Visualization Hybridization with Spatialization Cues

Thomas Trautner

Dissertation for the degree of Philosophiae Doctor (PhD)

Supervised by Stefan Bruckner

Department of Informatics
University of Bergen

2022

Scientific Environment

The work presented in this thesis was conducted as part of my PhD studies in the Visualization Group at the Department of Informatics, University of Bergen. As part of this, I was enrolled in the Research School in Information and Communication Technology (ICT). Thanks to the Research Council of Norway, my research was supported within the MetaVis project (#250133).

UNIVERSITY OF BERGEN
Faculty of Mathematics and Natural Sciences

ICT Research School In
Information and Communication Technology

The Research Council
of Norway

Acknowledgements

I would like to sincerely thank so many who have accompanied me to this point and apologize in advance if I do not name all of you. It all started with an interview for an internship at the VRVis research center while I was still at school. I remember this interview with Gerd Hesina and Robert Tobler as if it were yesterday. Thank you for giving me the opportunity to breath the wonderful air of computer graphics and visualization although I was just a pupil. Because of you, a wonderful journey began!

Thanks to the VisGroup in Bergen, who accepted me for an internship during my master's program. The potpourri of interesting and funny moments made it easy for me to apply for a PhD. Thanks to Juraj Pálenik, who started at the same time and shared the office with me. Our friendship, the same passion for sailing and beer brewing, and the contact that never broke off although I had to return to Vienna was one of the reasons for coming back to Bergen. Sharing the same sense of humor has often made me laugh to tears. Exactly this lightness in extremely stressful times has helped me to overcome difficult phases. The Norwegian sweater you gave me back then on behalf of the Visgroup is still one of my favorites and makes me smile every time I wear it. Thank you Åsmund Birkeland for paragliding and landing safely next to Brann stadion on my last day. Thanks to Andreas Lind for the unforgettable funny game evenings and thanks to Jan Byška, Ivan Kolesar, Julius Parulek, Veronika Šoltészová, and Daniel Patel with whom I always have a good time, whenever we happen to meet. Thanks to Chaoran Fan who also shared his office with me and on the rare occasions we saw each other, was a pleasant colleague too. It was always fun when we accidentally met in the office long after midnight. Your Chinese snacks and spicy noodles are the very best. Thank you Sherin Sugathan for being the most polite office neighbor one could wish for. I will definitely remember the incredibly fun dinner including your marriage tips at the ICT research school in Flåm. Thanks to Fabian Bolte, by the way, your TV is still fine and happy in my living room. Thank you Fourough Gharbalchi for the best Persian tea with saffron sugar outside Iran, and thanks to Laura Garrison for the trail runs around Bergen. Thank you Sergej Stoppel for the many helpful tips already at the beginning of my PhD, they helped me until the end. Thanks to you and Sarah for keeping our group together with your many social events, making us friends rather than colleagues.

Thank you Eric Mörth for the many adventures we have explored together. It is absolutely amazing what we have experienced, from via ferrata, fjell-ski tours with northern lights, hammock sleepovers in the wild, swimming in cold lakes and fjords, trips to the most remote regions of Scandinavia, and (although I usually do not even drink coffee) the very best espresso and chocolate mousse on your balcony. Your talent for handicraft work and manual skills are absolutely enviable, without you my e-scooter would have ceased to exist a long time ago. It was a pleasure to get to know you!

Thank you Yngve Kristiansen for not only teaching me how to cast fishing nets, gut fish, and smoke fillets, but also for extensively explaining to me how to clean fishing nets in below-zero temperatures. We had so much fun shooting clay pigeons or trying out new cooking recipes. My mouth still waters when thinking of the portobello port wine sauce! Thank you for the time you took explaining to me how to row boats, your tireless asking if I want to join playing table tennis, and the many weekends at your cabin in Mundheim. I do not think there is anything more relaxing than sitting by the fireplace late in the evening while eating blackberries and plums from your garden.

A special thank you also to Rikke Aas. Even when you were stressed, you always found time to proofread my papers, do trial runs of my user studies with your siblings, or discuss concerns about current research project. Teaching INF250 with you was so much more fun! Seeing how easily and cool you tackle complicated problems is absolutely impressive. Working with smart and curious students like you, Anders Syvertsen or Maximilian Sbardellati was the greatest pleasure for me at university!

Thanks to Hauke Bartsch, your spontaneous meetings have often showed me new perspectives that made my research even more interesting to me, but also to my reviewers. Thanks also to Noeska Smit, you impressed me both with your professional knowledge and your never-ending positive attitude. Knowing how much you work while maintaining a competent coolness is absolutely desirable. Your detailed feedback during INF358 has helped me to become a better presenter and researcher. Thanks also for every time you reminded me not to stay in the office too long, I am sure future PHDs will be grateful as well. Thank you Helwig Hauser for the many great hikes and your help with their planning. These many local and global maxima are what make Bergen such a lovely city surrounded by endless outdoor possibilities. Thank you for the many conversations, both professional and private. It was nice to be surrounded by such bright minds and you are one of the most critical, but also most knowledgeable. Without you I would not have learned so much about volcanoes, church sizes, languages, classical music, art, but also inflamed toenails or tick bites.

Thanks also to Stefan Bruckner, your ideas and creativity made me survive in the world of researchers. There are not many people who know as many sides of me as you do, from extremely euphoric when a paper was accepted to sadly discouraged when reviewers were particularly harsh. Thanks to you, we were able to loosen up the strict separation between computer graphics and visualization, and convinced visualization experts of cool rendering effects! You quickly recognized what I am interested in and how I can benefit from your expertise and experience. Thank you for teaching me how to stand on my own feet so that I can walk my own path. I do not know anyone who has as much theoretical knowledge about visualization but also practical understanding about the actual implementation as you, and that sets a desirable example for me!

Thanks to Tamara and Emanuel Paludo, Michi and Stephanie Lichtblau, Patrick Stampf, and Thomas Pemsel. Your friendship bridges even the greatest distances and every time we meet again it feels like we have never been apart. You make life worth living and I cannot imagine any better friends. I hope you will always be in my life!

My biggest thanks go to my parents, especially for the joy in your faces at the airport and not knowing which of you I want to hug first and not let go. Thank you for telling me straight away that I should do a PhD in Norway, not because you wanted to

get rid of me but because you only want the best for me, even if that means we cannot see each other that often. In the end, our phone calls probably made us chat more than if I had stayed in Vienna. Growing up with your broad interests, generous support, cosmopolitan attitude, and anticipation of what I might be interested in as child already helped me finding my way. Thank you!

Abstract

Visualization as a tool for visual processing of any underlying data has proven to be an accepted and legitimate part of the scientific reasoning process. Many different techniques help gaining new insights from captured phenomena, support the development or evaluation of hypotheses about collected data, reveal potential misconceptions or false assumptions, simplify communicating knowledge and novel findings, and enable a multitude of additional opportunities. The reason for this effectiveness is that the human visual system is ideally suited to capture and process visually encoded data. The development of visualization from a niche to an established scientific field has made a significant contribution to this success story. A large number of journals, conferences, seminars, and workshops regularly publish new results, evaluate presented approaches, and help making knowledge globally accessible. However, this large number of contributions tailored to variable user groups, the underlying data, and the wide variety of tasks that could be performed with them, emphasizes the plethora of available techniques and the resulting difficulty in choosing the most suitable visualizations.

Therefore, we investigated common data sets and analyzed typical tasks normally performed with them. Based on this, we selected well-established and most effective visualization techniques, combining them to form a hybrid representation. The goal of such a visualization hybridization was to merge advantages of individual techniques and, thereby, simultaneously eliminate their limitations. We present so-called hybrid vigors that make the underlying visualizations more widely applicable instead of either having to change required techniques sequentially, or not being able to perform certain tasks at all. Our contributions are intended to simplify the process of finding suitable visualizations for already established data sets. During our research, we focused on two-dimensional point data, depicted on the one hand as scatter plots and, on the other hand, as relationships between consecutive point such as in line charts. Our techniques can be used especially when data sets are so large, dense, and overplotted that conventional techniques reach their limits. We show that hybrid representations are well suited for combining discrete, continuous, or aggregated forms of visual representation. Our hybridizations additionally exploit spatialization cues. Such visual cues emphasize spatiality of the underlying data through shading, without having to embed the data in 3D space including its potential disadvantages. We chose this method of encoding as we consider it the most appropriate choice, given that visualization users interact naturally and preattentively with a spatial world on a daily basis.

Abstract in Norwegian

Visualisering som et verktøy for visuell prosessering av underliggende data har vist seg å være en akseptert og legitim del av den vitenskapelige argumentasjonsprosessen. Mange forskjellige teknikker bidrar til å oppnå ny innsikt fra et målt fenomen, støtter å lage eller evaluere hypoteser om innsamlet data, avslører mulige misforståelser eller uriktige antagelser, forenkler kommunikasjon av kunnskap og nye funn, i tillegg til flere andre muligheter. Grunnen til denne effektiviteten er at menneskets visuelle system er nøye tilpasset å fange opp og prosessere visuell informasjon. Utviklingen av visualisering fra en nisje til et etablert forskingsfelt har vært et betydelig bidrag til denne suksesshistorien. Et stort antall tidsskrifter, konferanser, seminarer, og workshoper publiserer regelmessig nye resultater, evaluerer presenterte tilnærminger, og bidrar til at kunnskapen er tilgjengelig verden over. Men dette store antallet bidrag rettet mot diverse brukergrupper, underliggende data, og et bredt utvalg av oppgaver som kan utføres med dem, fremhever mengden tilgjengelige teknikker og den resulterende vanskeligheten i å velge den mest passende visualiseringen.

Derfor undersøkte vi vanlige datasett og analyserte typiske oppgaver som normalt blir utført med dem. Basert på dette, valgte vi veletablerte og effektive visualiseringsteknikker, og kombinerte dem til en hybrid representasjon. Målet med en slik hybridisering av visualiseringer var å slå sammen fordelene av de individuelle teknikkene, og dermed samtidig eliminere begrensingene deres. Vi presenterer hybride vigører, som gir de underliggende visualiseringene en bredere anvendelighet i stedet for å enten måtte endre teknikk sekvensielt, eller å ikke være i stand til å utføre visse oppgave i det hele tatt. Våre bidrag er ment til å gjøre det enklere å finne passende visualiseringer for allerede etablerte datasett. Vi satte søkelys på todimensjonal punktdata som på den ene siden var representert som spredningsplott, og på den andre siden, som forhold mellom etterfølgende punkt i, for eksempel, et linjediagram. Teknikkene våre er spesielt nyttige når datasettene er så store, tette, og overplottede at vanlige teknikker når sine begrensinger. Vi viser at hybride representasjoner er passende for å kombinere diskret, kontinuerlige, eller aggregerte former av visuell representasjon. Hybridiseringene våre utnytter i tillegg spatialisering. Slike visuelle signaler fremhever spatialitet av den underliggende dataen gjennom skyggelegging, uten å måtte plassere dataen i 3D med sine mulige ulemper. Vi valgte denne kodingen siden vi betrakter den som det mest passende valget, gitt at visualiseringsbrukere interagerer naturlig og preattentivt med en 3D-verden i det daglige.

List of Papers

This thesis is based on the following publications:

(**A**) **Thomas Trautner**, Fabian Bolte, Sergej Stoppel, and Stefan Bruckner. **Sunspot Plots: Model-based Structure Enhancement for Dense Scatter Plots**. In *Computer Graphics Forum (EuroVis 2020)*, 39(3), pages 551–563, 2020. doi: 10.1111/cgf.14001

(**B**) **Thomas Trautner**, Maximilian Sbardellati, Sergej Stoppel, and Stefan Bruckner. **Honeycomb Plots: Visual Enhancements for Hexagonal Maps**. Accepted for publication at the *Symposium on Vision, Modeling, and Visualization (VMV 2022) in Konstanz, Germany.*

(**C**) **Thomas Trautner** and Stefan Bruckner. **Line Weaver: Importance-Driven Order Enhanced Rendering of Dense Line Charts**. In *Computer Graphics Forum (EuroVis 2021)*, 40(3), pages 399–410, 2021. doi: 10.1111/cgf.14316

The following paper corresponds to my master's thesis which was published during my doctoral studies, but it is not part of the present thesis per se:

(**1**) Jan Byška, **Thomas Trautner**, Sérgio M. Marques, Jiří Damborský, Barbora Kozlíková, and Manuela Waldner. **Analysis of Long Molecular Dynamics Simulations Using Interactive Focus+Context Visualization**. In *Computer Graphics Forum (EuroVis 2019)*, 38(3), pages 441–453, 2019. doi: doi:10.1111/cgf.13701

The manuscripts presented in this thesis were written during my PhD studies, supervised by Stefan Bruckner who significantly contributed to the realization and publication of the scientific work, providing advice and guidance.

Paper **A** was coauthored by Fabian Bolte, who supported the implementation/ evaluation of the user study, and Sergej Stoppel, who improved the text of the paper.

Paper **B** was coauthored by Maximilian Sbardellati, who supported the implementation, and Sergej Stoppel, who helped evaluating the user study.

Contents

Part I

Overview

Chapter 1

Introduction

Visualization—what began about 30,000 to 65,000 years ago with the first man-made images in the form of stone-age cave paintings, has developed into its own scientific field. At that time, visual representations were used to, e.g., represent hunting scenes or mystical depictions of spiritual occurrences. They were either painted with sand, charcoal, blood, and similar, or engraved in stone using simple tools. Both visualization techniques and tools have, nevertheless, evolved over centuries and now rely on faster, more powerful, and fully customizable high-performance computing hardware.

However, the visualization knowledge gained over centuries, driven by innumerable research results and a multitude of publications, has also led to a vast number of wide-ranging visualization techniques, each of them being ideal for the specific scope they were designed, implemented, and tested for. Unfortunately, this also means that there is not necessarily a unique visualization technique available for any given use case, as they are often tailor-made for the underlying *data*, targeted *users*, and *tasks* to be performed [118]. We have, therefore researched heuristics that generate the most suitable hybrid visualizations, combining multiple visual representations, based on typical tasks derived from common data types. The resulting hybrids, however, are not constant but their visual representation can vary smoothly according to the underlying data.

1.1 Problem Statement

The requirement for the best possible choice of visual representation is highly dependent on target users and tasks for which it is intended. If both, i.e., users and tasks, can be defined accurately, it might be trivial to choose the most suitable visualization technique. If, however, at least one of the two, i.e., users or tasks, are too broad or difficult to narrow down, several different visualization techniques may be required. The user is, therefore, forced to either choose the most suitable but potentially insufficient approach, or has to tediously switch back and forth between various visualization techniques in order to be able to gain insights from several techniques sequentially.

As an alternative, we have investigated whether distinct visualizations can be meaningfully combined, resulting in what we call **visualization hybridizations**. If multiple visualization techniques are required, quantifiable data aspects can be examined with a heuristic, i.e., through an *importance function*, which specifically decides the type of visual representation. The potential advantage of displaying hybrids is to be able to

explore various data aspects simultaneously through diverse representations, originally developed for different task-dependent purposes or users. Visualization hybridization is broadly applicable, opening up unexploited possibilities ranging over diverse aspects, such as:

- **Data:** Being able to combine different data-specific encodings, i.e., visual primitives, such as points, lines, convex/concave polygons, surfaces, etc.
- **Task:** Being able to perform tasks that would otherwise require different visualization techniques simultaneously, e.g., density estimation and outlier detection.
- **User:** Being able to use visualizations for communication with different target users, e.g., novice users, professionals, domain experts, and so forth.
- **Technique:** Being able to apply different mathematical methods, e.g., discrete, continuous, or aggregated methods, for estimation or calculation simultaneously.
- **Scientific Area:** Being able to expand approaches inspired by other research, e.g., exploiting rendering techniques from computer graphics in visualization.

