artifacts at the silhouette and shadow edges, as illustrated by Figure 6.6a. Hence, a specific enhanced operator provides smoother, bicubic interpolations. As Figure 6.6b shows, bicubic interpolation results in less artifacts, however computation complexity ( $O(n^3)$ ) is enormously increased compared to bilinear interpolation ( $O(n)$ ).

As a midway solution, we provide an approximated bicubic interpolation, where the surface is continuously refined by generating additional points along a bicubic function, though the eventual intersection point computation is again based on a bilinear interpolation in between these points. This operator provides both a sufficient performance and an adequate quality.

However, even the application of bicubic interpolation is sometimes not sufficient if the resolution of the heightfield does not provide fine granular details. In this case, we can modify the surface by using additional *micro-structures*. By applying different operators to enrich a terrain surface with extra details, image quality can be improved. To define the micro-structures, either noise functions or additional relief maps are suitable. Figure 6.7 exemplifies our approach. Though micro-structures increase image quality, ray tracing the additional details typically increases rendering time and memory consumption.

Another approach to improve render quality is *antialiasing*. If only one ray per pixel is traversed, aliasing artifacts appear in the distance where the surface is under-sampled. Therefore, we provide enhanced operators to counteract these issues. On the one hand, we utilize an average mipmap that allows for trilinear interpolation of height values depending on the distance. As a drawback, this increases the memory footprint. Alternatively, multiple rays can be traversed per pixel. In this case, enhanced operators invoke supplementary ray

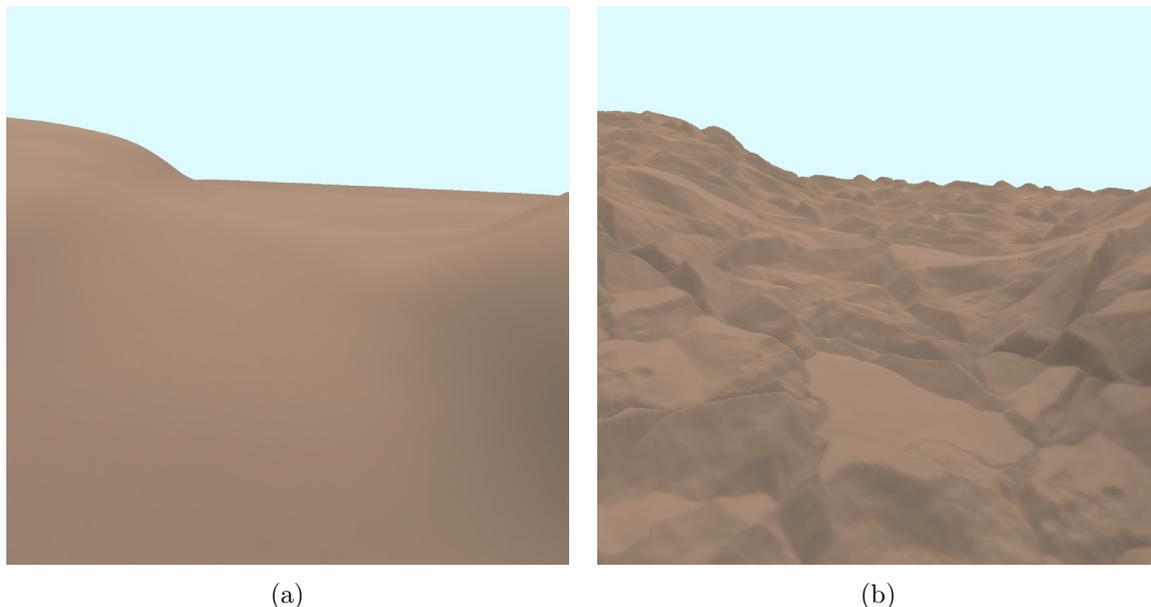


Figure 6.7: Surface generation through application of micro-structure. (a) The smooth interpolated surface of the heightfield lacks fine granular details. (b) To increase realism, the surface can be enriched by micro-structures that displace the height values along the z-axis.

tracing for surfaces in the distance. This, in turn, increases rendering time.

With respect to *shading*, the ray tracing kernel assigns a material color to each visible point. In order to adjust the image quality, the modular extension unit provides different enhanced operators for customization. For one thing, we support a full recursive ray tracing, which is the most time consuming but also most accurate method. For another thing, we support different shading methods that we already described in [Section 2.1.2.3](#). Depending on the respective shading method, render time and memory footprint can vary considerably.

In conclusion, we provide a highly efficient but fixed ray tracer that allows to visualize the terrain on a rather abstract level. In order to improve the quality of the terrain depiction, we developed various enhanced operators that facilitate sophisticated shading and detailed surface representation. We apply these operators in a modular manner to enable scalability. Therefore, we utilized the approach of dynamic task management (cf. [Section 6.2.2.1](#)). To implement our ray tracer using this architecture, we developed a functional model that converts the generated task into a token stream.

### 6.2.2.3 Functional Model

The functional model forms the interface between the scalable ray tracer and the dynamic task management that is executed on the hardware. It is composed of individual tasks. Each task represents an operator of the ray tracing stage. The tasks are connected with regard to the processing flow.

They compose a network of sequential, parallel, and recursive tasks. The basic configuration consists of a combination of ray traversals, intersection tests, and additional enhanced operators of the extension unit. The choice of enhanced operators hereby determines the

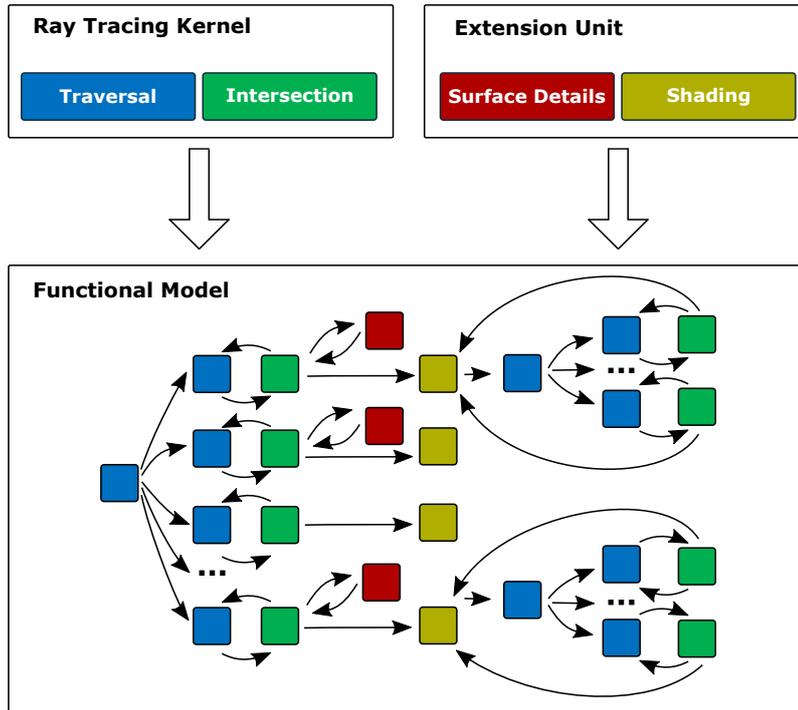


Figure 6.8: Example for task generation from the scalable ray tracer to the functional model. The individual tasks form a parallel and recursive network.

network’s topology. Recursive ray tracing, for instance, maps to a recursive network structure of the tasks, while the parallel traversal of individual rays maps to a parallel structure, as is exemplarily shown in Figure 6.8. Afterwards, the network is transposed into a token stream by first inserting a single task of ray traversal that expands automatically with the invocation of further tasks, such as intersection calculation.

To support the decision which enhanced operators to use, we can describe presets that either focus on rendering time, rendering quality, or memory consumption. Hereby, we support flexible adjustments to current hardware capability and workload. For example, when rendering time is prioritized, we use simple bilinear interpolation and simple shading. For quality-based configuration, we utilize bicubic interpolation, add details through micro-structures and reduce artifacts in the distance through antialiasing. Moreover, we can apply high-quality operators, such as full recursive ray tracing. Memory-based set-ups, in turn, will omit precomputed shading operators and additional micro-structures to reduce memory consumption. When antialiasing is used, sub-sampling will be favored over additional average mipmaps.

Through these presets that define the functional model we can scale our algorithm to previously known hardware conditions with respect to a desired quality level. But, varying data complexity and/or further constrains, such as ensuring minimal frame rates, might require the adaption of the functional model on the fly. This does not pose a problem, since we can dynamically insert, remove, or replace tasks onto the token stream. The dynamic task management is then able to solve the dynamic token stream in parallel on the hardware.

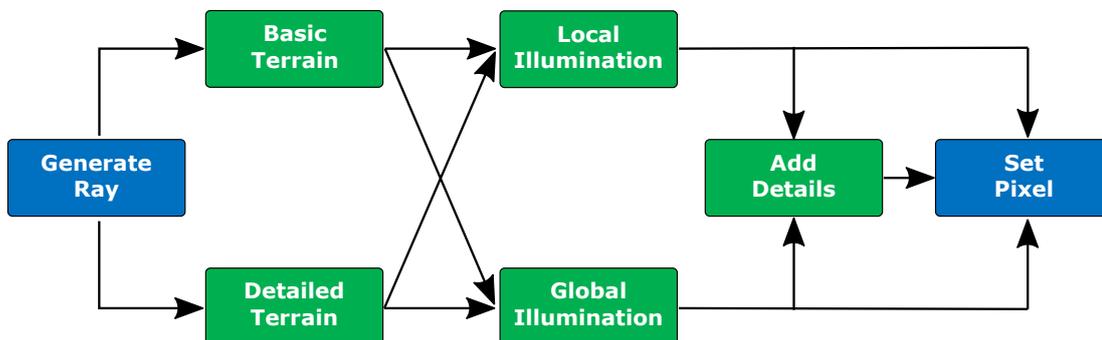


Figure 6.9: Exemplary function consisting of fixed (blue) and flexible (green) operators for ray generation, surface computation, and shading.

### 6.2.3 Implementation

A first prototype of our scalable architecture for terrain rendering was implemented and evaluated to show principle feasibility. As a testing, parallel hardware, we utilize a GPU, in particular a GeForce GTX 780, using the CUDA 7.0 API.

The rewriting process is started by a single kernel launch and performs the algorithm described in the previous section. For this purpose, the stream is divided into blocks of 512 tokens and each thread block stores at most two of them in the shared memory. In addition, there are 16 to 64 threads per thread block that decode the stream in parallel and execute the detected functions.

The structure of our prototypical ray tracer is shown in Figure 6.9 and consists of several stages, described by our functional model. In particular, we can switch between two detail levels of geometry and two lighting modes to scale between quality, resource usage, and computation time. First, the fixed ray tracing kernel, which creates a ray for each pixel, generates the initial stream. It flexibly emits function calls either to the stages *Basic Terrain* or *Detailed Terrain* that compute the intersection point of the ray and the terrain surface through a combination of uniform and binary search. Similarly, we can switch in the next stage between local or global illumination models. For the local model, we utilized Phong illumination, while for the global model we applied as a first test ambient aperture lighting (cf. Section 2.1.2.3). In addition, the color is modulated by a micro-structure and is interpolated for nearby pixels of the detailed terrain (*Add Details*). Finally, in the stage *Set Pixel* the color data are written into the rendering target.

The ray tracer was evaluated using three different DEMs: (PerlinNoise, Fuji, and Himalaya) with an equal size of  $512 \times 512$  samples and different configurations for geometry details and lighting models to show the fundamental feasibility of the approach. For performance comparison, each test setup is used to draw 10 frames. The average rendering times per frames are summarized in Table 6.2.

Rendering times are similar for all three data sets. We therefore conclude that the fixed ray tracing kernel is robust for data of equal size. Also, the rendering times for static and dynamic lighting differ only slightly. Accordingly, this decision affects only the quality of the image. However, this would be different if more complex illumination models, such as full recursive ray tracing, would be used. The results also show that the detailed rendering mode

Table 6.2: Rendering time for different configurations. Net Render Time is determined by the average render time without the system overhead (65 ms) to handle stream rewriting.

Terrain	Geometry	Lighting	Avg. Time	Net. Time
PerlinNoise	Basic	Static	99.5 ms	34.5 ms
		Dynamic	101.1 ms	36.1 ms
	Detail	Static	136.5 ms	71.5 ms
		Dynamic	136.9 ms	71.9 ms
Fuji	Basic	Static	99.6 ms	34.6 ms
		Dynamic	99.7 ms	34.7 ms
	Detail	Static	136.6 ms	71.6 ms
		Dynamic	137.0 ms	72.0 ms
Everest	Basic	Static	99.1 ms	34.1 ms
		Dynamic	100.0 ms	35.0 ms
	Detail	Static	136.4 ms	71.4 ms
		Dynamic	136.6 ms	71.6 ms

of all terrains requires more computational resources, but also creates more sophisticated images (cf. Figure 6.6 & 6.7).

Tests showed that a great portion of the rendering time ( $\approx 65$  ms) is consumed by the dynamic task management and thus by the rewriting algorithm itself. Therefore, additional tests to evaluate the core rewriting algorithm were performed. The scalability of the rewriting algorithm was analyzed for the basic and detailed *Everest* terrain by varying the number of launched thread blocks. Since different thread blocks can run on distinct multiprocessors in parallel, a linear speed-up should be expected, but there are two possible bottlenecks. First, the linked list of blocks represents a global synchronization point and is protected by a lock. However, each thread holds the lock only for a short amount of time. Second, the complete stream is stored in global memory and must be copied into shared memory for rewriting. But the memory is accessed coherently in order to maximize the available bandwidth.

The resulting measurements, as shown in Figure 6.10a, indicate continuously decreasing render times that stagnate at approximately 15 thread blocks. When the number of thread blocks is doubled, performance increases accordingly up to 16 blocks (Figure 6.10b). Since our graphics hardware used for evaluation consists of 15 streaming multiprocessors and similar tests on other graphics hardware showed the same coherence, we conclude that each thread block is mapped to a different multiprocessor and that each multiprocessor executes at most one thread block. Since the hardware can execute at most 16 threads per thread block, we set the number of threads for this test to 16. However, the configuration resulting in the best performance, as seen in Table 6.2, utilized 8 streaming multiprocessors and 256 threads per block. This indicates that internally, thread blocks and threads can be mapped differently depending on configuration and driver.

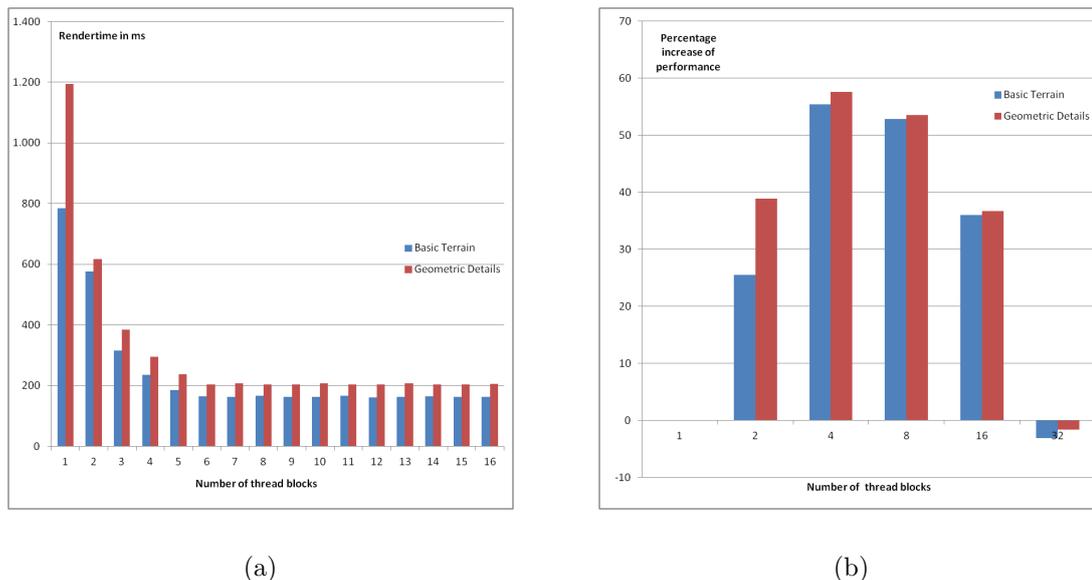


Figure 6.10: Results of the performance tests. (a) Rendering time for different number of thread blocks and 16 threads per block and (b) increase of performance when number of thread blocks are doubled.