As part of this thesis, we have explored a fraction of this untapped potential. Unfortunately, however, hybridizing different visualization aspects is neither trivial nor free of costs. The type of hybridization may influence the effectiveness of the resulting visualization, for example, whether two visual stimuli are simply juxtaposed, superimposed, overloaded, or nested. Potential changes in the encoding may also lead to hybrids that are more difficult to interpret and understand, for example, because of an increased cognitive workload. Additionally, hybridization may further increase the initial time required to understand and apply a learned visual encoding, since several techniques have to be interpreted simultaneously. The aim of this thesis was, therefore, not only to develop and implement novel visualization hybridization techniques, but also to verify and evaluate supposed advantages over potential disadvantages.

1.2 Scope and Contributions

There is a sheer endless number of possible, in principle, arbitrary approaches of how hybridization could be implemented. We, therefore, chose to explore visual cues that may potentially be easier to understand and interpret than others. Since we, as humans, inevitably live in and interact with a three-dimensional world, we chose to explore how visualization hybridization could benefit from **spatialization cues**. Unlike the trend that started when computers were finally able to display 3D content, however, we do not want to naively embed visualizations into three-dimensional space, especially if they themselves or their data are not 3D per se. This could tend to be disadvantageous since it, i.a., requires user interaction for navigation and creates perspective and view-dependent occlusions. Instead, we systematically selected cues which correspond to spatial hints that are orthogonal to the basic encoding of the underlying data and, therefore, avoid typical drawbacks of 3D visualizations. Our initial approach was to analyze challenges and task variations that can be performed on underlying data, and to use these as potential control mechanisms, i.e., heuristics, for the hybridization.

In order to show whether our considerations are expedient, we initially selected visual representations from a wide range of well-established and commonly used visualization techniques. The analysis of their disadvantages, i.e., tasks that would require different visual representations, then provides an implicit guide to which other techniques are needed in order to make the resulting hybridization a so-called *hybrid vigor*. Similar to genetic engineering in agriculture, a hybrid vigor may combine several positive traits of different plants species, such as resistance to drought or flooding, and a large variety of flowers, fruits, or seeds. Our goal is to transfer this hybrid vigor metaphor to visualization. Our publications relate to several research areas, e.g., conceptual, technical, and empirical contributions, and can be summarized as follows:

1. We propose a novel kernel density-based visualization technique, hybridizing discrete and continuous representations of large bivariate point data.

2. We demonstrate that such density-based hybridizations are extendable to spatial point aggregations through binning, for example, using hexagonal grids.

3. We introduce additional visual encodings for hybrid aggregation plots, which (a) serve as subtle aid to perceive color differences between neighboring tiles, (b) encode the regression plane of underlying bin densities, and (c) blend underlying point data depending on the bin density with colored hexagonal tiles.

4. We propose spatialization cues for line-based data using importance functions. Our hybrid blending and weaving approach exploits ordering as well as quantitative importance, which can either be part of the underlying data or derived from geometric data properties, such as length or complexity of lines.

5. We develop a straightforward algorithm for deriving importance functions for grouped line data that reduces occlusion and thus improves visibility.

6. We demonstrate that hybridization using a heuristic, i.e., importance function, is well suited for user interaction. For example, using a magic lens which can either be used to locally change the bandwidth of a density estimation, or to highlight individual line bundles by local reordering, e.g., pulling them forward.

7. We verify that all our techniques for high-quality rendering of points, lines, and grids can be implemented efficiently on modern GPU architectures. Through a detailed analysis, we show that the frame rates of our hybridizations remain interactive even with large amounts of data, containing thousands of items.

8. We analyze perceptual properties of visualization hybridizations through detailed user studies. We examine a wide variety of visual stimuli, e.g., classic scatter plots, color-coded scatter plots, heat maps, (un-)shaded hybrids combining scatter plots and heat maps, (un-)shaded hexagonal aggregation plots, and binned hybrids exploiting blending or glyph encodings, based on common and frequently performed locate, explore, lookup, and brows tasks [22].

9. We find that shading—used with care—provides meaningful shape, i.e., spatialization, cues for diverse visualizations, increasing their information content.

1.3 Thesis Structure

This thesis consists of two main parts: Part I provides an overview of the research carried out as part of my PhD studies and Part II contains the resulting paper publications. The format of the papers, however, was adjusted to fit the layout of this thesis and all bibliographies were merged into a single unified bibliography at the end.

Part I—Overview—is structured as follows: Chapter 1 introduces visualization hybridization including potential problems and explains how our contributions address them. Chapter 2 provides the current state of the art in visualization hybridization including previous work and how our contributions differ from or expand it. Additionally, it describes visualization techniques that rely on spatialization cues, and concludes with relevant work from shape perception. Chapter 3 outlines the contribution of this thesis and Chapter 4 concludes the first part by highlighting future research opportunities.

Part II—Included Papers—contains the three publications corresponding to our contributions in full detail and describes how they were evaluated qualitatively, i.e., in the context of use cases, but also quantitatively, i.e., through user studies:

Paper **A** *sunspot plots*, paper **B** *honeycomb plots*, and paper **C** *line weaver*.

Chapter 2

Related Work

The following chapter discusses approaches related to our contribution, such as the general functioning of composite views. This relatively broad term refers to techniques whose overall goal it is to combine multiple visualization techniques into one visualization. Subsequently, we outline a specific approach of this research area namely what we refer to as visualization hybridization. The main distinguishing aspect here is the presence of an additional mechanic, e.g., heuristic, that drives this composition and defines especially *how* and *where* the individual visualizations are combined. Using representative examples, we discuss how composition concepts become hybrid representations through heuristics. We then continue with visual encodings that are capable of enhancing and augmenting the resulting hybrids in an orthogonal way, i.e., that does not conflict with the previous encodings while being informative and remaining understandable to the viewer. Therefore, we decided to explore spatialization cues and how these can be used to, on the one hand, make visualizations more expressive and, on the other, increase their information content. Shading is an established and effective approach to implement shape cues which has also been examined in perception research. We, therefore, close this chapter with excerpts from shape perception literature.

In summary, Section 2.1 introduces the broader foundation of our research, namely composite views; Section 2.2 presents related work and explores representative examples from visualization hybridization relying on heuristics; Section 2.3 focuses on approaches that exploit spatialization cues to generate expressive visualizations; and Section 2.4 analyzes the human visual system's perception of shape cues from shading.

2.1 Composite Views

The term "composite views", i.e., the combination of multiple visualizations, covers a larger number of related contributions, also referred to as *visual multiplexing*, *hybrid mixture*, *composite/nested/embedded/integrated visualizations*, *interleaving*, *dual representations* referring to two underlying visualization techniques, *multi-layered visualization* corresponding to a higher number of combined visual stimuli, etc.

Javed and Elmqvist [87] highlight the necessity of composite visualizations due to the high complexity, large scale, and heterogeneity of data sets nowadays. They, therefore, explore its design space and define four (plus one) general strategies, so-called "composite visualization view (CVV) design patterns". Their foundation for composition relies on spatial mapping on the one hand and on data relationships between

the underlying visualizations on the other. Their first strategy, *juxtaposed views*, positions two or more visualizations side by side within a single view. This corresponds to the simplest and easiest approach to implement, especially for heterogeneous data usually requiring fundamentally different visualizations. Juxtaposition, however, is purely implicit and relationships are potentially hard to understand without advanced user interaction, for example, brushing and linking. Second, *superimposed views* refer to two or more overlaid visualizations. Multiple visualizations can, therefore, be directly compared in the same underlying spatial domain. This, however, requires all data sets to originate from the same spatial domain, and may also implicitly lead to occlusions. The hybrid visualizations in our contribution correspond to a further development of superimposed views. Our goal was to reduce clutter and overplotting by using typical tasks performed by users as driving factor for the heuristics that combine the visual representations. Third, *overloaded views* correspond to individual areas, e.g., unused canvas space, of one representation which are then used by another visualization instead. Here, data sets can originate from different spatial domains, which consequently allows for a wider range of clutter prevention mechanisms. Fourth, *nested views* refer to additional visualizations either inside visual primitives of another visualization, or even as their replacement. As a result, the visualization remains compact but may become more difficult to interpret. Fifth, i.e, the plus-one strategy, *integrated views* refers to augmented juxtapositions wherein additional glyphs, e.g., arrows or lines, are used to create explicit links between the visual stimuli. This, however, may again result in clutter or overplotting. Figure 2.1 illustrates all four main strategies.

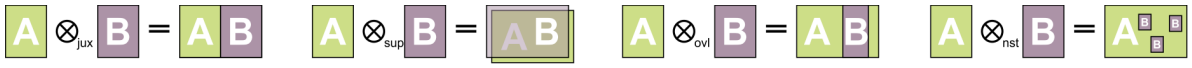

Figure 2.1: Overview of four composition strategies according to Javed and Elmqvist [87]. From left to right: juxtaposition, superimposition, overloading, and nesting.

Schulz and Hadlak [142] propose presets, a visualization approach combining already existing techniques through deformation. Their approach relies on numerical parameters, e.g., sliders, that are used to interpolate between different visual designs. This way, inexperienced users can select the most adequate visualization by choosing from a range of interpolation steps, for example presented in a grid layout. Experts, on the contrary, can actively change individual slider parameters in order to be able to precisely configure the influence of the most diverse visualization techniques. Examples of both approaches, i.e., for (a) novices and (b) experts, are shown in Figure 2.2.

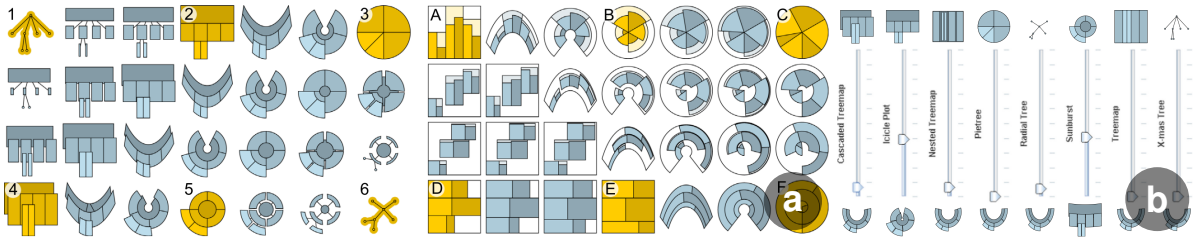

Figure 2.2: Examples of (a) a palette showing already interpolated candidates embedded in a grid, and (b) a mixer relying on numeric slider values that can be used to combine different visualization techniques through continuous deformations by Schulz and Hadlak [142].

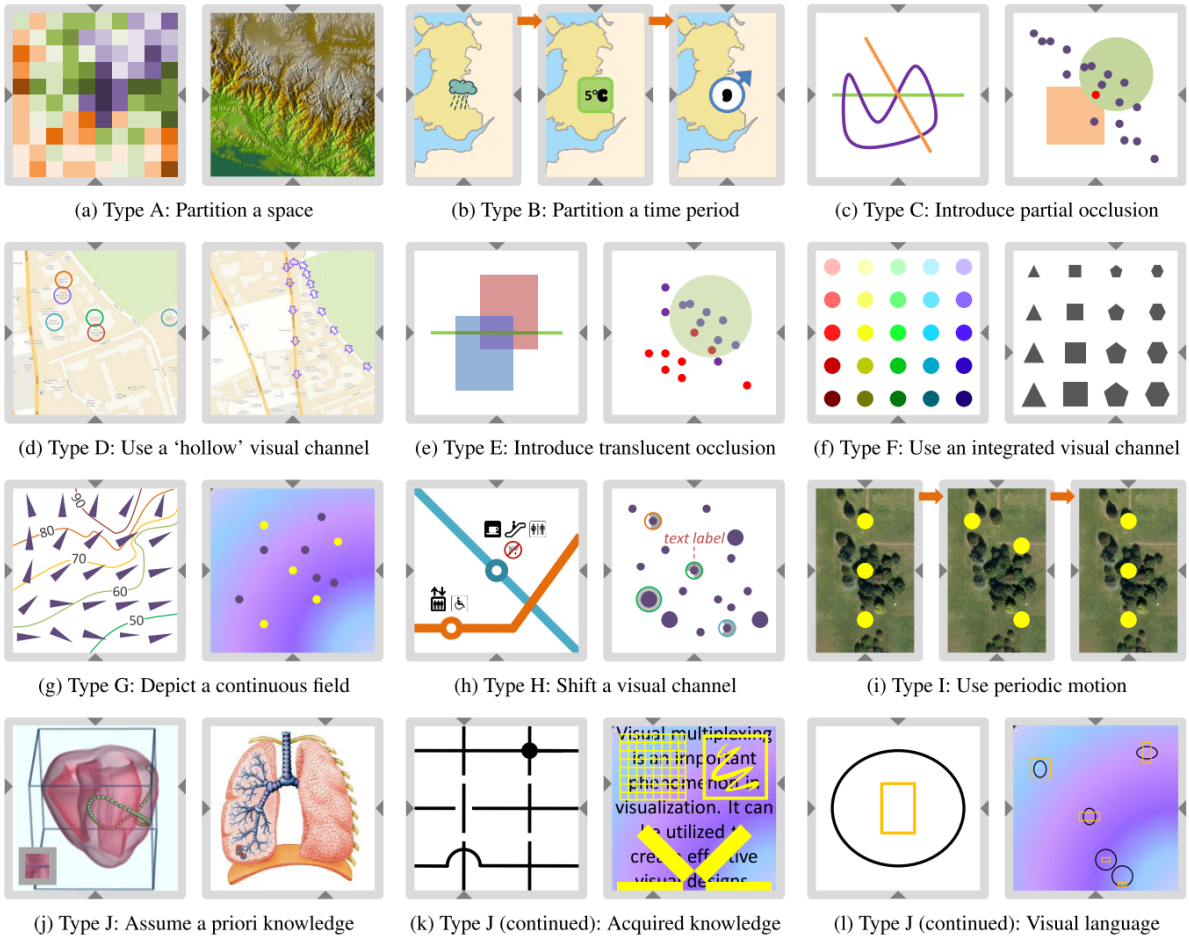

Figure 2.3: Overview of ten categories of visual multiplexing according to Chen et al. [34].