#### 6.2.4 Discussion

Aim of our approach is to balance rendering time, quality, and resource consumption with respect to given hardware requirements. For this purpose, we examined how to incorporate ray tracing into a scalable architecture for parallel hardware. Our approach is composed of a fixed, high-efficient ray tracer and flexible, modular extensions. By combining them, we can adjust the rendering process on demand, considering previously known hardware conditions or dynamic requirements, such as guaranteeing particular frame rates when the current workload changes. To facilitate such a scalable approach, we utilize the dynamic task management based on the works of Middendorf (2014) to map our scalable ray tracer onto the graphics hardware. A first prototype demonstrates the feasibility.

Our tests show the fundamental scalability of our approach, but hardware limitation on our used graphics card currently restricts further scalability. Therefore, the overall-performance is still lower than the classical rasterization approach (e.g., TEDAVIS) and provide only interactive but not real-time frame rates. Further improvements and optimizations, however, should permit the execution of multiple thread blocks per multiprocessor to further scale our approach and increase performance. The algorithm is not limited to the GPU but can be implemented on other parallel hardware, including CPUs and FPGAs. Thus, our approach can enable scalability along special hardware environments where classical rasterization may not be feasible, for instance in embedded systems.

Nevertheless, open questions still remain. Currently, manually composed presets determine the structure of the functional model. However, an automated adjustment by means of restrictions, such as minimum frame rates or minimum quality standards, has to be further examined. Moreover, we have currently investigated ray tracing for terrain rendering only. But visualizing spatial data along with the terrain is still desired. To this end, our architecture that facilitates dynamic invocation of additional tasks is well suited to extend

the visualization by additional elements. For example, the shading operators can be used to implement further intrinsic data visualizations or additional operators can render further graphical primitives that encode the data. In order to still ensure interactive frame rates, we would need to save render time during the terrain ray tracing. This can be easily facilitated by our flexible architecture through replacing, for instance, quality-based operators with operators that focus on render speed. For example, we would use simple illumination models and fast surface approximation for the terrain and utilize the now-available render time for data visualization.

### 6.3 Summary

In this chapter we proposed two implementations to facilitate scalability on processing level. Firstly, we described TEDAVIS, a high performing visualization framework that integrates various visualization designs and enables interactive presentation of spatial data in 3D terrain. By implementing the strategies proposed in this thesis into one unified system, TEDAVIS also works as proof of concept. Using TEDAVIS, we are able to design, test, apply, and evaluate our visualizations. By computing a saliency map and measuring the visual clutter, first indications for the visualization's quality can be given by our tool. Furthermore, our modular architecture and the unified exchange format for design configurations allow us to use our tool along multiple systems, such as standard desktop PCs and embedded cockpit workstations, to which end we facilitate hardware scalability. Thereby, we support the exchange of visual designs between different systems by Config Files that store the configuration of a concrete representation.

Secondly, we introduced a scalable architecture for terrain ray tracing based on a functional model that permits the flexible composition of complex and recursive render modules. To this end, we developed a ray tracer that allows to strike a balance between image quality, render time, and resource consumption depending on the demands at hand. In order to implement our scalable approach, we argue that current implementation models of GPU and CPU are not sufficient. Instead, we utilized a flexible task management architecture that exploits the concurrency of general parallel hardware for a large number of dynamically created tasks with interdependencies. This architecture operates on a token stream that is iteratively rewritten via pattern matching on multiple shader cores in parallel. We provide this stream by mapping our ray tracer onto a functional model.

## Chapter 7

# Conclusion and Outlook

Designing suitable visualizations for spatial data in 3D terrain is challenging. For one thing, data sets used today are large, complex, and heterogeneous. For another thing, the scenarios where these visualizations are needed are similarly multifarious. In order to engage these challenges, we argued that we need sophisticated, scalable visualization approaches that flexibly adapt to given demands. To this end, this thesis investigated concepts for scaling visualization in the context of large spatial data and changing requirements.

### 7.1 Summary

In this work, we first gave a comprehensive overview about visualization approaches regarding spatial data in [Chapter 2](#). We hereby investigated in particular the visualization of three data aspects, that are, the terrain, the data, and uncertainty. We briefly summarized existing visualization methods for the terrain and the diverse data types and identified open questions regarding the development of a scalable visualization.

In [Chapter 3](#), we introduce the field of application, that is, the area of avionics, in which context this thesis was developed. We exemplarily demonstrated varying flight scenarios, which induce the need for scalable visualization. We discussed the prerequisites for such scalability on level of presentation and on level of processing and from here outlined our approach.

As a first step, we presented in [Chapter 4](#) a novel systematization for spatial data visualization with respect to the dimensionality—either 2D or 3D—of the terrain’s representation and the data’s representation, respectively. The systematization hereby considers the four possible combinations of 2D and 3D presentations for terrain and data. The worked out fundamental characteristics of either category that facilitate a comprehension of the respective method’s capabilities.

Afterwards, we investigated scalability with respect to the data by demonstrating diverse design options for varying data aspects, in particular for terrain, geometrical and numerical data, as well as for uncertainty information. We concluded the particular characteristics, and thus lied the foundation for a flexible application of respective presentation methods depending on the given demands.

With these wealth of design options at hand, we developed in [Chapter 5](#) two different strategies to combine them into comprehensive presentations. We first presented a novel strategy, that is, Prioritization. This strategy allows for emphasizing one particular relevant data aspect in a visualization. Which aspect is relevant is determined by the given scenario.

In this sense, the terrain would be particularly relevant in landing approach scenarios and hence would be prioritized. The prioritized aspect is presented with particular detail or specifically prominent. For example, the prioritized terrain is represented using high quality shading instead of just contours to communicate the fine details in the surface. Meanwhile, the other aspects are represented using less details and emphasis. For instance, we show them by using aggregation or less prominent visualization methods. We showed feasibility of this approach by means of exemplary, aviation-related application scenarios.

Our second strategy was the application of Focus & Context for spatial data visualization in 3D terrain. While Prioritization operates on a global level considering one whole data aspect, Focus & Context can be used on a more local level to facilitate scaling the information presentation for a certain subset of the data. To this end, we propose the utilization of two types of foci: spatial focus and data focus. The spatial focus selects a spatial region of interest, such as a point, path, area, or volume. The data focus selects a region in the attributes of the data, such as an interval of interesting data values. To represent the terrain or the data in the focus, we use a similar concept of accentuation that we did for Prioritization, that is, using a low level of abstraction on these data. We then deduce the context region from the focus and attenuate the herein depicted data.

Finally, we proposed another innovative approach that addresses the evaluation of visualization designs using saliency and visual clutter information. Saliency, in this respect, means the prominence or conspicuity of a feature in the presentation. Knowing about the salient regions in an image allows to specifically exploit these regions to communicate the relevant data. Knowing about visual clutter in the image allows to apply particular reduction strategies. We implemented suitable models to compute information about saliency and visual clutter interactively during the design process.

In [Chapter 6](#), we introduced our sophisticated visualization framework TEDAVIS that implements all the proposed design strategies from [Chapter 4](#) and [5](#). TEDAVIS facilitates the design of comprehensive presentations for spatial data in terrain and enables the application of the visualization for the domain experts. The rasterization-based rendering component of TEDAVIS allows for rapid prototyping and seamlessly switching between different visualization designs. The scalable architecture furthermore enables the integration of the tool in varying application environments, such as desktop PCs or target platforms embedded in the aircraft's cockpit.

Finally, we presented a novel, scalable architecture for terrain ray tracing on parallel hardware. By using a flexible functional model that enables scheduling multiple rendering tasks in parallel, we build a scalable ray tracer that facilitates dynamically balancing between image quality, render time, and resource consumption.

## 7.2 Comments on Design Process and Experts' Feedback

The strategies and designs presented in this thesis were developed in collaboration with domain experts from the aviation industry. They have to deal with multi-faceted data and 3D terrains in cockpit environments. Therefore, we will briefly summarize the design process that yielded the results presented in this work.

In a first phase, we discussed the general and specific requirements in the domain context. It was agreed that visualizing multiple aspects of data is a necessity. However, the domain experts requested that the complexity of the visual representations to be kept at a reasonable level. We identified various aspects of interest, such as 3D terrain, weather data, uncertainty,

trajectories, airports, obstacles, cities, mountains, and flight zones. Additional aspects, such as flight data or the temporal aspect of the data, were deemed secondary and left for future work.

In a second phase, we investigated the design options for the different data aspects that we described in [Chapter 4](#). As a result, we narrowed down the results to the approaches reasonable in cockpit environments. In the light of the huge amount of data to be visualized, we concluded that plainly showing all information regardless of their relevance would lead to visual clutter, which implies high mental load when interpreting the visual representations. Therefore, together with the domain experts, we looked into the different scenarios that cover typical flight settings. We discussed the requirements and tasks that are relevant in these scenarios, as described in [Chapter 3](#).

In the third phase, we develop generic strategies for prioritized visualization that can later on be refined according to particular application requirements. As a result, we obtained the strategies described in [Chapter 5](#). Based on these strategies, we developed various principal visualization designs for each scenario.

We followed an iterative process so that feedback from our partners could be incorporated to improve the designs. For example, initially, we proposed to use local illumination when the 3D terrain is less relevant. However, the domain experts expressed concerns that even such a low-fidelity representation could be too realistic in some situations. They suggested to go for more abstract representations based on silhouettes only. As a good compromise in terms of abstractness, we finally settled with a combination of silhouettes and contours.

To facilitate the close cooperation with our partners, we developed predefined design suggestions created using TEDAVIS and exchanged the configuration in form of Config Files with the experts. Since multiple instances of TEDAVIS were deployed in our and in their systems, quick tests and rapid adjustments were easily facilitated.

A necessary next step is to conduct a formal user study of our approach. Given the rich set of options for parameterizing the visualization, designing a controlled study poses considerable challenges. Therefore, a larger evaluation is planned in close collaboration with the domain experts using TEDAVIS integrated into a system targeted for cockpits.

## 7.3 Future Work

In this work, we have investigated many aspects of scalable data visualization in terrain. Nevertheless, there are still open questions and research direction for future work.

**Further data aspects** This thesis investigates various visualization designs regarding spatial data in 3D terrain. We also considered the visualization of the data's quality. However, we did currently not consider the temporal aspect of the data. Though much research has been conducted regarding the visualization of temporal data, embedding them into a depiction of 3D terrain is still a current challenge. Future work could examine, for instance, the visualization of the temporal aspect using Prioritization. This aspect could then be visualized additionally in combination with others. Further data aspects that have not been considered yet, but are often useful in terrain representations, are, for example, buildings, streets, or vegetations.

**Enhancements on scalable visual design** For both Prioritization and Focus & Context, we currently used only two levels of abstraction, that are, a detailed, emphasized representation and a coarser, de-accentuated representation of the data. Further research could help to find more distinct levels in between. These additional levels would allow to go beyond the current binary choice and facilitate a better customization of the visualization. With view to Prioritization, we currently deem one aspect most important and consider the others having an equally low relevance. Additional abstraction levels would facilitate different importance levels for multiple aspects and thus a more tailored visualization.

Such a finer gradation could also be useful when considering our Focus & Context approach. Currently, we use many details for the focus and less details for the intermediate context. For the remaining context, we currently omit the data and show the spatial frame-of-reference only. Using additional levels of abstraction, data in this remaining context could be shown, yet in an even more abstract manner and thus provide a better overview on the data.

**Enhancements on implementation** In the future, we will further extend our framework TEDAVIS by integrating additional data sources, such as 3D models of obstacles or cities. We are also interested in integrating further enhanced render techniques and to improve the GUI of the application.

With respect to our ray tracing approach, we need to further optimize the system in order to achieve sufficient high frame rates. Moreover, we need to investigate how data representations can be added into the terrain depiction. In particular, we need to identify efficient data structures and other acceleration techniques in order to facilitate rapid ray tracing of the data encodings.

**Evaluation:** Currently, we pursued two strategies to evaluate our visual designs. On the one hand, we used a quantitative measure for saliency and visual clutter. To this end, we implemented two specific models. Further research is required to investigate and test additional models and compare the result to our current approach. Moreover, we are interested in further perception-based measures, such as visual spatial attention measures.

On the other hand, for a qualitative evaluation we used domain expert feedback as we designed our visualizations in close collaboration with our partners. As a next step, we want to perform a formal study with end-users to evaluate the visualization and to receive further hints for improvements. Such a study is planned for the near future.

In this thesis, we focused on scalability regarding different facets, in particular varying data, varying tasks, and varying hardware. However, we currently addressed these facets in the context of our application case only. In order to apply scalability in other domains, we need to take a more general view on them and investigate how the respective requirements can be engaged. Moreover, we currently considered only three facets (data, task, and hardware). Another relevant scalability aspect, for example, would be the user. Though in general our visualizations are developed having only a single user in mind, in practice, visual data analysis often involves multiple users, in occasion even at the same time. Thus, we need to examine how to scale our visual design towards collaborative work and how to account for

the user's individual characteristics. Additional scalability aspects that are also relevant would be scalability regarding display size and software scalability. How these types of scalability can be addressed, however, is subject of future work.



# References

- Aila, T. and Laine, S. (2009). Understanding the efficiency of ray traversal on gpus. In Proceedings of the Conference on High Performance Graphics 2009, HPG '09, pages 145–149, New York, NY, USA. ACM.
- Akimoto, T., Mase, K., and Suenaga, Y. (1991). Pixel-selected ray tracing. IEEE Computer Graphics and Applications, 11(4):14–22.
- Andrienko, N. and Andrienko, G. (2006). Exploratory analysis of spatial and temporal data: a systematic approach. Springer Science & Business Media.
- Arthur, J. J., Kramer, L. J., and Bailey, R. E. (2005). Flight test comparison between enhanced vision (flir) and synthetic vision systems. In Proceedings Volume 5802, Enhanced and Synthetic Vision 2005, volume 5802, pages 5802 – 5802 – 12.
- Aspert, N., Santa-Cruz, D., and Ebrahimi, T. (2002). Mesh: Measuring errors between surfaces using the hausdorff distance. In Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on, volume 1, pages 705–708. IEEE.
- Beck, M. (2003). Real-time visualization of big 3d city models. In Gruen, A., Murai, S., Niederoest, J., and Remondino, F., editors, International Archives of the Photogrammetry Sensing and Spatial Information Sciences, volume 34.
- Beecham, R., Rooney, C., Meier, S., Dykes, J., Slingsby, A., Turkay, C., Wood, J., and Wong, B. L. W. (2016). Faceted views of varying emphasis (favves): a framework for visualising multi-perspective small multiples. Computer Graphics Forum, 35(3):241–249.
- Belhadj, F. (2007). Terrain modeling: a constrained fractal model. In Proceedings of the 5th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa, pages 197–204. ACM.
- Bertin, J. (1983). Semiology of Graphics. University of Wisconsin Press.
- Bolton, M. L., Bass, E. J., and Comstock, J. R. (2007). Spatial awareness in synthetic vision systems: Using spatial and temporal judgments to evaluate texture and field of view. Human Factors: The Journal of the Human Factors and Ergonomics Society, 49(6):961–974.
- Bonaventura, X., Sima, A. A., Feixas, M., Buckley, S. J., Sbert, M., and Howell, J. A. (2017). Information measures for terrain visualization. Computers & Geosciences, 99:9 – 18.