Another theoretical composition framework was introduced by Chen et al. [34]. Their design space is strongly influenced by signal-processing concepts. The authors refer to their contribution as *visual multiplexing* and highlight that the human visual system is capable of separating individual pieces of information from a combined visual signal. They therefore distinguish between four different types of multiplexing: frequency-division multiplexing (FDM), time-division multiplexing (TDM), space-division multiplexing (SDM), and code-division multiplexing (CDM), with SDM and TDM corresponding to the two main concepts of visual multiplexing. Using SDM, the available 2D visualization canvas is divided into areas, e.g., pixels or ink dots, which all simultaneously transmit their own visual signal. TDM, on the contrary, multiplexes visual signals over time, e.g., as animations. These two, i.e., SDM and TDM, and eight additional classification types are shown in Figure 2.3. Types A and B correspond to SDM and TDM, respectively. Type C represents a simultaneous superimposition of all visualizations, which typically leads to occlusions and uncertainty about underlying elements. Therefore, type D exchanges opaque shapes for their outlines and hollow interiors. Type E replaces opaque forms with transparent objects, but the interpretation of newly synthesized colors may require cognitive reasoning as to whether they correspond to blending results or separate information. Another form of multiplexing is through integrated visual channels, i.e., type F, such as using hue and luminance or shape and edge count of visual primitives. Type G exploits continuous scalar fields in which information is extracted through mental interpolation, e.g., bilinear interpo-

lation between evenly distributed glyphs. Type H uses nearest-neighbors information instead, and Type I relies on animations to encode periodic motion revealing otherwise occluded representations. Type J builds upon a viewer's prior knowledge such as, e.g., the rule of symmetry to mentally complete a visualization, or knowledge that can be learned in short term, such as novel specialized depictions or new forms of representations, e.g., glyphs. These representations can then in turn also exploit shape, size, color, location, etc.

Now that we have discussed theoretical composition concepts which highlight the wide variety of strategies, we focus on the actual hybridization in the next section, discussing how and where it can be carried out by a heuristic. We do this with exemplary publications that are also intended to illustrate the individual approaches in more detail.

2.2 Visualization Hybridization

The most straightforward approach to allow users to simultaneously explore different visual representations of the same data is through side-by-side visualizations, i.e., *juxtaposition*. In such representations the used heuristic is trivial, as visualizations are displayed next to each other without being changed. The basis for this is either the use of multiple program windows or adjacent monitors, and ideally a synchronisation of the user interaction. For example, Matkovic et al. [113] present ComVis, a research tool for rapid prototyping and testing of interactive visual analysis techniques. Their system is built upon juxtaposed views and supports different visual representations such as histograms, scatter plots, parallel coordinates, etc. A typical setup for the exploration of meteorological data consisting of eight different visualizations is shown in Figure 2.4(a). Another juxtaposition example is the visualization framework Improvise by Weaver [180], supporting chemical processes, networks, country maps, music analysis, etc. Figure 2.4(b) shows multiple visualizations of simulated trajectories of chemical elements. Both tools benefit from synchronized user interaction as key concept to link various different views. Our approach, however, simplifies finding an adequate representation as we instead derive the hybrid visualization based on information inherently embedded in the data, e.g., potential tasks and possible user interaction.

Integrated views serve as an extension of standard juxtaposition: they are augmentations of juxtaposed views with additional visual primitives, such as arrows or lines. In such representations, heuristics still display individual visualizations next to each

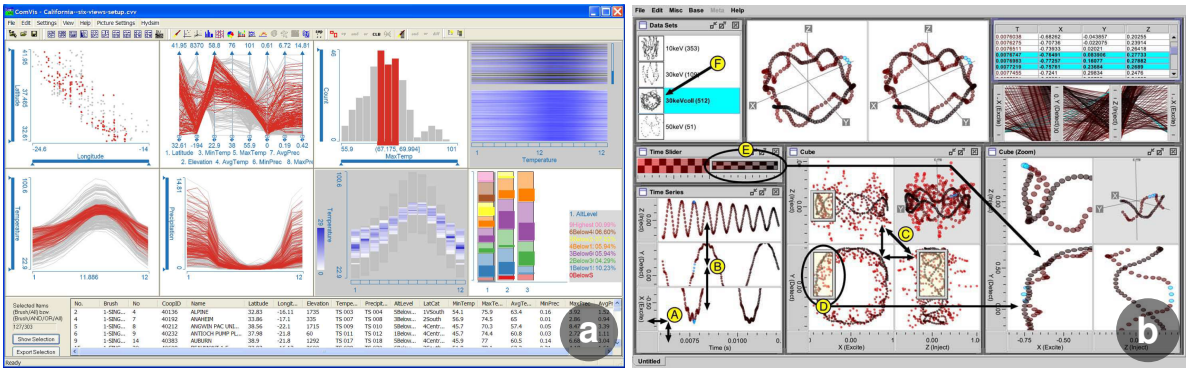

Figure 2.4: Interfaces of ComVis [113] and Improvise [180], both exploiting juxtaposition.

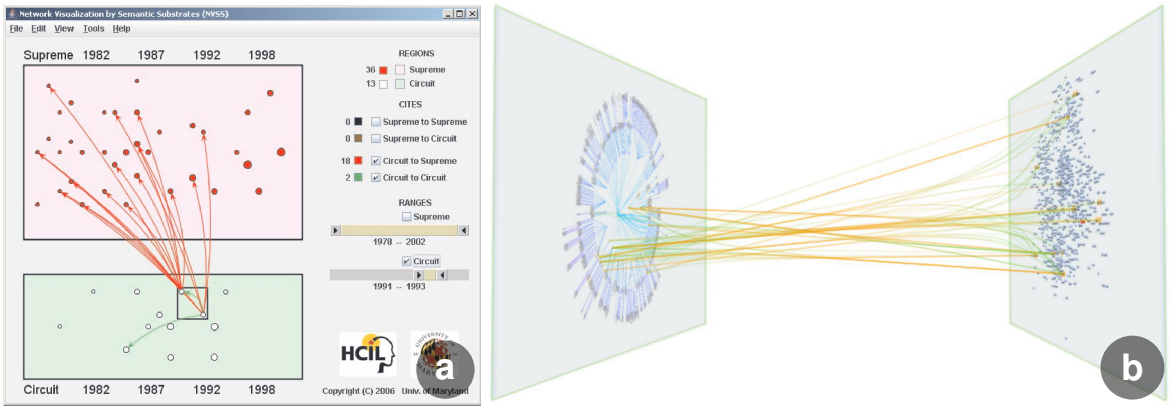

Figure 2.5: Two representative examples of integrated views, (a) semantic substrates [144] and (b) VisLink [40], both relying on additional visual encodings to highlight relational links.

other in a non-superimposed manner, but add explicit marks indicating how the visualizations (or parts of them) relate to each other. On the one hand, this reduces the cognitive workload as links between views are now explicitly encoded. On the other hand, however, this may simultaneously introduce additional clutter. Shneiderman and Aris [144] present semantic substrate, a network visualization tool based on node-link diagrams. The resulting hybrid consists of non-overlapping screen regions divided in categorical data attributes, whose size corresponds to the total number of data elements contained. In order to reduce clutter while increasing visibility, the links between elements are only displayed through user interaction. An example of an integrated view with overlaid arrows is shown in Figure 2.5(a). It corresponds to supreme court, circuit court, and district court cases between 1978 to 2002, and highlights which previous cases each judgment relates to. For example, green arrows indicate circuit court judgements and red arrows supreme court cases. We, however, show that a manual selection may not always be necessary to reduce clutter. Instead, we derive visual representations already defined by the tasks that are possible to perform with the data, and thus optimize the likelihood of a meaningful hybrid. VisLink by Collins and Carpendale [40] represents another example of integrated views. It correspond to individual visualizations mapped to planes, one per technique, connected by relational links. As shown in Figure 2.5(b), different planes can be positioned in 3D and their relations are visually emphasized. The presented data set contains lexical English words and the resulting visualization shows IS-A relationships, i.e., sets of synonyms. To reduce clutter, a distinction is made between two types of link encoding: straight lines which represent one-to-one edges, and line bundles which correspond to one-to-many edges. Nevertheless, the planes have to be arranged manually and connecting links are only displayed between two adjacent planes. In our approach, we explicitly avoided embedding data sets in an artificial 3D space and used expressive 2D spatialization cues instead.

To illustrate *nested views*, i.e., the approach of using a heuristic to embed a visualization in (graphical elements of) another, we compare naive juxtaposition to such an alternative. An example of conventional juxtaposition is the MatrixExplorer by Henry and Fekete [77], shown in Figure 2.6(a). It allows users to explore networks as a synchronised visualization consisting of a matrix representation and a node-link diagram. Unfortunately, using juxtaposition, the user is forced to continuously switch context, not only visually but also mentally. This may increase the cognitive load and could

2

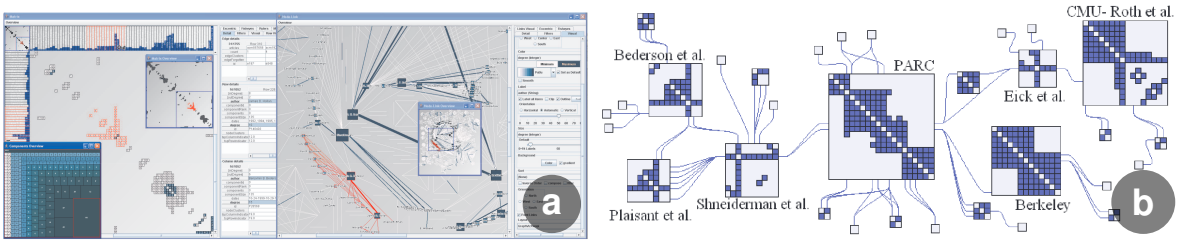

Figure 2.6: Two examples of graph or network hybridizations: (a) MatrixExplorer [77] using juxtaposition, i.e., a side-by-side visualization of a matrix representation and a node-link diagram, and (b) NodeTrix [78] where user interaction defines the visual representation.

further reduce attention. Therefore, Henry et al. [78] present NodeTrix, a representative example of a nested view shown in Figure 2.6(b). It is an interactive representation combining node-link diagrams with adjacency matrices. The node-link diagrams provide the user with an overview of the global structure of the graph, whereas the adjacency matrices nested inside the nodes show precise connectivity. The authors, however, highlight that their key contribution includes a set of interaction techniques which allow the user to toggle between the two visualization techniques. Using a smooth animation as visual transition between the two representations, novice users are supported to better understand their relationship. The heuristic in this example is, therefore, solely influenced by user interaction which controls where and how a node is rendered. Our heuristics, on the contrary, automatically derive the representation from the underlying data. Nested visualizations, furthermore, often require additional interaction, e.g., zooming, as a visualization may only be a small part of another. A similar approach called ChordLink [3] is also based on node-link diagrams but the interconnections are instead shown as chord diagrams. We, however, rely on hybrid visualizations adjusted to underlying data and tasks, allowing the user to immediately focus on the exploration rather than having to first find an adequate representation (manually).

Zhao et al. [189] combine hierarchical node-link diagrams with treemaps, shown in Figure 2.7(a). The authors chose these two techniques as they encode topological structures while being space efficient, compact, and scalable. They furthermore emphasize that hybrid visualizations require a particularly good visual design as they, on the one hand, build on established visualizations but, on the other hand, rely on novel visual encodings. Therefore, we also conducted multiple user studies analyzing the efficiency and understandability of our contributions compared to established techniques.

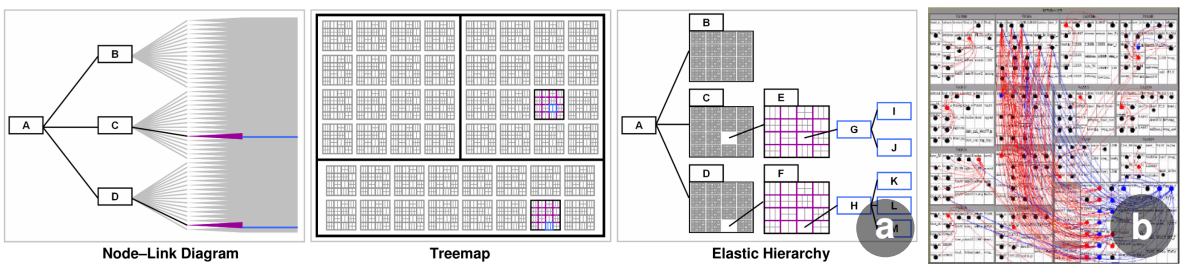

Figure 2.7: Graph and network examples using treemaps: (a) Elastic Hierarchy [189] combining node-link diagrams interactively with standard treemaps, and (b) Overlaid Graph Links exploiting Bézier curves [60] capable of mapping a graph to an overloaded tree structure.

Next, we explore an exemplary *overloaded view*, i.e., an at least partially overlapping visualization that exploits unused canvas areas. Fekete et al. [60] present treemaps overlaid with graph links, shown in Figure 2.7(b). The resulting visualization consists of several layers, but without a connecting spatial component in the data. Their technique initially decomposes a graph into a tree, again displayed as classic treemap, but encodes the excess edges as overloaded Bézier curves. To reduce overplotting, either only manually selected curves or curves starting/ending below the mouse cursor are displayed. The authors chose curves as an initial attempt because relying on straight line resulted in even more visual clutter. Due to different curvatures, a curve can additionally be used to distinguish between the start and end points. The importance function of our heuristic, however, can implicitly reduce overplotting and analytically maximize visibility by controlling when which visual encoding, e.g., layer, is topmost.

An established alternative to standard side-by-side or partially overlapping visualizations consists in *superimposed views*, i.e., fully blended visualization layers on top of each other, using transparency. They form the basis of our research and are often used especially when there is a spatial, e.g., geographical, connection between the individually layered visualizations. Such hybrids require less visual canvas space, compared to side-by-side representations that need twice as much, but may suffer more quickly from clutter or visual artifacts caused by blending. Voisard [175] presents Mapgets, a geographic visualization system from cartography. It is based on superimposed information layers, so-called presentation stacks, e.g., polygonal shapes of European countries, their names as labels, and line representations of rivers. A potential result of such a composition is shown in Figure 2.8(a). Here, however, it is the task of the user to arrange the layers in such a way that they are as expressive and meaningful as possible, while we transfer this task to a heuristic, i.e., an importance function, which does not only make global decisions about layer arrangement, but also decides locally and smoothly in image space. Lokuge and Ishizaki [106] present GeoSpace, a related approach also from cartography. Their visualizations are progressively influenced by text or speech queries. For example, if the user searches for Cambridge, the name of the city is enlarged and the opacity of its label is increased. The resulting geographic map of Cambridge, MA, including a subsequent second search query asking for crime data, highlighted as superimposed red points, is shown in Figure 2.8(b). Here, too, the order of the displayed layers depends on the order of the search queries making an optimal visualization dependent on the user.

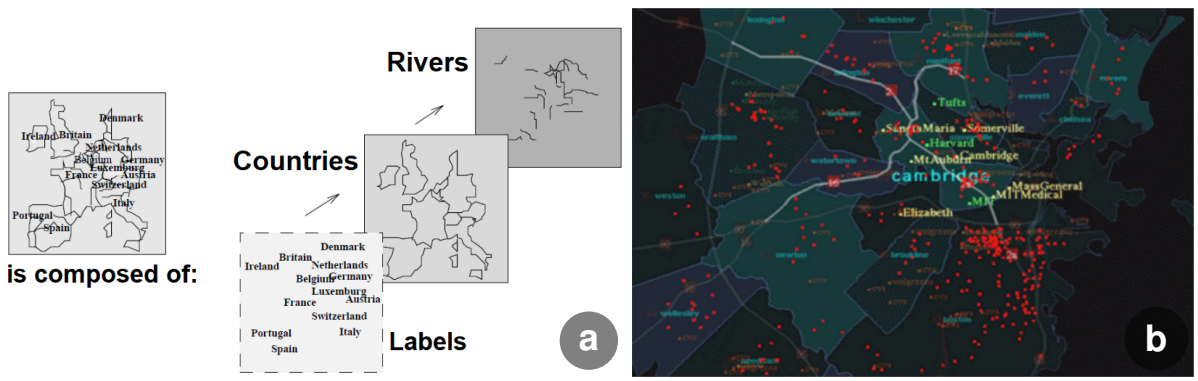

Figure 2.8: Two examples from cartography, (a) Mapgets [175] and (b) GeoSpace [106], both relying on superimposed views of separate information layers queried through user input.