- Borji, A. and Itti, L. (2013). State-of-the-art in visual attention modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(1):185–207.
- Borji, A., Sihite, D. N., and Itti, L. (2013). Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. IEEE Transactions on Image Processing, 22(1):55–69.
- Breuel, F. (2013). Visualizing multivariate data along trajectories. Master’s thesis, University of Rostock, Rostock, Germany.
- Brodlie, K., Osorio, R. A., and Lopes, A. (2012). A review of uncertainty in data visualization. In Expanding the Frontiers of Visual Analytics and Visualization, pages 81–109. Springer.
- Brown, R. (2004). Animated visual vibrations as an uncertainty visualisation technique. In Proceedings of the 2Nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, GRAPHITE ’04, pages 84–89, New York, NY, USA. ACM.
- Bruce, N. D. B. and Tsotsos, J. K. (2005). Saliency based on information maximization. In Proceedings of the 18th International Conference on Neural Information Processing Systems, NIPS’05, pages 155–162, Cambridge, MA, USA. MIT Press.
- Carpendale, M. (2003). Considering visual variables as a basis for information visualisation. Technical report, University of Calgary, Department of Computer Science.
- Cockburn, A. (2004). Revisiting 2d vs 3d implications on spatial memory. In Proceedings of the Fifth Conference on Australasian User Interface - Volume 28, AUIC ’04, pages 25–31, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.
- Cockburn, A., Karlson, A., and Bederson, B. B. (2009). A review of overview+detail, zooming, and focus+context interfaces. ACM Computing Surveys, 41(1):2:1–2:31.
- Cockburn, A. and McKenzie, B. (2002). Evaluating the effectiveness of spatial memory in 2d and 3d physical and virtual environments. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’02, pages 203–210, New York, NY, USA. ACM.
- Cook, K. A. and Thomas, J. J. (2005). Illuminating the Path: The Research and Development Agenda for Visual Analytics. IEEE Computer Society, Los Alamitos, CA, United States.
- Cook, R. L., Porter, T., and Carpenter, L. (1984). Distributed ray tracing. SIGGRAPH Comput. Graph., 18(3):137–145.
- Cox, J., House, D., and Lindell, M. (2013). Visualizing uncertainty in predicted hurricane tracks. International Journal for Uncertainty Quantification, 3(2):143–156.
- Davis, T. J. and Keller, C. (1997). Modelling and visualizing multiple spatial uncertainties. Computers & Geosciences, 23(4):397 – 408. Exploratory Cartographic Visualisation.
- DeCarlo, D., Finkelstein, A., Rusinkiewicz, S., and Santella, A. (2003). Suggestive contours for conveying shape. ACM Transactions on Graphics (Proc. SIGGRAPH), 22(3):848–855.

- 
- Deitrick, S. and Edsall, R. (2006). The Influence of Uncertainty Visualization on Decision Making: An Empirical Evaluation, chapter 45, pages 719–738. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Dick, C., Krüger, J., and Westermann, R. (2009). GPU ray-casting for scalable terrain rendering. In Proceedings of Eurographics 2009 - Areas Papers, pages 43–50.
- Doleisch, H., Gasser, M., and Hauser, H. (2003). Interactive feature specification for focus+context visualization of complex simulation data. In Proceedings of the Symposium on Data Visualisation 2003, VISSYM '03, pages 239–248, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.
- Doneus, M. (2013). Openness as visualization technique for interpretative mapping of airborne lidar derived digital terrain models. Remote Sensing, 5(12):6427–6442.
- Dowson, K. (1994). Towards extracting artistic sketches and maps from digital elevation models. PhD thesis, University of Hull.
- Draper, D. (1995). Assessment and propagation of model uncertainty. Journal of the Royal Statistical Society. Series B (Methodological), 57(1):45–97.
- Drecki, I. (2002). Visualisation of uncertainty in geographical data. Spatial data quality, pages 140–159.
- Dübel, S., Middendorf, L., Haubelt, C., and Schumann, H. (2015). A flexible architecture for ray tracing terrain heightfields. Proceedings of the International Summerschool on Visual Computing.
- Dübel, S., Röhlig, M., Schumann, H., and Trapp, M. (2014). 2d and 3d presentation of spatial data: A systematic review. In 3DVis (3DVis), 2014 IEEE VIS International Workshop on, pages 11–18.
- Dübel, S., Röhlig, M., Tominksi, C., and Schumann, H. (2017). Visualizing 3d terrain, geo-spatial data, and uncertainty. Informatics, 4(1).
- Dübel, S. and Schumann, H. (2017). Visualization of features in 3d terrain. ISPRS International Journal of Geo-Information, 6(11):357:1–357:20.
- Duchaineau, M., Wolinsky, M., Sigeti, D. E., Miller, M. C., Aldrich, C., and Mineev-Weinstein, M. B. (1997). Roaming terrain: Real-time optimally adapting meshes. In Proceedings. Visualization '97 (Cat. No. 97CB36155), pages 81–88.
- Egeth, H. E. and Yantis, S. (1997). Visual attention: Control, representation, and time course. Annual Review of Psychology, 48(1):269–297. PMID: 9046562.
- Eick, S. G. and Karr, A. F. (2002). Visual scalability. Journal of Computational and Graphical Statistics, 11(1):22–43.
- Endsley, M. R., Farley, T. C., Jones, W. M., Midkiff, A. H., and Hansman, R. J. (1998). Situation awareness information requirements for commercial airline pilots. Technical report, International Center for Air Transportation.
- Fowler, R. J. and Little, J. J. (1979). Automatic extraction of irregular network digital terrain models. SIGGRAPH Comput. Graph., 13(2):199–207.

- Frank, A. U. and Timpf, S. (1994). Multiple representations for cartographic objects in a multi-scale tree—an intelligent graphical zoom. *Computers & Graphics*, 18(6):823–829.
- Frintrop, S. (2006). *VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search (Lecture Notes in Computer Science / Lecture Notes in Artificial Intelligence)*. Springer-Verlag, Berlin, Heidelberg.
- Fuchs, G. and Schumann, H. (2004). Visualizing abstract data on maps. In *Proceedings. Eighth International Conference on Information Visualisation, 2004. IV 2004.*, pages 139–144.
- Gahegan, M. and Ehlers, M. (2000). A framework for the modelling of uncertainty between remote sensing and geographic information systems. *ISPRS Journal of Photogrammetry and Remote Sensing*, 55(3):176 – 188.
- Gao, D. and Vasconcelos, N. (2005). Discriminant saliency for visual recognition from cluttered scenes. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*, pages 481–488. MIT Press.
- Garcia-Diaz, A., Fdez-Vidal, X. R., Pardo, X. M., and Dosil, R. (2012). Saliency from hierarchical adaptation through decorrelation and variance normalization. *Image and Vision Computing*, 30(1):51 – 64.
- Garland, M. and Heckbert, P. S. (1995). Fast polygonal approximation of terrains and height fields. Technical report, Carnegie Mellon University.
- Garland, M. and Heckbert, P. S. (1997). Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216. ACM Press/Addison-Wesley Publishing Co.
- Gershon, N. (1998). Visualization of an imperfect world. *IEEE Computer Graphics and Applications*, 18(4):43–45.
- Glassner, A. S. (1989). *An introduction to ray tracing*. Elsevier.
- Goodchild, M., Battenfield, B., and Wood, J. (1994). On introduction to visualizing data validity. *Visualization in geographical information systems*, pages 141–149.
- Govindaraju, V., Djeu, P., Sankaralingam, K., Vernon, M., and Mark, W. R. (2008). Toward a multicore architecture for real-time ray-tracing. In *Proceedings of the 41st MICRO*.
- Griethe, H. and Schumann, H. (2006). The visualization of uncertain data: Methods and problems. In *Proceedings of simulation and visualization '06*, pages 143–156. SCS Publishing House.
- Grigoryan, G. and Rheingans, P. (2002). Probabilistic surfaces: Point based primitives to show surface uncertainty. In *Proceedings of the conference on Visualization'02*, pages 147–154. IEEE Computer Society.
- Gross, M. H., Gatti, R., and Staadt, O. (1995). Fast multiresolution surface meshing. In *Proceedings of the 6th Conference on Visualization '95, VIS '95*, pages 135–, Washington, DC, USA. IEEE Computer Society.

- 
- Guo, H., Huang, J., and Laidlaw, D. H. (2015). Representing uncertainty in graph edges: An evaluation of paired visual variables. IEEE Trans. Vis. Comput. Graph., 21(10):1173–1186.
- Haber, R. B. and McNabb, D. A. (1990). Visualization idioms: A conceptual model for scientific visualization systems. Visualization in scientific computing. Silver Spring MD: IEEE Computer Society Press, pages 74–93.
- Hägerstrand, T. (1970). What about people in regional science? Papers in regional science, 24(1):7–24.
- Hahmann, S. and Burghardt, D. (2013). How much information is geospatially referenced? networks and cognition. International Journal of Geographical Information Science, 27(6):1171–1189.
- Hajirahimova, M. S. and Aliyeva, A. S. (2017). About big data measurement methodologies and indicators. International Journal of Modern Education and Computer Science, 9(10):1–9.
- Hall, K. W., Perin, C., Kusalik, P. G., Gutwin, C., and Carpendale, S. (2016). Formalizing emphasis in information visualization. Computer Graphics Forum, 35(3):717–737.
- Han, B. and Zhou, B. (2007). High speed visual saliency computation on gpu. In 2007 IEEE International Conference on Image Processing, volume 1, pages I – 361–I – 364.
- Hanrahan, P. (2011). The future of visual analytics. Talk on Visual Computing Trends 2011.
- Harel, J., Koch, C., and Perona, P. (2006). Graph-based visual saliency. In Proceedings of the 19th International Conference on Neural Information Processing Systems, NIPS’06, pages 545–552, Cambridge, MA, USA. MIT Press.
- Hertzmann, A. (1999). Introduction to 3d non-photorealistic rendering: Silhouettes and outlines. Non-Photorealistic Rendering. SIGGRAPH, 99(1).
- Hertzmann, A. and Zorin, D. (2000). Illustrating smooth surfaces. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’00, pages 517–526, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- Hesse, R. (2010). Lidar-derived local relief models—a new tool for archaeological prospection. Archaeological prospection, 17(2):67–72.
- Hirt, C. (2014). Digital Terrain Models, pages 1–6. Springer International Publishing, Cham.
- Hobbs, F. (1995). The rendering of relief images from digital contour data. The Cartographic Journal, 32(2):111–116.
- Hoffman, N., Preetham, A. J., et al. (2002). Rendering outdoor light scattering in real time. In Proceedings of Game Developer Conference, volume 2002, pages 337–352.
- Hoppe, H. (1997). View-dependent refinement of progressive meshes. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pages 189–198. ACM Press/Addison-Wesley Publishing Co.

- Hou, X. and Zhang, L. (2007). Saliency detection: A spectral residual approach. In 2007 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8.
- Howard, D. and MacEachren, A. M. (1996). Interface design for geographic visualization: Tools for representing reliability. Cartography and Geographic Information Systems, 23(2):59–77.
- Itti, L. and Baldi, P. (2009). Bayesian surprise attracts human attention. Vision Research, 49(10):1295 – 1306. Visual Attention: Psychophysics, electrophysiology and neuroimaging.
- Itti, L. and Koch, C. (2000). A saliency-based search mechanism for overt and covert shifts of visual attention. Vision Research, 40(10):1489 – 1506.
- Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(11):1254–1259.
- Ize, T., Wald, I., Robertson, C., and Parker, S. G. (2006). An evaluation of parallel grid construction for ray tracing dynamic scenes. In 2006 IEEE Symposium on Interactive Ray Tracing, pages 47–55.
- J. Hunter, G. and F. Goodchild, M. (1993). Managing uncertainty in spatial databases: putting theory into practice. Journal of Urban and Regional Information Systems Association, 5(2):55–62.
- Jankun-Kelly, T. J. and Ma, K.-L. (2003). Moiregraphs: radial focus+context visualization and interaction for graphs with visual nodes. In IEEE Symposium on Information Visualization 2003 (IEEE Cat. No.03TH8714), pages 59–66.
- Johnson, C. R. and Sanderson, A. R. (2003). A next step: Visualizing errors and uncertainty. Computer Graphics and Applications, IEEE, 23(5):6–10.
- Judd, T., Ehinger, K., Durand, F., and Torralba, A. (2009). Learning to predict where humans look. In 2009 IEEE 12th International Conference on Computer Vision, pages 2106–2113.
- Judzinsky, N. (2016). Using visual cues to facilitate visual perception of trajectories in 3d terrain. Master’s thesis, University of Rostock.
- Kajiya, J. T. (1986). The rendering equation. SIGGRAPH Comput. Graph., 20(4):143–150.
- Kardos, J., Benwell, G., and Moore, A. (2005). The visualisation of uncertainty for spatially referenced census data using hierarchical tessellations. Transactions in GIS, 9(1):19–34.
- Kehrer, J. and Hauser, H. (2013). Visualization and visual analysis of multifaceted scientific data: A survey. IEEE Transactions on Visualization and Computer Graphics, 19(3):495–513.
- Keim, D., Kohlhammer, J., Ellis, G., and Mansmann, F. (2010). Mastering the Information Age Solving Problems with Visual Analytics. Eurographics Association.
- Keim, D. A. (2002). Information visualization and visual data mining. IEEE Transactions on Visualization & Computer Graphics, 8:1–8.

- Keller, P. R., Keller, M. M., Markel, S., Mallinckrodt, A. J., and McKay, S. (1994). Visual cues: practical data visualization. *Computers in Physics*, 8(3):297–298.
- Kennelly, P. and Kimerling, A. (2006a). Non-photorealistic rendering and terrain representation. *Cartographic Perspectives*, 0(54).
- Kennelly, P. and Kimerling, A. (2006b). Non-photorealistic rendering and terrain representation. *Cartographic Perspectives*, 0(54).
- Kinkeldey, C., MacEachren, A. M., and Schiewe, J. (2014). How to assess visual communication of uncertainty? a systematic review of geospatial uncertainty visualisation user studies. *The Cartographic Journal*, 51(4):372–386.
- Klir, G. and Wierman, M. (1999). *Uncertainty-Based Information: Elements of Generalized Information Theory*, volume 15. Springer Science & Business Media.
- Kreuseler, M. (2000). Visualization of geographically related multidimensional data in virtual 3d scenes. *Computers & Geosciences*, 26(1):101 – 108.
- Kunz, M., Grêt-Regamey, A., and Hurni, L. (2011). Visualization of uncertainty in natural hazards assessments using an interactive cartographic information system. *Natural hazards*, 59(3):1735–1751.
- Lafortune, E. P. and Willems, Y. D. (1993). Bi-directional path tracing. In *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*, pages 145–153, Alvor, Portugal.
- Lang, C., Nguyen, T. V., Katti, H., Yadati, K., Kankanhalli, M., and Yan, S. (2012). Depth matters: Influence of depth cues on visual saliency. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C., editors, *Computer Vision – ECCV 2012*, pages 101–115, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Lee, C.-H. and Shin, Y. G. (1997). A terrain rendering method using vertical ray coherence. *The Journal of Visualization and Computer Animation*, 8(2):97–114.
- Lee, W.-J., Shin, Y., Lee, J., Kim, J.-W., Nah, J.-H., Jung, S., Lee, S., Park, H.-S., and Han, T.-D. (2013). Sgrrt: A mobile gpu architecture for real-time ray tracing. In *Proceedings of the 5th High-Performance Graphics Conference, HPG '13*, pages 109–119, New York, NY, USA. ACM.
- Li, Y., Zhou, Y., Yan, J., Niu, Z., and Yang, J. (2010). Visual saliency based on conditional entropy. In Zha, H., Taniguchi, R.-i., and Maybank, S., editors, *Computer Vision – ACCV 2009*, pages 246–257, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L. F., Faust, N., and Turner, G. A. (1996). Real-time, continuous level of detail rendering of height fields. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '96*, pages 109–118, New York, NY, USA. ACM.
- Losasso, F. and Hoppe, H. (2004). Geometry clipmaps: Terrain rendering using nested regular grids. *ACM Trans. Graph.*, 23(3):769–776.