2

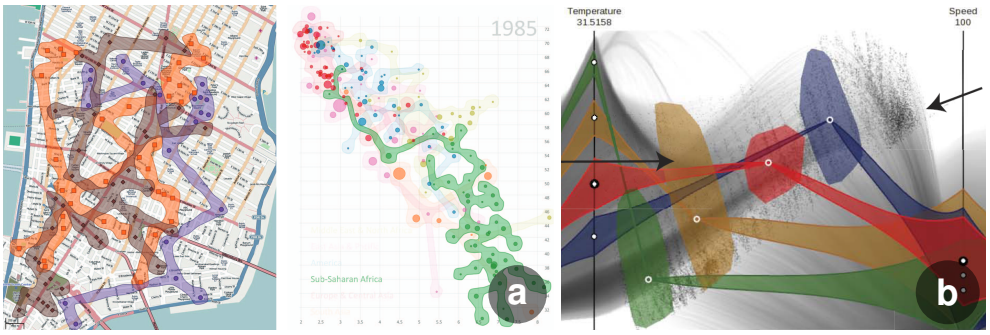

Figure 2.9: Two visualization hybridization examples that both exploit transparent overlays, (a) Bubble Sets [41] and (b) multivariate transfer functions by Guo et al. [68].

In the following, we present two more hybrids that exploit *transparent overlays* to create visualization hybridizations. Collins et al. [41] introduce Bubble Sets, a hybridization applicable to scatter plots, trees, graphs, text, maps, etc. Using their approach, both spatial organizations, i.e., connections, as well as set memberships relations without close spatial proximity can be encoded. As shown in Figure 2.9(a), the authors rely on additional continuous, transparent, and potentially concave iso-contours highlighting set memberships, blended over an already existing visualization. Guo et al. [68] present another approach for multivariate volume visualization which requires a novel hybrid transfer-function as combination of parallel coordinates plots (PCP) and multidimensional scaling (MDS) plots. During the interaction, this ensures that the user does not have to constantly switch context between different visual representations necessary for the high-dimensional feature space. An example of the overlaid multivariate transfer functions on top of the MDS plot is shown in Figure 2.9(b).

Other superimposed types of hybridizations, for example from the depiction of continuous fields, may require already *acquired knowledge* or a previously learned visual language. Ware [179] presents an approach for accurately readable bivariate scalar maps. Instead of two-dimensional color schemes, his approach relies on a combination of color for one variable and texture for the other. Such textures consist of quantitative glyph-like symbols, so-called textons, which encode different numerical values. These show, for example, small dots, lines, circles, squares, plus or minus symbols, stars, etc. An example of such a map is shown in Figure 2.10(a). An additional study confirmed that such a hybrid encoding is superior to classic 2D color maps, although textons cannot keep up with the spatial resolution of bivariate color maps.

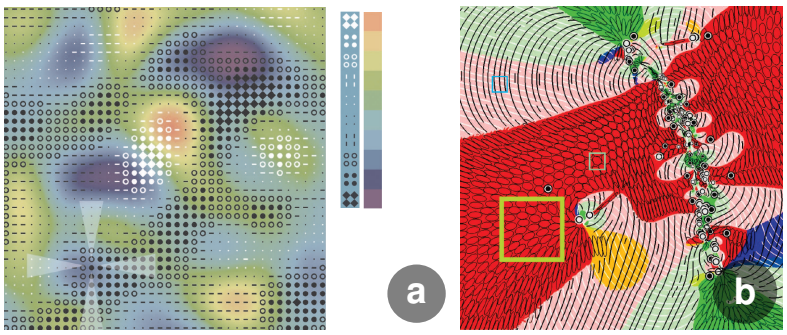

Figure 2.10: Examples of hybridizations based on learned visual encodings such as (a) bivariate maps [179] or (b) asymmetric tensor fields [127], both exploiting color and glyphs.

Palke et al. [127] present a related approach from asymmetric tensor fields, shown in Figure 2.10(b), that relies on hyperstreamlines and elliptic glyphs visualizing complex flow behaviour. The density of the streamlines and the size of the glyphs correspond to the magnitude of the respective tensor, freeing color for other data properties. In addition, such an encoding can be used to distinguish between real and complex domains as they are represented by streamlines or ellipses, respectively. In contrast, our additional encodings rely on spatialization cues avoiding the need of a complex legend.

We also drew inspiration for our contribution from volume visualization. Balabanian [11] presents techniques transforming linked views into integrated views and discusses difficulties such as occlusions and information overload. Our idea of using continuous importance functions that steer the hybridization heuristic was inspired by Viola et al. [174]. They introduce importance-driven focus and context rendering for volumetric data. In their approach, each object is assigned an importance value which prevents structures such as, e.g., organs or tumors, to be occluded by less relevant elements such as skin and bones. Three possible results that could not be generated by classic view-independent transfer functions are shown in Figure 2.11. Similarly, the importance of visual elements in our contribution can either be explicitly provided by the user, or implicitly derived and calculated from data or tasks.

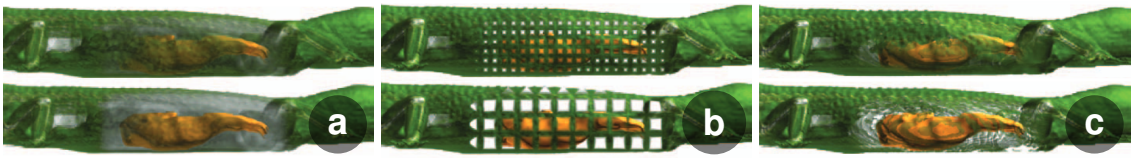

Figure 2.11: Possible results of importance-driven volume rendering using (a) opacity and color saturation modulation, (b) screen door transparency, and (c) volume thinning.

Finally, we present hybridization strategies that use a *hollow* visual channel such as, for example, the augmentation of volume data with pen-and-ink line drawings. Treavett and Chen [165] present non-photorealistic rendering wherein details such as texture and color of an iso-surface are replaced by simple line grids providing context information through sketch-style strokes. Figure 2.12(a) shows how properties such as the thickness of the tissues, e.g., skin, muscles, tendons, etc., around the skull can be captured through a hybrid visualization technique. A related approach for layered surface visualizations by Bair and House [10] extends single surface visualizations to multiple surfaces simultaneously. As shown in Figure 2.12(b), they exploit uniform grid textures with high luminance and varying levels of opacity, similar to our blending.

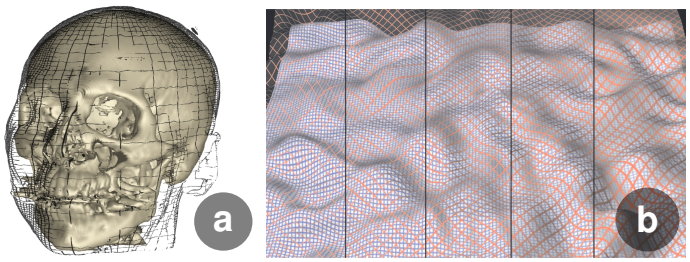

Figure 2.12: Hybridization examples through hollow visual channels such as (a) pen-and-ink lines on top of an iso-surface [165] or (b) grid textures for layered shaded surfaces [10].

2

2.3 Spatialization Cues

Our visualization hybridization contribution furthermore investigates potential benefits of spatialization cues to encode additional data-dependent properties. Similar to simple drop shadows in GUIs indicating, for example, whether buttons are convex or concave, i.e., pressed or not, we decided to investigate preattentive cues corresponding to an analogy of the real world. Shading is a simple yet powerful effect through which we intended to orthogonally expand the design space of our hybrids. Our aim was to investigate whether spatialization cues can be used to, e.g., enhance their legibility, emphasize local image variations, improve the users' accuracy with value estimation tasks, etc. Shading in visualization is not new, one of the most famous examples can be seen in Figure 2.13. It represents cushion treemaps [171], i.e., shaded treemaps. For this, trees are converted into shaded structural surface representations using directional lighting consisting of an ambient and diffuse illumination model.

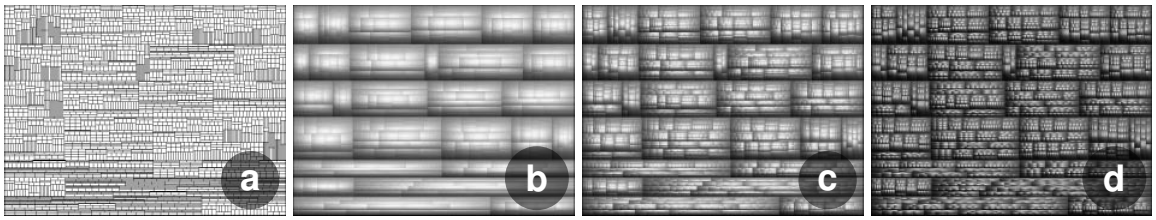

Figure 2.13: Comparison between (a) a treemap and (b-d) cushion treemaps [171] with different shading parameters, emphasizing the tree structure explicitly through spatialization cues.

Holten et al. [79] developed the idea of cushion treemaps further, by additionally applying bump mapping [16] to represent hierarchical software metrics. This should, on the one hand, simplify the perception by reducing the cognitive load and, on the other hand, reduce the time required to understand the visualized software system. The input for bump mapping, i.e., perturbing the surface normally used for shading, is a displacement array from a 2D noise texture synthesis, producing seamless texture tiles. These textures resemble wrinkled surfaces and are not only used as bump maps, but also as diffuse surface textures. Additionally, when visualizing software projects consisting of thousands of lines, highlighting packages containing multiple classes with black outlines and adding labels with varying font sizes appeared to be advantageous. Figure 2.14 shows two examples of such hierarchical software visualizations. The gray-scale color range of a cell encodes fan-in, i.e., the number of methods that call this

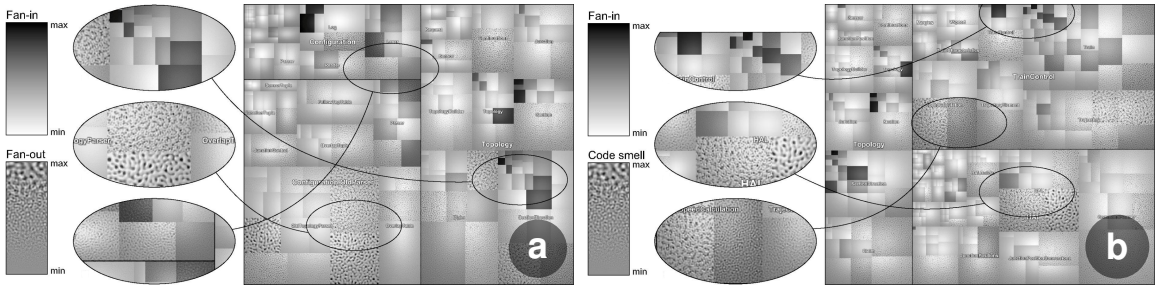

Figure 2.14: Juxtaposition of two cushion treemaps augmented with bump mapping [79], both encoding overall structures and source-code features of large software projects without color.

method, and the bump frequency encodes fan-out, i.e., the number of methods that are called by the function itself. Another way to exploit bump mapping could be to encode code smell, i.e., how easily a method can be refactored. Finally, the authors [79] point out that this type of double encoding avoids the use of bivariate chromatic maps, where the contained information is known to be more difficult to extract. Telea and van Wijk [157], furthermore, present an extension of cushion treemaps applied to Voronoi diagrams. Here, the shading is used to visually indicate the Voronoi seed that is closest. This approach can also be easily adapted to k-order Voronoi diagrams, where all pixels within the resulting polygons have the same k closest neighbors. An example of shaded (k-order) Voronoi diagrams is shown in Figure 2.15.

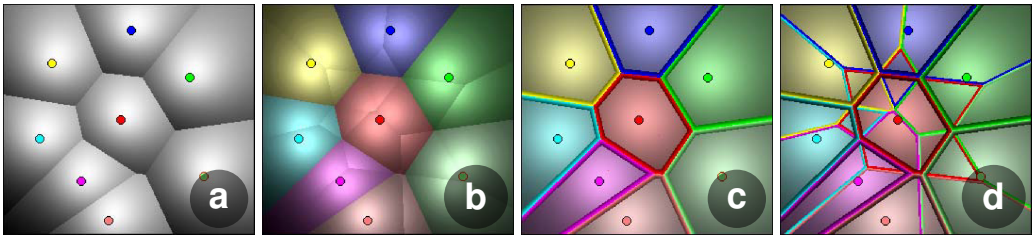

Figure 2.15: Overview of shaded (k-ordered) Voronoi diagrams [157] shown as (a,b) cushions and (c,d) their extension with bevels highlighting k-ordered, in this example 2, neighbors.

Bristle Maps by Kim et al. [95] present a technique for multivariate map encodings based on aggregation, abstraction, and stylization. The authors highlight that aggregation as commonly used to counter clutter, unfortunately, destroys the specificity of data point locations. Bristle Maps, instead, extend map elements such as roads, train lines, subway lines, etc. and exploit visual encodings such as length, density, color, orientation, or transparency of bristles for data properties. The resulting spatialized and, according to the authors, aesthetically pleasing visualization is shown in Figure 2.16(a).

An approach that visualizes hierarchical structures over time is presented by Bolte et al. [18]. By splitting nested streamgraphs and, additionally, exploiting halos and drop shadows, the authors generate expressive spatialized SplitStreams. An example of evolutionary changes between species over time is shown in Figure 2.16(b).

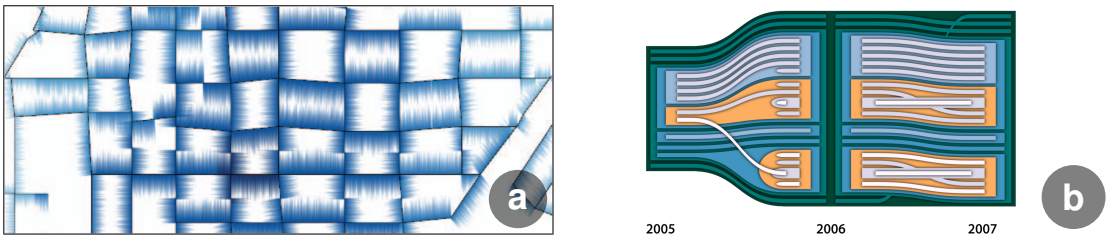

Figure 2.16: Examples of (a) a Bristle Map [95] of an area with high variance in graffiti vandalism and (b) spatialized SplitStreams [18] encode hierarchical changes over time.

Van Wijk and Telea [170] visualize scalar fields as height maps, i.e., reminiscent of a geographic topology containing shaded mountains and valleys. The authors highlight that, this way, structures corresponding to data properties can be perceived as 3D features such as ridges, peaks, and saddle points. Additionally, quantitative information can be extracted through color maps or iso-lines. Multiple examples and potential use cases of enridged contour maps, e.g., weather maps, are shown in Figure 2.17.