- Lu, A., Morris, C. J., Ebert, D. S., Rheingans, P., and Hansen, C. (2002). Non-photorealistic volume rendering using stippling techniques. In Proceedings of the Conference on Visualization '02, VIS '02, pages 211–218, Washington, DC, USA. IEEE Computer Society.
- Lu, M., Chen, S., Lai, C., Lin, L., and Yuan, X. (2017). Frontier of information visualization and visual analytics in 2016. Journal of Visualization, 20(4):667–686.
- Luboschik, M., Schumann, H., and Cords, H. (2008). Particle-based labeling: Fast point-feature labeling without obscuring other visual features. IEEE Transactions on Visualization and Computer Graphics, 14(6):1237–1244.
- Luo, A., Kao, D., and Pang, A. (2003). Visualizing spatial distribution data sets. In Proceedings of Symposium on Data Visualization.
- MacEachren, A. M. (1992). Visualizing uncertain information. Cartographic Perspectives, pages 10–19.
- MacEachren, A. M., Brewer, C. A., and Pickle, L. W. (1998). Visualizing georeferenced data: Representing reliability of health statistics. Environment and Planning A: Economy and Space, 30(9):1547–1561.
- MacEachren, A. M., Robinson, A., Hopper, S., Gardner, S., Murray, R., Gahegan, M., and Hetzler, E. (2005). Visualizing geospatial information uncertainty: What we know and what we need to know. Cartography and Geographic Information Science, 32(3):139–160.
- MacEachren, A. M., Roth, R. E., O'Brien, J., Li, B., Swingley, D., and Gahegan, M. (2012). Visual semiotics and uncertainty visualization: An empirical study. Visualization and Computer Graphics, IEEE Transactions on, 18(12):2496–2505.
- Mack, M. L. and Oliva, A. (2004). Computational estimation of visual complexity. In The 12th annual object, perception, attention, and memory conference.
- Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. ACM Trans. Graph., 5(2):110–141.
- Mccallum, A. K. (1996). Reinforcement Learning with Selective Perception and Hidden State. PhD thesis, The University of Rochester. AAI9618237.
- Middendorf, L. (2014). Dynamic task scheduling and binding for many-core systems through stream rewriting. PhD thesis, University of Rostock.
- Munzner, T. (2014). Visualization Analysis and Design. CRC Press.
- Nah, J.-H., Kwon, H.-J., Kim, D.-S., Jeong, C.-H., Park, J., Han, T.-D., Manocha, D., and Park, W.-C. (2014). Raycore: A ray-tracing hardware architecture for mobile devices. ACM Trans. Graph., 33(5):162:1–162:15.
- Nah, J.-H., Park, J.-S., Park, C., Kim, J.-W., Jung, Y.-H., Park, W.-C., and Han, T.-D. (2011). T&i engine: Traversal and intersection engine for hardware accelerated ray tracing. ACM Trans. Graph., 30(6):160:1–160:10.

- NASA and METI (2019). Aster global digital elevation model: A product of ministry of economy, trade, and industry (meti) of japan and the united states national aeronautics and space administration (nasa).
- Newman, T. S. and Lee, W. (2004). On visualizing uncertainty in volumetric data: techniques and their evaluation. *Journal of Visual Languages & Computing*, 15(6):463–491.
- Nowrouzezahrai, D. and Snyder, J. (2009). Fast global illumination on dynamic height fields. In *Computer Graphics Forum*, volume 28, pages 1131–1139. Wiley Online Library.
- Oat, C. and Sander, P. V. (2007). Ambient aperture lighting. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 61–64. ACM.
- Ochi, Y. and Kanai, K. (1999). Automatic approach and landing for propulsion controlled aircraft by  $h_\infty$  control. In *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No. 99CH36328)*, volume 2, pages 997–1002. IEEE.
- Oliva, A., Mack, M. L., Shrestha, M., and Peeper, A. (2004). Identifying the perceptual dimensions of visual complexity of scenes. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 26.
- Osorio, R. S. A. and Brodlie, K. W. (2008). Contouring with uncertainty. In Lim, I. S. and Tang, W., editors, *Theory and Practice of Computer Graphics*. The Eurographics Association.
- Paletta, L., Fritz, G., and Seifert, C. (2005). Q-learning of sequential attention for visual object recognition from informative local descriptors. In *Proceedings of the 22Nd International Conference on Machine Learning, ICML '05*, pages 649–656, New York, NY, USA. ACM.
- Pang, A. (2001). Visualizing uncertainty in geo-spatial data. In *Proceedings of the Workshop on the Intersections between Geospatial Information and Information Technology*, pages 1–14. National Research Council Arlington, VA.
- Pang, A. T., Wittenbrink, C. M., and Lodha, S. K. (1997). Approaches to uncertainty visualization. *The Visual Computer*, 13(8):370–390.
- Parker, S. G., Bigler, J., Dietrich, A., Friedrich, H., Hoberock, J., Luebke, D., McAllister, D., McGuire, M., Morley, K., Robison, A., and Stich, M. (2010). Optix: A general purpose ray tracing engine. *ACM Trans. Graph.*, 29(4):66:1–66:13.
- Patterson, T. and Jenny, B. (2011). The development and rationale of cross-blended hypersometric tints. *Cartographic Perspectives*, 0(69).
- Peterson, J. C. and Porter, M. B. (2013). Ray/beam tracing for modeling the effects of ocean and platform dynamics. *IEEE Journal of Oceanic Engineering*, 38(4):655–665.
- Pharr, M. and Green, S. (2004). Ambient occlusion. *GPU Gems*, 1:279–292.
- Phong, B. T. (1975). Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317.

- Plumlee, M. and Ware, C. (2003). Integrating multiple 3d views through frame-of-reference interaction. In Proceedings International Conference on Coordinated and Multiple Views in Exploratory Visualization - CMV 2003 -, pages 34–43.
- Policarpo, F. and Oliveira, M. M. (2007). Relaxed cone stepping for relief mapping. GPU gems, 3:409–428.
- Policarpo, F., Oliveira, M. M., and Comba, J. a. L. D. (2005). Real-time relief mapping on arbitrary polygonal surfaces. In Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games, I3D '05, pages 155–162, New York, NY, USA. ACM.
- Potter, K., Rosen, P., and Johnson, C. R. (2012). From quantification to visualization: A taxonomy of uncertainty visualization approaches. In Uncertainty Quantification in Scientific Computing, pages 226–249. Springer.
- Preim, B. and Bartz, D. (2007). Visualization in Medicine - Theory, Algorithms, and Applications. The Morgan Kaufmann Series in Computer Graphics. Morgan Kaufmann, Burlington.
- Richter, C. (2016). Application of focus & context techniques for visualization of weather data embedded into 3d terrain. Master’s thesis, University of Rostock.
- Richter, C., Dübel, S., and Müller-Divéky, S. (2018). Conversations and collaborative work on flight scenarios and the required data for representation.
- Richter, C., Dübel, S., and Schumann, H. (2017). Visual Analysis of Geo-spatial Data in 3D Terrain Environments using Focus+Context. In Sedlmair, M. and Tominski, C., editors, EuroVis Workshop on Visual Analytics (EuroVA). The Eurographics Association.
- Richter, C., Luboschik, M., Dübel, S., and Müller-Divéky, S. (2019). A method for generating a 2d image of a 3d surface. Pending Patent Application: US20180342088A1.
- Ripken, C. (2019). Using saliency information for priority-based information visualization in 3d terrain. Master’s thesis, University of Rostock.
- Robertson, G., Ebert, D., Eick, S., Keim, D., and Joy, K. (2009). Scale and complexity in visual analytics. Information Visualization, 8(4):247–253.
- Robinson, A. C. (2011). Highlighting in geovisualization. Cartography and Geographic Information Science.
- Röhlig, M. (2014). Flexible visualization of weather data in a spatial context with regard to uncertainty. Master’s thesis, University of Rostock.
- Röhlig, M., Luboschik, M., and Schumann, H. (2017). Visibility widgets for unveiling occluded data in 3d terrain visualization. Journal of Visual Languages & Computing, 42(Supplement C):86 – 98.
- Rosenholtz, R. (1999). A simple saliency model predicts a number of motion popout phenomena. Vision Research, 39(19):3157 – 3163.
- Rosenholtz, R., Dorai, A., and Freeman, R. (2011). Do predictions of visual perception aid design? ACM Transactions on Applied Perception (TAP), 8(2):12:1–12:20.