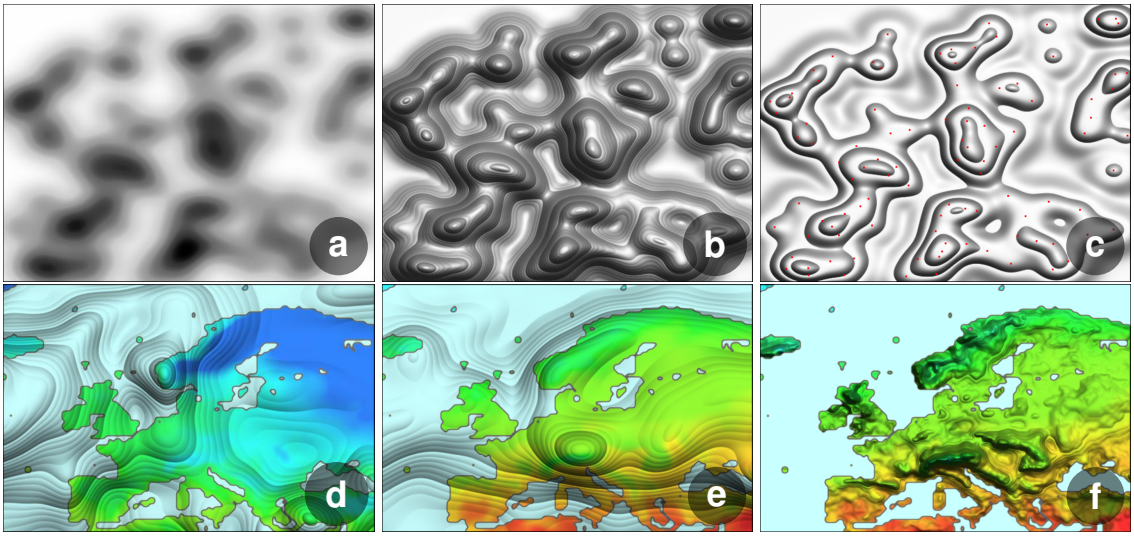

Figure 2.17: Enridged contour maps [170] rely on (a) non-linearly mapped height fields displayed as (b) hierarchical ridges, simultaneously encoding (c) underlying point cluster information. Typical applications are average temperatures (as color) and precipitation (as height) in Europe during (d) January and (e,f) July in low or high resolution, respectively.

Robertson et al. [136] introduced Cone Trees, i.e., 3D visualizations of hierarchical information structures. Since their trees are embedded into three-dimensional space, the authors rely on interactive animations to reduce the increased cognitive workload. The results consist of transparently shaded cones, avoiding occlusions of nodes behind them. When a node is selected, the tree rotates in an approximately one-second long animation, highlighting the selected node at the top. Depending on user preferences, the tree could otherwise also rotate continuously. According to the authors [136], this shifts the cognitive load to the human perceptual system. In order to further improve the 3D perception through depth enhancement, the resulting tree is additionally illuminated. As a result, individual graphical elements cast shadows, as illustrated in Figure 2.18. Such a 3D embedding, however, has to be done with caution. Although works like that of Brath [20] generally argue for 3D in information visualization, they also discuss possible issues such as 3D navigation, occlusions of 3D environments, selection and manipulation of 3D objects, perspective 3D projections, etc. We, therefore, rather strove for spatialization cues that only convey a pseudo three-dimensional impression on the actually underlying two-dimensional visualization avoiding these problems.

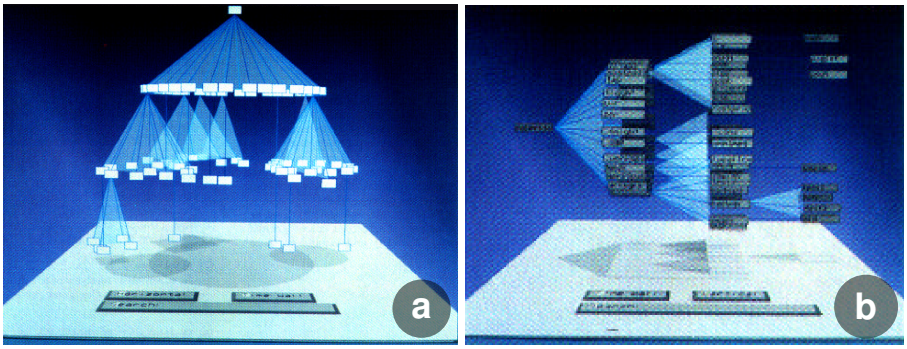

Figure 2.18: Example of different shading applications by Robertson et al. [136], (a) Cone Trees encoding clustering information and (b) Cam Trees emphasizing hierarchical structures.

Irani and Ware [85] investigated potential benefits of representing node-link diagrams as shaded 3D elements, so-called geon diagrams. Here, individual visual elements such as links between nodes as well as additional attachments correspond to actual simple three-dimensional geometry. The size of each element is determined by its importance and their appearance is shaded, emphasizing their three-dimensional structure. The authors further highlight that additional data attributes can be mapped to surface properties, e.g., color or textures. Examples of two geon diagrams including their corresponding hierarchical structures are shown in Figure 2.19. The authors, furthermore, conducted experiments and found that geon diagrams outperform UML diagrams in terms of error rate and required time. Additionally, they found that shaded geons were both more accurate and easier to remember than unshaded geons. The cognitive load may, however, have already been reduced by replacing text by color, being potentially easier to process and remember, independent of other additional cues.

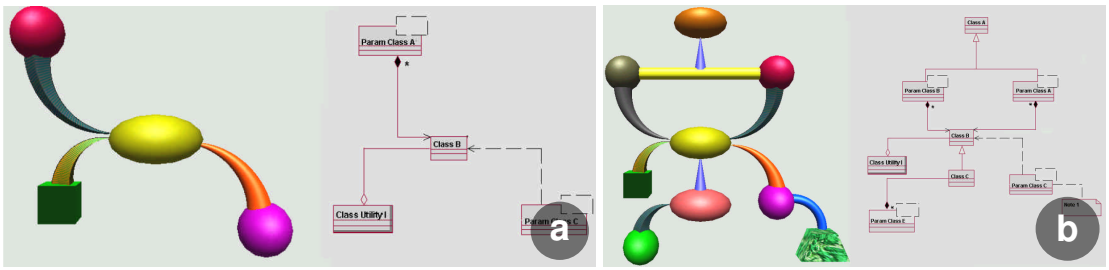

Figure 2.19: Examples of (a,b) geon diagrams [85] and their corresponding UML diagrams.

Irani et al. [84] furthermore conducted studies to analyze shading used in other, potentially more established, information visualizations. In these studies, the authors compared cushion treemaps to standard treemaps, as shown in Figure 2.20. Each tree visualization corresponded to a file system in which the color coding represented the included file types. The authors found that participants perform better on structure-based tasks as well as identification tasks when shading was enabled. The tasks' participants had to perform focused, among others, on the number, size, and content of files in (sub)folders. Further tasks were related to being able to mentally transform hierarchical node-link diagrams into space-filling visualizations by finding the right location of a given (sub)folder structure within a displayed treemap. Irani et al. [84] found that users were both faster and made fewer mistakes using shaded cushion treemaps. The authors also highlighted that participants preferred interacting with shaded over non-shaded

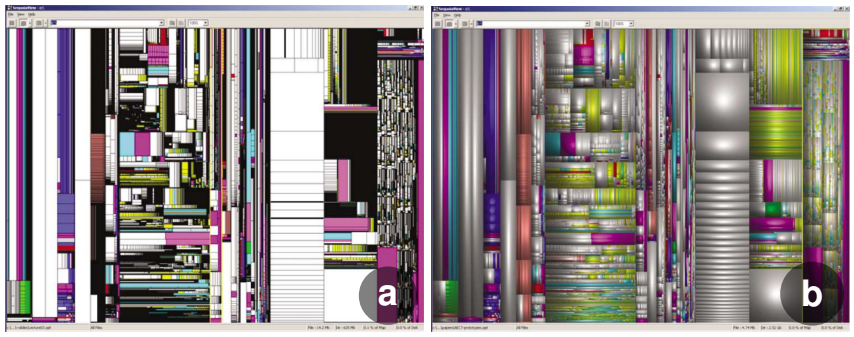

Figure 2.20: Examples of the visual stimuli used during the studies of Irani et al. [84], corresponding hierarchical structures encoded as (a) standard treemaps and (b) cushion treemaps.

visualizations. Finally, the authors [84] concluded with three findings, insights, and design suggestions that all served as inspiring guidelines and were decisive in developing our visualization hybridization techniques relying on spatialization cues:

1. *"Space-filling representations should use shading information to depict structural components."*
2. *"Visualizations in general that rely on common borders between elements and that require attention to specific elements could take advantage of shaded scenes for assisting the visual system in segmenting elements in the scene."*
3. *"In most interfaces, shading properties create visual effects that lead users to naturally click on items (as indicated by subjects in our experiments). [...]"*

Luft et al. [108] improved shape perception by exploiting depth information of images. In their approach, spatially important regions are identified by calculating the difference between the original depth information from a low-pass filtered version of it. The authors refer to this implicit high-pass filter as spatial importance function. Adding the high frequencies back to the original image provides indirect sharpening. Further depth cues can then locally enhance color, luminance, or contrast. This emphasizes features of complex scenes, for example, through slight darkening or additional bright halos. Figure 2.21 shows exemplary results of their technique.

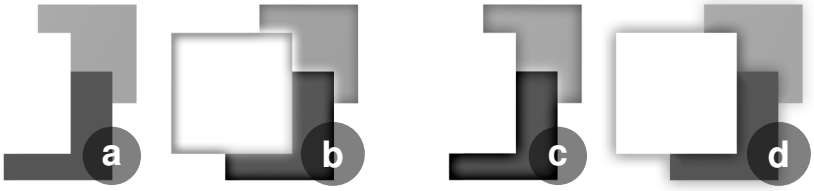

Figure 2.21: Enhancement of (a) an input image containing three squares by (b) adding the spatial importance function to the original image, (c) weighting this combination by absolute values of the importance function only, or (d) through depth-darkening [108].

A related approach by Everts et al. [59] exploits depth information for illustrative dense line visualizations. Their approach emphasizes tight line bundles through non-photorealistic rendering, i.e., using depth-dependent halos and depth cuing through line-width attenuation. The authors apply their technique to the most important line elements, e.g., the lines closest to the viewer and closest to each other, whereas line elements that are further away are filtered and de-emphasized. Figure 2.22 provides an overview of various currently used techniques, for example, (a) colored and shaded tubes, or (b) classic black lines rendered without and (c,d) with halos.

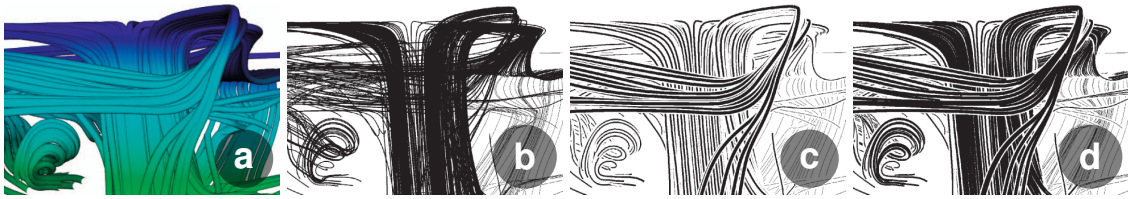

Figure 2.22: Comparison of line-rendering approaches such as (a) shaded tubes, (b) traditional lines, (c) lines with halos, and (d) illustrative lines with depth-dependent halos [59].

Rusinkiewicz et al. [139] developed a non-photorealistic shading technique conveying shape as well as surface details using exaggerated shading. As shown in Figure 2.23, their shaded reliefs are reminiscent of terrain representations used in cartography. For this purpose, the lighting is "softly" clamped as in toon shading, the used normals are smoothed at multiple scales, and the light direction is varied, i.e., projected onto the plane perpendicular to the smoothed normal, maximizing surface details.

Figure 2.23: Comparison of (a) a hand-drawn shaded terrain relief and (b,c) two results of exaggerated shading for depicting shape and detail by Rusinkiewicz et al. [139].

An alternative with similar goals of surface enhancement is presented by Vergne et al. [173]. Here, the authors improve the visual perception of shape through shading, independent of the used surface material and applied illumination model. They encode view-dependent shape features, e.g., the distance of an object to the observer or its orientation, through specific patterns of reflected light. For this purpose, surface properties such as the view-dependent surface curvature are initially extracted with a novel shape descriptor. Then the light is locally deformed accordingly through warping. This approach is based on the assumption that flat surfaces reflect small regions of the environment, whereas curved surfaces represent large and, therefore, more compressed regions. The resulting pipeline for enhanced surface depiction is shown in Figure 2.24. Similarly, we highlight relevant data features of our hybrids using spatialization cues.

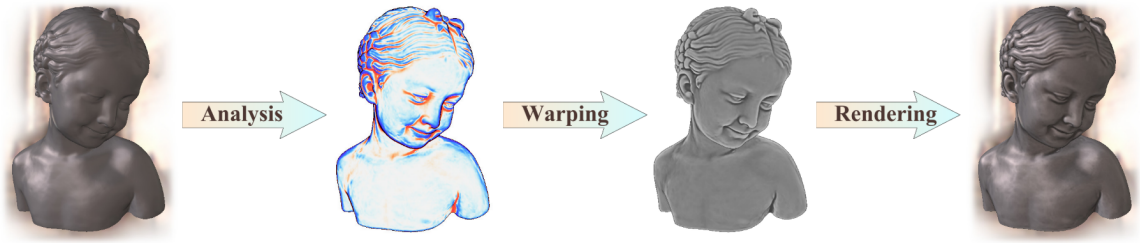

Figure 2.24: Light warping by Vergne et al. [173] relies on a shape descriptor that analyzes surface properties and warps the incoming light before rendering the model accordingly.

2.4 Shape Perception

In this section, we present findings from human perception studies focused on the effect of spatialization, e.g., structure from shading. On the one hand, they form the conceptual fundamentals of augmented hybridizations that exploit shading from computer graphics in visualization. On the other hand, they build the basis of the user studies we conducted in order to to understand the impact of spatialization cues on hybridizations.

2

According to Ramachandran [135], shape from shading corresponds to one of the strongest but also least understood aspects of human vision. It assumes that there is exactly one global light source, usually located at the top of an image, illuminating the entire visualization. If, however, this assumption is violated, the three-dimensional structure of different illuminated objects can only be perceived separately and sequentially. Furthermore, shading can also help to detect motion and resulting occlusions can be used to distinguish foreground from background objects. An example that illustrates the effectiveness of shading to provide strong depth impressions is shown in Figure 2.25(a). Additionally, shading simultaneously allows to distinguish between convex and concave structures, as shown in Figure 2.25(b). Ramachandran [135] further emphasizes that it is a misconception to assume that our brains always derive the simplest geometric interpretations. For example, in Figure 2.25(b) this simplicity would correspond to either only convex or concave objects illuminated by two light sources.

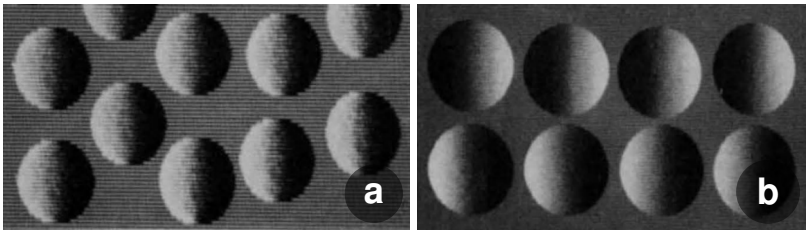

Figure 2.25: Illustrations from Ramachandran [135] showing that (a) depth, as well as (b) convex and concave objects, can be encoded through different shades of gray only.

Jochen Braun [21] conducted perception studies using diffusely illuminated spheres, randomly arranged in the form of a hexagonal array. Examples of the used stimuli are shown in Figure 2.26. His basic assumption is that shape from shading has properties similar to textural features such as luminance, orientation, or spatial frequency, which can all be processed in parallel. In the first experiment, participants had to (a) find a bottom lit sphere surrounded by top lit sphere distractors. Here, subjects performed significantly faster than when asked to find a top lit sphere within bottom lit spheres, i.e., the position of the light sources for target and distractors are swapped. The second experiment (b) differs from the first in that participants had to find two differently illuminated spheres instead of just one. In this case, time and correctness of the answers showed that shape from shading can be processed in parallel at different locations of the display. In the third experiment (c), participants had to detect whether a randomly rotated T or L letter target was contained beyond the differently illuminated spheres.

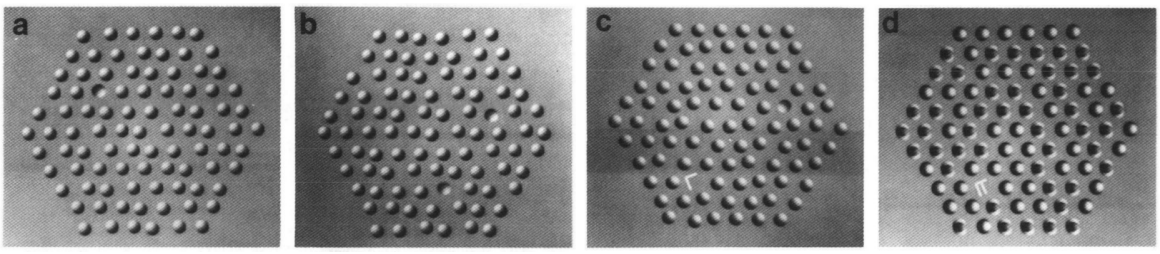

Figure 2.26: In the experiment setup of Braun [21], each trial began with a mark on the display with dimensions 512 × 512, 80 cm away from the viewers. Participants had to fixate this mark, followed by a blank interval, a visual stimulus (a-c), another blank, and then (d) a mask.

The results showed that shape from shading is independent of visual attention as both tasks can be fulfilled at the same time, although the discrimination between T and L already has a significant effect on the visual attention. In summary, his findings suggest that discontinuities in shading behave similarly to discontinuities in textural features.

Hanazawa and Komatsu [70] analyzed brain cell activity of macaque monkeys, namely their visual area V4 situated in the visual pathway for object recognition. They found that these cells extract 3D texture features of shaded granular surfaces. Such features encode surface characteristics including its composition, e.g., size, shape, orientation, and density of texture elements, which allow the monkeys to determine an object's friction. The visual stimuli for their study were generated by randomly scattered and partially overlapping Gaussian bell-shaped curves. The images used in their study with a resolution of 1024×768, 50 cm from the monkeys' eyes, are shown in Figure 2.27. Both the density and the size of the bell curves influence the brain signal intensity, with a maximum in stimulus 8 and a luminance gradient of 90°. This could be an indication of their adjustment to vertical light from above, but remains contradictory since the bell curves could be interpreted both as convex or concave.

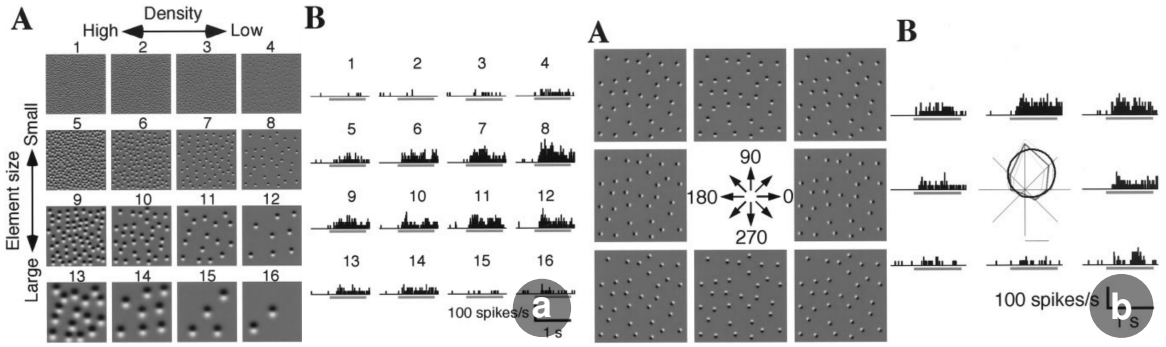

Figure 2.27: Findings from Hanazawa and Komatsu [70] showing V4 brain cell activity by means of texture features of (a) different sizes and densities, and (b) different illuminations.

In their work, Enns and Rensink [58] analyze sensitivity of preattentive vision in visual search. Their research is based on spatialization cues such as three-dimensionality or illumination direction. Their user studies focus on where exactly a target item surrounded by distractor symbols is visible on screen. The targets used in their studies correspond to simple polygonal objects colored in white, gray, and black. Some of them can be interpreted as 3D objects, for example differently oriented illuminated cubes. An overview of all used targets is shown in Figure 2.28, wherein each row of symbols corresponds to a single experiment. Participants had to find the target surrounded by one, six, or twelve distractors. The first experiment showed that recognition of 3D objects is faster than the of 2D targets (Figure 2.28, first row). In experiment two, they showed that different intensities, for example cube illumination alone, are sufficient and that 3D targets are again superior to 2D targets (Figure 2.28, second row). Cubes with a white top are considered to be lit from above, whereas cubes with a black top are lit from below. In the third experiment, Enns and Rensink [58] examined whether different types of lighting function with different degrees of efficiency (Figure 2.28, third row). The results showed that participants were quickest and made least errors when the search target deviated from default illumination direction, i.e., from above. Furthermore, the viewing perspective did not seem to have an influence on visual search.

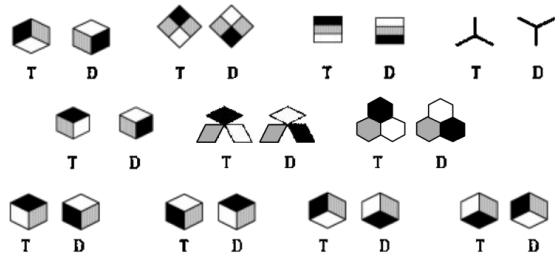

Figure 2.28: Overview of the visual search experiments conducted by Enns and Rensink [58]. Each experiment corresponds to a row of symbols, wherein (T) stands for the target that participants had to look for and (D) for the primitives used as distractors surrounding the targets.

In their work, Sun and Perona [153] investigate preattentive cues of 3D perception. These cues make 3D objects "pop out" compared to other non-3D targets. As part of their research, they analyzed whether pure 3D edge information, e.g., corner junctions with or without outlines, were sufficient or if additional shading applied to the shapes was required. In their first experiment, they compared shaded cubes, shown in Figure 2.29(a), to simple line representations of cubes (b) with and (c) without outlines. Each task was carried out by first showing participants a stimulus followed by a blank, and finally a mask. The mask, for example used in the shaded cube experiment, is shown in Figure 2.29(d). In addition to shaded cubes, similar to the work of Enns and Rensink [58], Sun and Perona [153] compared the acquired results to similarly colored, i.e., white, gray, and black, flat targets without an inherent 3D shape. They concluded that 2D shapes that resemble illuminated three-dimensional polyhedrons, e.g., cubes, can be processed in parallel and faster than serially processed line drawings of the same shape, i.e., a line representation of a cube. Furthermore, they found that only few participants recognized (e) the 3D shape of a dodecahedron, and although several participants recognized (f) truncated pyramids, they also indicated that the time during which the stimulus was visible was too short to be interpreted meaningfully. This might

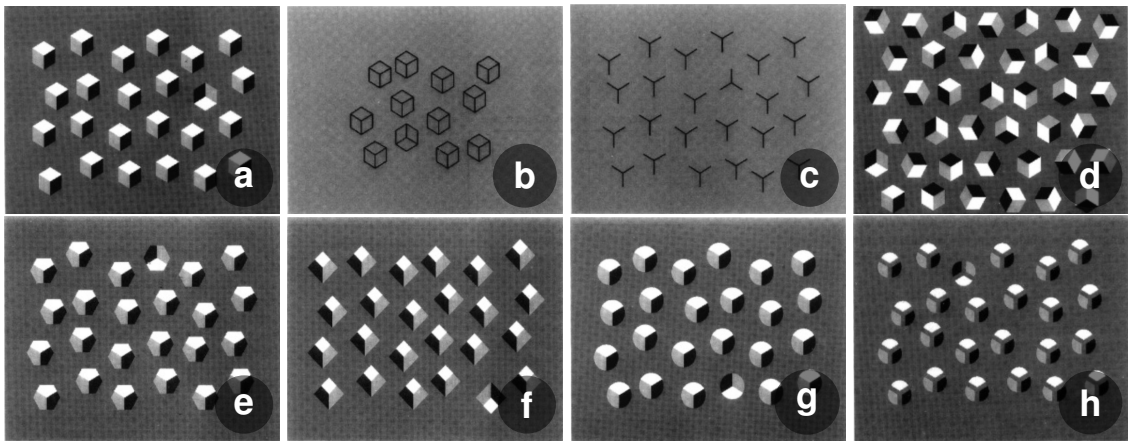

Figure 2.29: Sun and Perona [153] conducted perception studies comparing (a) shaded cubes, (b) line representation of cubes, and (c) Y shapes resembling the cube's edge structure without the hexagonal outline. After each stimulus, participants were shown a mask, for example, (d) of the cube experiments. The authors, furthermore, compared differently shaped polyhedrals such as (e) dodecahedrons, (f) truncated pyramids, (g) symbols that may resemble 2D pie charts, and (h) 2D pie charts with an additional gap separating the three regions.

stem from the fact that these shapes are less common and participants were, therefore, less familiar with them in comparison to generic cubes. Sun and Perona [153] also conducted experiments with shapes that (g) resemble 2D pie charts. They describe these as possible view of a 3D object with only a single corner, for example, shown through a circular aperture. The participants were able to solve this task in parallel and it proved even easier than the shaded cubes. However, if the individual parts were separated by (h) gaps, the task was significantly more difficult. Finally, the authors highlighted that the reverse task, i.e., in which all targets were rotated by 180°, was much more difficult for participants. This suggests that the 3D interpretation played an essential role in preattentive perception, more than just the symbol itself.

Nevertheless, caution is required when using depth cues, especially when emphasizing data that is not native in 3D. Zacks et al. [186] highlight that speed, accuracy, and the difficulty of extracting information from a graph depend on its visual properties. The authors conducted studies in which they analyzed how depth cues influence magnitude judgements. Participants had to estimate quantities of bar charts, as shown in Figure 2.30. They found that estimates based on additional 3D perspective depth cues were significantly less accurate, especially for higher bars. Interestingly, however, the error between volumetric bars and representation with additional visual "junk" near the elements of interest, i.e., bars, were similarly inefficient. This suggests that this is not due to depth cues but due to added visual complexity.

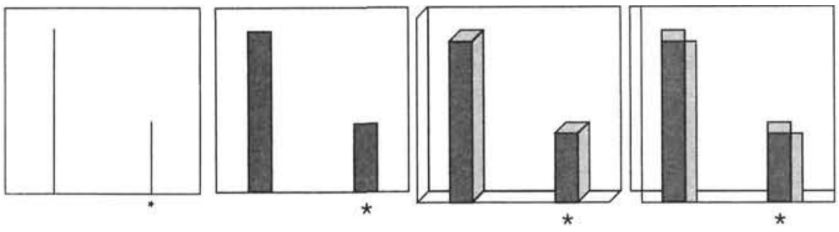

Figure 2.30: Visual stimuli used by Zacks et al. [186] evaluating the impact of 3D perspective depth cues on magnitude judgements. From left to right: a bar chart, an area chart, a volumetric representation, and an area chart with "junk" similar to volumetric representations.

A variety of graphical elements commonly used in visualization were examined by Spence [147]. These include horizontal and vertical lines, bars, pie and disk slices, cylinders, cubes, and numerical table entries, all shown in Figure 2.31. Their visual properties such as length, area, size, or volume often describe quantities such as aggregated frequencies, percentages, or ratios. Therefore, different visual encodings can be applied to the same underlying data. The length of a bar chart could, for example, encode the same information as the area, angle, or arc length of a pie chart. For the study, each graph was scaled to 22×17 cm and placed about 1m away from the participants. Spence [147] found that tables, followed by pie and bar graphs, provide the most accurate estimations. Cubes, cylinders, and lines performed worse, but disks achieved the worst results, likely because ellipses are harder to interpret than circles. Furthermore, Spence [147] emphasizes that there appears to be no direct link between accuracy and apparent dimensionality of the visual coding. Interestingly, lower dimensionality has led to longer response times, approximately 50% slower than those for 2D and 3D elements. The author, furthermore, concluded that visually appealing graphs resembling real-world objects are perhaps better remembered than their abstract counterparts.

2

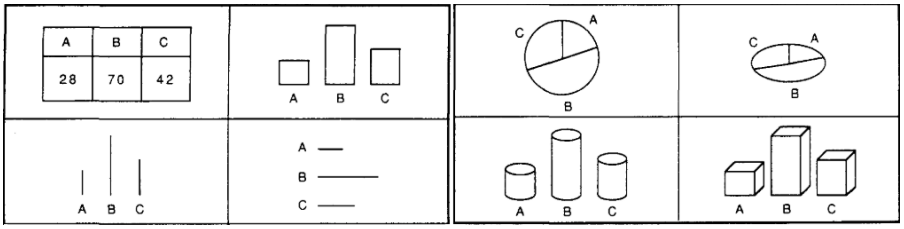

Figure 2.31: Overview of the commonly used graphical elements analyzed by Spence [147]. These can be divided into three groups, defined by their dimensionality: 1D as horizontal or vertical lines, 2D like bar and pie charts, and 3D such as cylinders or cubes.

2.5 Summary

In this chapter, we outlined the state of the art relating to our contribution. The theoretical basis of our work is formed by general composite views whose purpose it is to combine multiple visualizations. We have focused particularly on overloaded views, i.e., views in which the individual underlying visualizations are combined by superimposition. In particular, we have analyzed how heuristics can affect where and how visualizations are combined. These heuristics can be controlled by the user but also by the underlying data. The combination of individual underlying visualizations is done locally and can change smoothly in image space. Additionally, heuristics are able to implicitly reduce occlusions and visual clutter as they consider the tasks likely to be performed by the user. Such a result corresponds to what we refer to as visualization hybridization. We then explored how generated hybrids can be extended so that additional information or properties derived from the data can be encoded. Our focus was on augmentations that are orthogonal to the main visual encoding. Therefore, we have investigated spatialization cues as a simple yet powerful and well-established tool from computer graphics and visualization. We explored shading rather than an actual embedding in 3D as this could lead to common disadvantages such as 3D navigation, occlusions, perspective projections, selection and manipulation in 3D, etc. Subsequently, we explored already established visualizations that exploit spatialization cues. Finally, we have studied perception literature researching the human visual system's perception of shape cues. Here, we have presented already studied phenomena and explored to what extend related approaches could build on these.