- Rosenholtz, R., Li, Y., Mansfield, J., and Jin, Z. (2005). Feature congestion: A measure of display clutter. In *In Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 761–770. ACM.
- Rosenholtz, R., Li, Y., and Nakano, L. (2007). Measuring visual clutter. *Journal of Vision*, 7(2):17–17.
- Rossignac, J. and Borrel, P. (1993). Multi-resolution 3d approximations for rendering complex scenes. In Falcidieno, B. and Kunii, T. L., editors, *Modeling in Computer Graphics*, pages 455–465, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Ruzinoor, C. M., Shariff, A. R. M., Pradhan, B., Ahmad, M. R., and Rahim, M. S. M. (2012). A review on 3d terrain visualization of gis data: techniques and software. *Geo-spatial Information Science*, 15(2):105–115.
- Sakalauskas, T. (2007). Silhouette partitioning for height field ray tracing. *Proceedings of the 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 73–80.
- Sanyal, J., Zhang, S., Dyer, J., Mercer, A., Amburn, P., and Moorhead, R. J. (2010). Noodles: A tool for visualization of numerical weather model ensemble uncertainty. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1421–1430.
- Schmidt, G., Chen, S.-L., Bryden, A., Livingston, M., Rosenblum, L., and Osborn, B. (2004). Multidimensional visual representations for underwater environmental uncertainty. *Computer Graphics and Applications, IEEE*, 24(5):56–65.
- Schroeder, W. J., Zarge, J. A., and Lorensen, W. E. (1992). Decimation of triangle meshes. *SIGGRAPH Comput. Graph.*, 26(2):65–70.
- Schumann, H. and Müller, W. (2000). *Visualisierung - Grundlagen und allgemeine Methoden*. Springer-Verlag Berlin Heidelberg.
- Seiler, L., Carmean, D., Sprangle, E., Forsyth, T., Abrash, M., Dubey, P., Junkins, S., Lake, A., Sugerman, J., Cavin, R., Espasa, R., Grochowski, E., Juan, T., and Hanrahan, P. (2008). Larrabee: A many-core x86 architecture for visual computing. In *ACM SIGGRAPH 2008 Papers, SIGGRAPH '08*, pages 18:1–18:15, New York, NY, USA. ACM.
- Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343.
- Sloan, P.-P., Kautz, J., and Snyder, J. (2002). Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph.*, 21(3):527–536.
- Stamminger, M., Haber, J., Schirmacher, H., and Seidel, H.-P. (2000). Walkthroughs with corrective texturing. In Péroche, B. and Rushmeier, H., editors, *Rendering Techniques 2000*, pages 377–388, Vienna. Springer Vienna.
- Stevens, K. A. (1980). Surface perception from local analysis of texture and contour. Technical report, Massachusetts, Institut of technology.

- Tevs, A., Ihrke, I., and Seidel, H.-P. (2008). Maximum mipmaps for fast, accurate, and scalable dynamic height field rendering. In Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games, I3D '08, pages 183–190, New York, NY, USA. ACM.
- Thomson, J., Hetzler, E., MacEachren, A., Gahegan, M., and Pavel, M. (2005). A typology for visualizing uncertainty. Proceedings of Spie, 5669:5669 – 5669 – 12.
- Tominski, C., Gladisch, S., Kister, U., Dachsel, R., and Schumann, H. (2014). A survey on interactive lenses in visualization. In EuroVis State-of-the-Art Reports, pages 43–62. Eurographics Association.
- Tominski, C., Schumann, H., Andrienko, G., and Andrienko, N. (2012). Stacking-based visualization of trajectory attribute data. IEEE Transactions on Visualization and Computer Graphics, 18(12):2565–2574.
- Toppe, R. (1987). Terrain models: a tool for natural hazard mapping. Avalanche formation, movement and effects, IAHS Publ, 162:629–638.
- Tory, M., Kirkpatrick, A. E., Atkins, M. S., and Moller, T. (2006). Visualization task performance with 2d, 3d, and combination displays. IEEE Transactions on Visualization and Computer Graphics, 12(1):2–13.
- Trapp, M. (2013). Interactive Rendering Techniques for Focus+Context Visualization of 3D Geovirtual Environments. PhD thesis, University of Potsdam.
- Trappe, J. (2014). Embedding geometric primitives into a three-dimensional terrain representation. Master’s thesis, University of Rostock.
- Treisman, A. M. and Gelade, G. (1980). A feature-integration theory of attention. Cognitive Psychology, 12(1):97 – 136.
- Ulrich, T. (2002). Rendering massive terrains using chunked level of detail control. ACM SIGGraph Course “Super-size it! Scaling up to Massive Virtual Worlds”.
- Verma, M. and McOwan, P. W. (2009). Generating customised experimental stimuli for visual search using genetic algorithms shows evidence for a continuum of search efficiency. Vision Research, 49(3):374 – 382.
- Vion-Dury, J.-Y. and Santana, M. (1994). Virtual images: Interactive visualization of distributed object-oriented systems. SIGPLAN Not., 29(10):65–84.
- Wald, I. (2005). Realtime ray tracing and interactive global illumination. PhD thesis, University of Saarland.
- Wald, I., Boulos, S., and Shirley, P. (2007). Ray tracing deformable scenes using dynamic bounding volume hierarchies. ACM Trans. Graph., 26(1).
- Wald, I., Johnson, G., Amstutz, J., Brownlee, C., Knoll, A., Jeffers, J., Gnther, J., and Navratil, P. (2017). Ospray - a cpu ray tracing framework for scientific visualization. IEEE Transactions on Visualization and Computer Graphics, 23(1):931–940.
- Wald, I. and Slusallek, P. (2001). State of the art in interactive ray tracing. State of the Art Reports, EUROGRAPHICS, 2001:21–42.

- 
- Wald, I., Slusallek, P., Benthin, C., and Wagner, M. (2001). Interactive rendering with coherent ray tracing. *Computer Graphics Forum*, 20(3):153–165.
- Wang, Y.-S. and Chi, M.-T. (2011). Focus+context metro maps. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2528–2535.
- Weinkauff, T. and Günther, D. (2009). Separatrix persistence: Extraction of salient edges on surfaces using topological methods. *Computer Graphics Forum*, 28(5):1519–1528.
- Whitted, T. (1980). An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349.
- Wittenbrink, C., Pang, A., and Lodha, S. (1996). Glyphs for visualizing uncertainty in vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):266–279.
- Wolfe, J. M. (1998). Visual search. *Attention*, pages 13–73.
- Woodruff, A., Landay, J., and Stonebraker, M. (1998). Constant information density in zoomable interfaces. In *Proceedings of the working conference on Advanced visual interfaces*, pages 57–65. ACM.
- Wu, J. and Amaratunga, K. (2003). Wavelet triangulated irregular networks. *International Journal of Geographical Information Science*, 17(3):273–289.
- Yoëli, P. (1967). The mechanisation of analytical hill shading. *The Cartographic Journal*, 4(2):82–88.
- Zakšek, K., Oštir, K., and Kokalj, Ž. (2011). Sky-view factor as a relief visualization technique. *Remote sensing*, 3(2):398–415.
- Zhang, L., Tong, M. H., Marks, T. K., Shan, H., and Cottrell, G. W. (2008). Sun: A bayesian framework for saliency using natural statistics. *Journal of vision*, 8(7):32–32.
- Zhu, B. and Chen, H. (2006). Information visualization. *Annual Review of Information Science and Technology*, 39(1):139–177.
- Zyda, M. J., Pratt, D. R., Falby, J. S., and Mackey, R. L. (1993). Npsnet: Hierarchical data structures for real-time three-dimensional visual simulation. *Computers & Graphics*.

## List of Scientific Contributions

### Journal Articles

Dübel and Schumann (2017): Dübel, S. and M.; Schumann, H. *Visualization of Features in 3D Terrain* ISPRS International Journal of Geo-Information, 2017, 6, 357:1-357:20

Dübel et al. (2017): Dübel, S.; Röhlig, M.; Tominski, C and Schumann, H. *Visualizing 3D Terrain, Geo-spatial Data, and Uncertainty* Informatics, 2017, 4

### Conference Publications

Richter et al. (2017): Richter, C.; Dübel, S. and Schumann, H. *Visual Analysis of Geo-spatial Data in 3D Terrain Environments using Focus+Context* EuroVis Workshop on Visual Analytics (EuroVA), The Eurographics Association, 2017

Dübel et al. (2015): Dübel, S. and Middendorf, Lars and Haubelt, Christian and Schumann, Heidrun *A Flexible Architecture for Ray Tracing Terrain Heightfields* Proceedings of the International Summerschool on Visual Computing, 2015

Dübel et al. (2014): Dübel, S.; Röhlig, M.; Schumann, H. and Trapp, M. *2D and 3D presentation of spatial data: A systematic review* 3DVis (3DVis), 2014 IEEE VIS International Workshop on, 2014, 11-18

### Miscellaneous Contribution

Richter et al. (2019): Richter, C.; Luboschik, M.; Dübel, S.; Müller-Divéky *A method for generating a 2D image of a 3D surface* Pending Patent Application (since 2017)

## **Statement of Originality**

This is to certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged. This thesis has not been submitted for any degree or other purposes.

## **Eidesstattliche Erklärung**

Hiermit versichere ich, dass ich die eingereichte Dissertation selbständig und ohne fremde Hilfe verfasst, andere als die von mir angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Diese Arbeit wurde noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegt und der Author sich noch an keiner Fakultät oder Universität um den Doktorgrad beworben.

Steve Dübel

December 18, 2019