Chapter 3

Contributions

In this chapter, we instantiate the concept of visualization hybridization with spatialization cues. For the presentation of our contribution, we focus on two fundamental and widely used data types. First, Section 3.1 highlights hybridization concepts for *point-based* visualizations and second, Section 3.2 presents augmentations through spatialization of *line-based* representations. We chose these as they correspond to some of the most common data types of information visualizations across various research areas, which are equally used for education at schools or universities, as newspaper graphics or on TV news, and so forth. Additionally, their broadness enables us to present general visualization hybridization concepts applicable for diverse usage scenarios, different data densities, diverse visual encodings, and different dimensionalities.

The foundation of our hybridization approach is the data domain including task-specific representation derived from it. By its classification, we find applicable tasks based on heuristics, e.g., sets of tasks that are likely to be performed on the underlying data. The resulting hybridization is then controlled through importance functions, i.e., they steer how different visualizations required for multiple tasks are simultaneously combined. Our approach implicitly reduces the likelihood of insufficient or inappropriate visual encodings, with which relevant tasks could not be reasonably performed. For example, a density encoding for outlier detection, or the exact opposite, i.e., estimating density based on few discrete data samples. Our heuristic provides a single, smoothly blended, hybrid visualization, relying on a continuous quantitative importance function, instead of multiple subsets of ideally adequate visual representations.

If hybridization is based on a heuristic that exploits information already contained in the data, two possibilities arise. First, data sets may contains lots of implicit information, i.e., already stored information content. Typical examples are cluster memberships, relevance of individual data items, order such as sorting or rankings, arrangements and connectivity of elements, their layout or regional context such as geographic area or mathematical number range, etc. Second, potentially little additional information is included, which may require assumptions or derived knowledge. This, however, does not exclude the possibility of being able to estimate or calculate parameters not explicitly included in the data set. For example, among others, geometric properties, density of data items, structure and spatial distribution, overplotting or visual clutter, etc. As part of our research, we addressed both, i.e., implicit and derived, data set information. The heuristics of our contributions are significantly controlled by data density, potentially increasing information content while reducing visual clutter and occlusions.

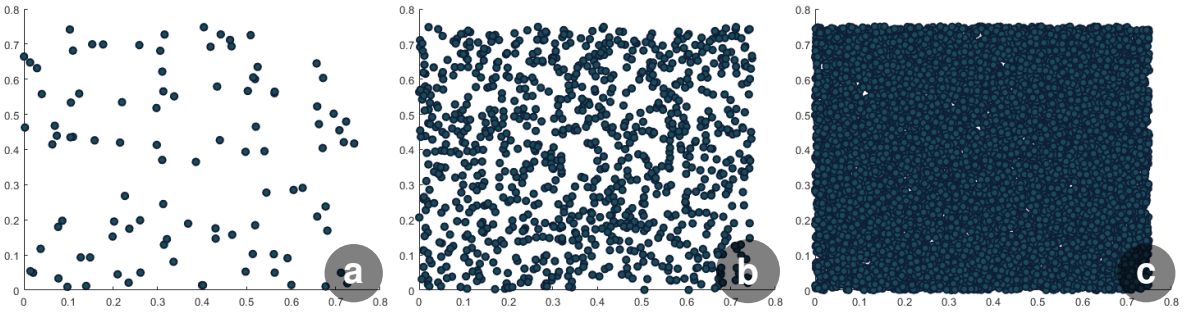

Figure 3.1: Overview of scatter plots with increasing numbers of points, (a) 100, (b) 1,000, and (c) 10,000, highlighting that interpretability is made more difficult by increasing overplotting.

3.1 Point-based Techniques

Bivariate data represent relationships between two variables, for example, encoded by the X and Y axis. This corresponds to a collection of points on the plane in the two-dimensional Euclidean space. Scatter plots present a common and well-established technique for visualizing such data sets of varying density and complexity. They support reading data values from individual points, but may also be used to explore diverse phenomena such as, i.a., identifying trends and clusters, locating outliers, etc. Unfortunately, especially when displaying a multitude of points, these visualizations can suffer from overplotting, i.e., data points with similar values are drawn on top of each other, which makes their interpretation difficult. It is then, for example, not easily possible to distinguish between areas of different density within the plot. Figure 3.1 shows three examples of scatter plots with an increasing number of points, (a) 100, (b) 1,000, and (c) 10,000 samples. Therefore, alternative or additional representations are required if scatter plots reach their limits as representation type for sparse data sets.

Figure 3.2(a) shows another data set containing 10,000 data samples, with a round central cluster. In contrast to before, however, further visualization techniques are now used to facilitate the interpretation of the density of such an overplotted data set. A possible approach is to color-code density separately using a heat map, e.g., a color gradient, from yellow corresponding to densest region, to purple representing sparser areas within the visualization. Assuming a scatter plot represents a continuous phenomenon, (b) contour lines can then be displayed on top of the scatter plot, similar to the topology of a hiking maps. Alternatively, this color coding can also be applied to (c) individual points corresponding to the number of overplots. For example, the more

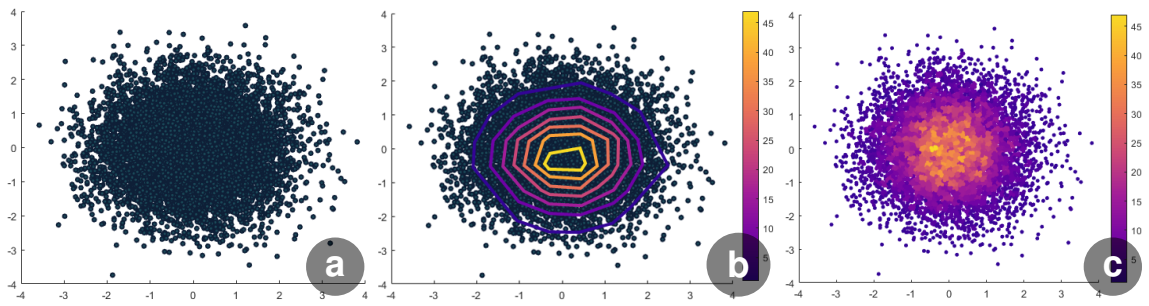

Figure 3.2: Examples of (a) a standard scatter plot with (b) additional color-coded density of the underlying points using contour lines and (c) points colored according to the overplots.

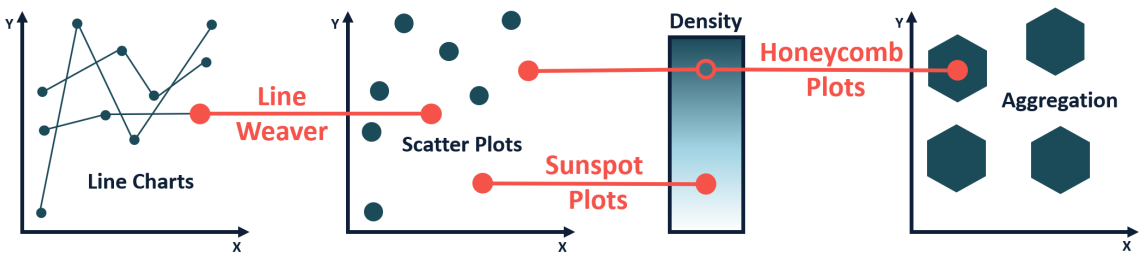

Figure 3.3: Classification of our contributions related to data types and applied techniques.

often points are drawn on top of each other, the brighter their colors get. These additional encodings reveal that the central points of the plot tend to be more overplotted than their surroundings. The form of visual representation for bivariate plots, therefore, highly depends on the underlying tasks, for example identifying off-center outliers or estimating the density distribution throughout the plot.

Besides task definitions by Brehmer and Munzner [22] which relate to whether a search object, i.e., target, or its location is known or unknown, there are other task-specific definitions especially for bivariate data. For example, Sarikaya and Gleicher [141] define a list of abstract analysis tasks for scatter plots divided into three groups. Group one covers object-centric tasks, for example, focusing on identifying, locating, and verifying an object including comparisons of multiple objects. Group two tasks all relate to browsing, for example, exploring properties of elements in an object's neighborhood, searching for known patterns such as clusters or correlations, and exploring potentially unusual phenomena. Tasks of group three rely on an aggregate level and, for example, characterize distributions, identify abnormalities as well as correlations, or focus on numerosity comparisons such as densities of different regions. Our goal was to support object as well as aggregate-centric tasks within a single visualization, which has currently often required different visualizations or a sequential analysis. Figure 3.3 shows an overview of how our three publications can be classified in this regard, e.g., *sunspot plots* as hybrid for discrete and continuous visual representations (Section 3.1.1) and *honeycomb plots* as hybrid for discrete and aggregated techniques (Section 3.1.2). Our third publication *line weaver* instead focuses on relationships between bivariate points, i.e., it is rather applicable to various types of line charts, such as time series or parallel coordinate plots (Section 3.2).

3.1.1 Hybridization of Continuous Representations

The efficiency of scatter plots depends on the density of the data set. If it becomes difficult to perceive individual data points, a point-based visualization reaches its limits. Well-established techniques such as opacity regulation like alpha blending, a reduction of the point marker size, and subsampling can increase the readability but, unfortunately, only to some degree. Even in an opacity-regulated scatter plot, the user can only distinguish between different densities until a certain threshold, e.g., full opacity, is reached. A reduction in the point size is also limited, for example, due to the used monitor resolution, which is why data points cannot become smaller than pixels without requiring additional user interaction. Subsampling also highly depends on the underlying statistical concepts, which have to ensure that no relevant data features are

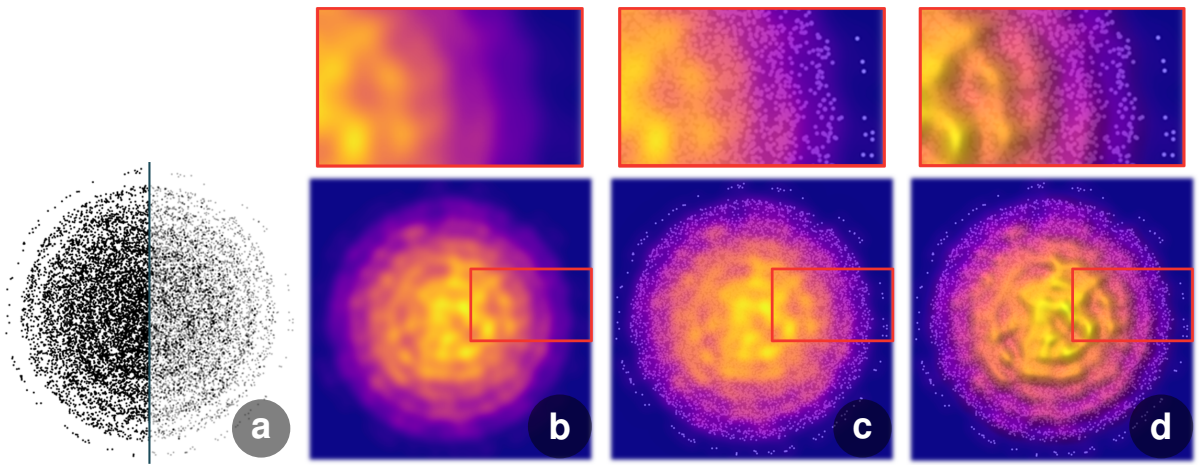

Figure 3.4: Overview of (a) a classic and an opacity-modulated scatter plots, and (b) a heat map of its density. Both build the basis of (c) sunspot plots and (d) their illuminated variant.

removed or that no artificial features, not being present in the underlying data, are generated. Alternatively, if bivariate data is so dense that its visual representation results in clutter and overplotting, visual encodings that correspond to the underlying density, e.g., through kernel density estimations (KDE), are common alternatives. The resulting continuous scalar field can then be directly highlighted using a heat map. Such color coding allows the observer to perceive densities of specific regions within the scatter plot, although there may not even be points. Nevertheless, if the density of the underlying data set is highly inhomogeneous, i.e., if it contains relevant sparse as well as dense features, a heuristic is needed to determine which technique should be applied to what part of the visualization. Particularly dense areas could be displayed using a density estimation, whereas sparser regions would benefit from a scatter plot representation.

We, therefore, combined two common techniques, i.e., hybridized scatter plots and density estimations, and consider individual data points as samples of a probability density function (PDF). By using an on-the-fly kernel density estimation, we achieve interactive performance allowing the user to adapt parameters blending the continuous surface model over the discrete data points in real time. The resulting hybrid visualization augments standard scatter plots by encoding the absolute values of the PDF using colors, i.e., the distance to the surface. Additionally, we wanted to encode important visual attributes like orientation of slopes and the accessibility of a surface area. Therefore, we added the support for local illumination models in combination with ambient occlusion. Figure 3.4 shows an overview of (a) a standard scatter plot including an extension which encodes density through opacity modulation. Additionally, it shows (b) a heat map encoding of the calculated density estimation, (c) a sunspot plot, i.e., a hybrid of scatter plots and density estimations, and (d) an augmented version of it using spatialization cues, i.e., illumination. As result, in case of overplotting the illuminated surface is visualized, e.g., in the yellow center region, whereas in sparse regions the individual sample points are maintained, e.g., in the circular purple border region.

Figure 3.5, furthermore, demonstrates how shading from illumination can highlight surface properties of the resulting scalar field. It contains (a) an imaginary viewpoint from "the side" showing a surface representation corresponding to the underlying point-density, illuminated by a global light source positioned top left. The resulting shading

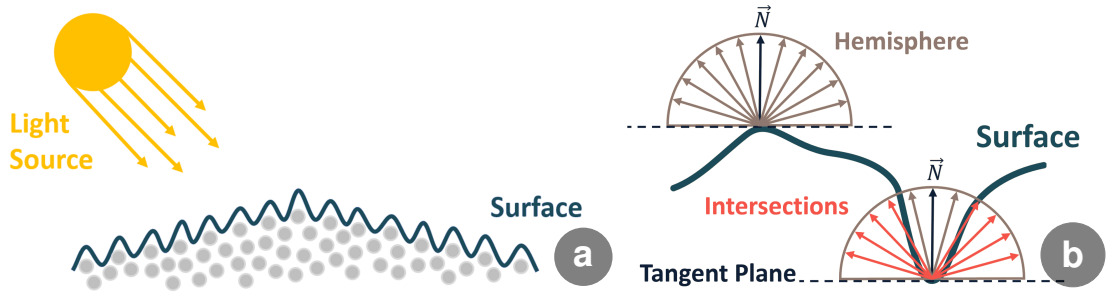

Figure 3.5: Juxtaposition of (a) an imaginary viewpoint from "the side" where a surface based on the density of underlying points is illuminated by a global light, and (b) an illustration showing how ambient occlusion can be determined using intersections of the local hemisphere.

emphasizes surface properties and can be interpreted similarly to topological hiking maps. In addition, we encode (b) the accessibility of the surface using ambient occlusion (AO), by examining to which degree its hemisphere, i.e., half-sphere, is obscured or intersects with neighboring topological structures.

A comparison of (a) a standard scatter plot with a polynomial trend line, and (b) a sunspot plot is shown in Figure 3.6. The visualized data set corresponds to the 2014 Boston Marathon [119] containing results of approximately 32,000 participants. The plots compare race times with bib numbers, which are assigned based on the qualification times. Similar to scatter plots but in contrast to classic density representations, sample points in sparse areas such as participants of the wheelchair race with the lowest bib numbers and a finishing time between one and two hours remain visible in sunspot plots. It becomes visible that sunspots can, in addition to density encodings, also indicate how well trends can be approximated, for example, precisely between bib numbers 0 and 27,000 and only fuzzily between 27,000 and 36,000.

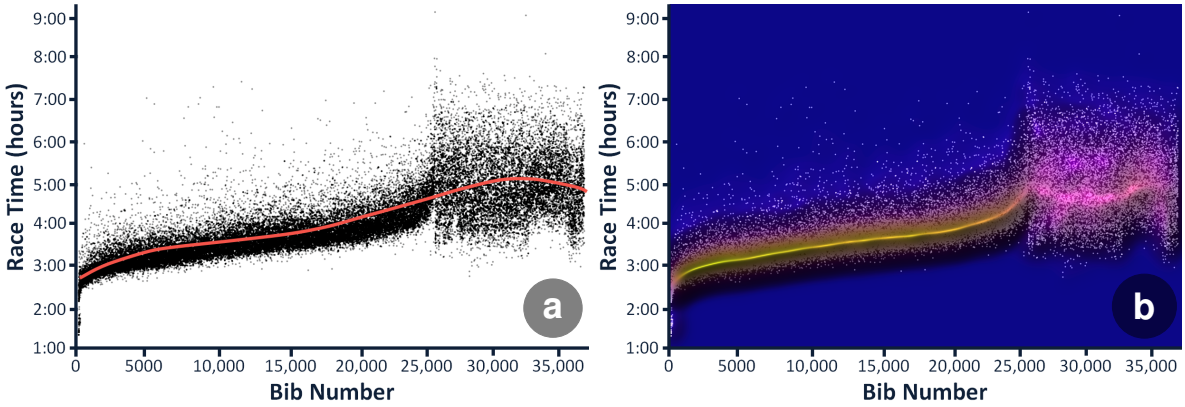

Figure 3.6: Comparing (a) a scatter plot to (b) a sunspot plot using marathon results [119].

Furthermore, we investigated how these light and shadow effects from spatialization cues affect the human visual perception when performing typical tasks on bivariate data. As part of our contribution, we evaluated the resulting hybridizations using qualitative use cases from synthetic and real-world data sets, as well as through a quantitative user study. We analyzed whether the desired advantages apply and disadvantages of individual methods have been reduced. Additionally, we estimated whether the tasks previously requiring multiple techniques can now be performed simultaneously using

hybridization. During the evaluation, we paid particular attention to verifying whether our techniques can be learned within short time, approximately 30 minutes. Our main concern was that the blending of individual points and the density representation might negatively influence user perception. In total, we compared five visualization stimuli: scatter plots, colored scatter plots, continuous heat maps, sunspot plots, and shaded sunspot plots. The study consisted of two analysis tasks, an assign task where participants had to estimate the average density within a highlighted region, and a compare task where participants had to find the highest or lowest density region, or could indicate that all regions are equally dense. We randomized the order of the tasks and assigned one visualization type to each participant, following a between-subject design and avoiding learning bias. In total, we had 350 participants corresponding to 70 per visualization type. Our findings suggest that color coding of scatter plots improves the users' accuracy in density estimation. Furthermore, the results demonstrate that there is no significant difference between continuous heat maps and sunspot plots, indicating that our technique utilizes the benefits of scatter plot without losing the advantages of density estimation provided by heat maps. We could not confirm that the introduced illumination model and AO improve the users' accuracy using sunspot plots and shaded sunspot plots; however, we observed significant differences between heat maps and shaded sunspot plots. Paper **A**—*sunspot plots* contains our hypotheses and their evaluation, including further use cases that illustrate the advantages of our hybridization.

Last but not least, we would like to highlight that the sunspot plots paper was featured in music videos of an ABBA cover band during EuroVis 2020 in Norrköping, Sweden. Two exemplary screenshots of their performance are shown in Figure 3.7.

Figure 3.7: Two examples of an ABBA cover band featuring sunspot plots (rightmost paper).

3.1.2 Hybridization of Spatial Aggregations

An alternative to continuous density estimation such as KDE, that helps interpreting dense and overplotted data sets, is spatial aggregation. Instead of visualizing individual points, hexplots accumulate points within grids of regular hexagons. The density of points within tiles is then encoded by heat map colors. An example of (a) a scatter plot and (b) its hexplot representation is shown in Figure 3.8. In this example, yellow corresponds to the densest tiles and purple to the sparsest. The visualized data shows an increasing trend from bottom left to top right. Here, however, it is interesting to highlight that the amount of white space or the apparently darker mass of points cannot be used to extract density information. For example, points in the upper right are plotted

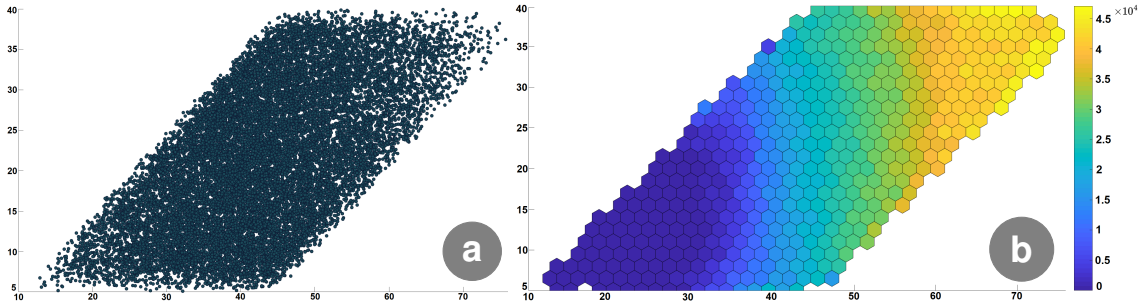

Figure 3.8: Comparison between (a) a scatter plot and (b) a hexplot of the same data set.

directly on top of each other and, therefore, visually appear sparser although these tiles actually contain most of the points. Unfortunately, counting "visible" points or relying on apparently cluttered masses of points does not provide the complete information.

Although hexplots are ideal countermeasures for visual clutter and overplotting, they have similar disadvantages as continuous density estimations. Users can no longer perceive how data points are distributed within tiles, for example, whether they are uniformly distributed, clustered, or form trends. This may lead to the assumption that their distribution is either uniform or irrelevant to the task. Furthermore, the differentiation of densities within hexplots is only possible via color, which makes the comparison of similar tiles dependent on the color fidelity of a monitor or the quality of a color print. We, therefore, hybridized scatter plots and aggregating hexplots applicable to data sets, for example, originating from scientific domains such as cartography. Again, our hybridization relies on a heuristic in the form of an importance function which smoothly blends between both visualization techniques, enabling density-based tasks requiring binning, as well as sparse-area tasks such as analyzing outliers. Figure 3.9 shows a comparison between (a) a standard scatter plot, (b) a hexplot encoding densities as colors, and (c) a honeycomb plot hybridizing both techniques into one.

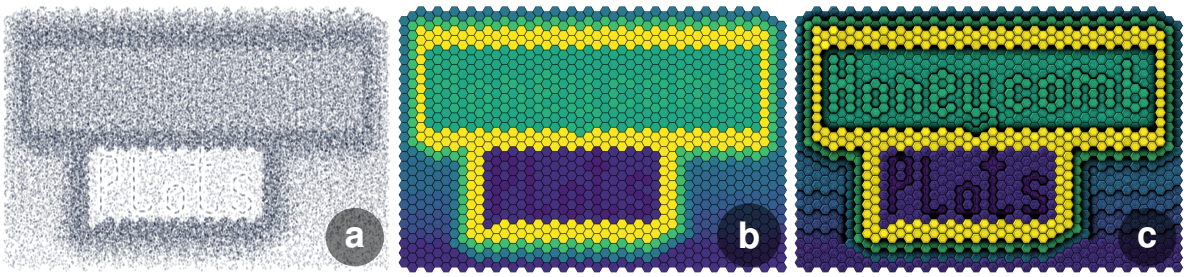

Figure 3.9: Honeycomb plots (c) combine (a) scatter plots and (b) heat maps using density. Through shading, ambient occlusion, and an implicit point-distribution encoding, the observer can explore data features that cannot be captured by either of the techniques alone.

In addition, honeycomb plots exploit three techniques that further augment the information content of common hexplots. As illustrated in Figure 3.10, we (a, b) propose a *relief mosaic* that exploits ambient occlusion highlighting differences between similarly colored tiles, (c) emphasize trends or clusters in sparse regions using an *amber inclusions* metaphor and finally, as shown in Figure 3.11, introduce a regression plane as *diamond cut* that reveals underlying point distributions. In the following, we present expressive usage examples exploiting all three techniques.

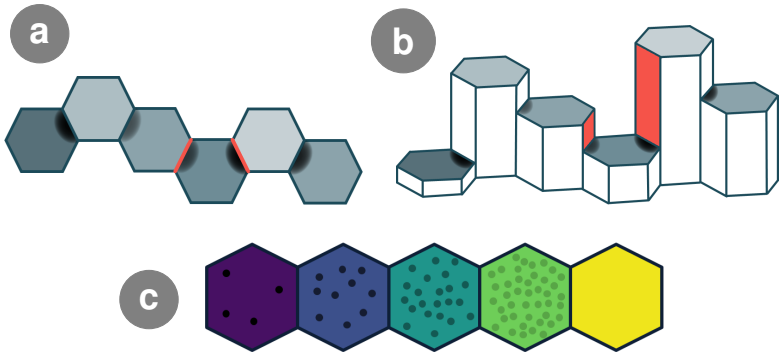

Figure 3.10: Illustrations of (a) AO emphasizing structures viewed from top-down and (b) "the side", including two red side walls that cause such an exemplary tile darkening. (c) Example of a hexplot using a viridis heat map encoding density, combined with a classic scatter plot depending on tile density. Sparse tiles without overplotting show underlying points, whereas the denser a tile becomes, the more the visibility of the underlying scatter plot decreases.

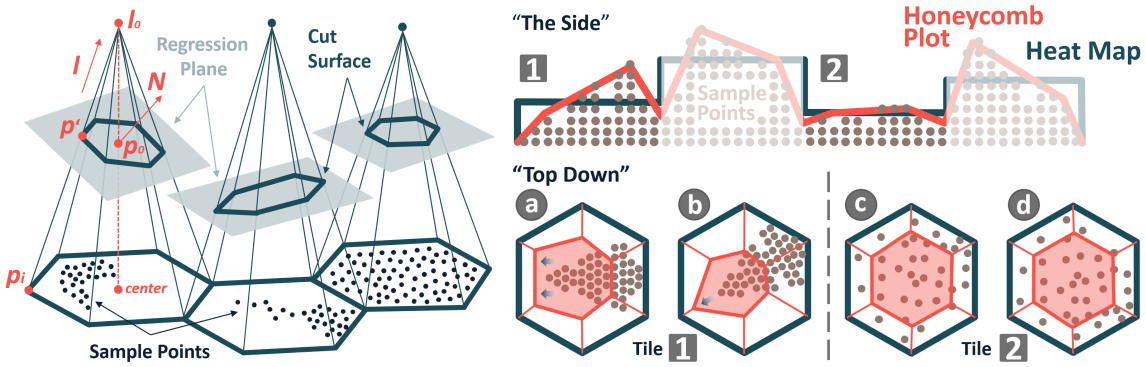

Figure 3.11: Illustration of (left) pyramid intersections with planes (tilted to the left/right and horizontally) that best approximates the underlying point density. Comparison (right) of four neighboring tiles aggregating points viewed from "the side". The heat map (blue line) corresponds to the average density, whereas the diamond cut (red line) provides per-tile density distributions. When viewed "top down" as usual, tile (1) shows two possible decreases from right to left (a, b) and tile (2) exemplifies two feasible, almost uniform, distributions (c, d).

Figure 3.12 visualizes starting positions of tornadoes between 1950 and 2019 [124] in North America. In the center of the US, there is a region with a significant number of tornadoes, the so-called "Tornado Alley", covering Texas, Oklahoma, Kansas, Colorado, Nebraska, Iowa, South Dakota, and the south of Minnesota. This is based on meteorological conditions such as cold dry air from the north, a jet stream from the west, warm dry air from south-west, and warm moist air from the south. It also explains the comparatively few tornadoes in the west, which is due to the comparably colder Pacific Ocean from which less water evaporates into the atmosphere. Tornadoes usually form over land, whereas hurricanes generally form over water, e.g., the Atlantic Ocean. It becomes clear that most of Florida's tornadoes form along the west coast between Tampa Bay and Forth Myers. Nevertheless, there is an increased number of tornadoes visible as continuous black dotted line along the coast too. This phenomenon describes so-called "waterspouts" which are extraordinary tornadoes forming over water. Waterspouts occur particularly often along the south and east coast of Florida, for example, in Miami or at Key West, and are usually only statistically covered if they affect land.

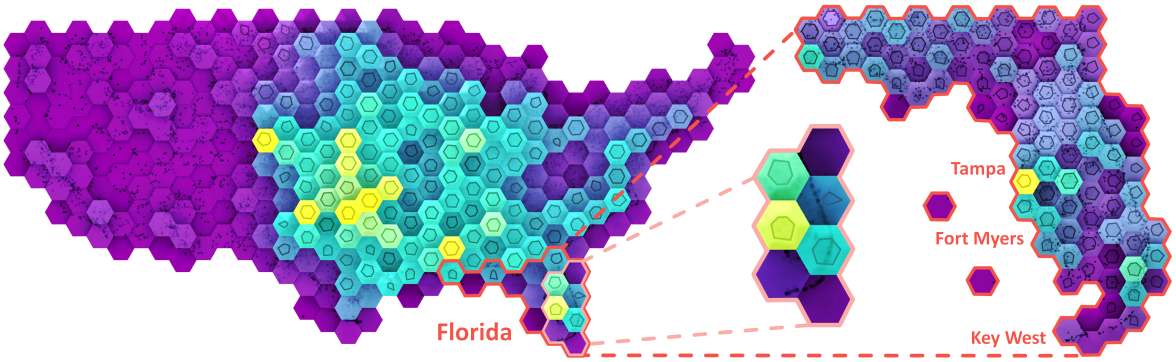

Figure 3.12: Honeycomb plot of US tornadoes between 1950 and 2019 [124].

A well-known data set, also used by the machine learning community, is the California Housing data [161] collected during the 1990 US census. It contains, among other attributes, the longitude and latitude of houses, their age, the number of rooms, the number of households and number of residents, their income, the house value, and the ocean proximity. Figure 3.13 shows a hexplot of their locations, where one point corresponds to the geographical position of a house. As highlighted, the four largest metropolitan areas are located along the west coast: Los Angeles (4 M), San Diego (1.4 M), San Jose (1 M), and San Francisco (0.9 M). A heat map without blended points, however, fails to encode how houses are arranged, such as in San Francisco along the San Francisco Bay, and may lack emphasis of regions where density only varies slightly corresponding to minor changes in colors, such as in Redding (0.09 M), still being the largest city in California north of Sacramento. Using honeycomb plots, however, both data properties relevant for analysis and exploration are subtly highlighted.

We, furthermore, show that our techniques can be implemented efficiently in real time on the GPU exploiting 2D textures. For example, Figure 3.14 shows how a Cartesian grid of texels relates to a hexagonal grid, including the local neighborhood, using

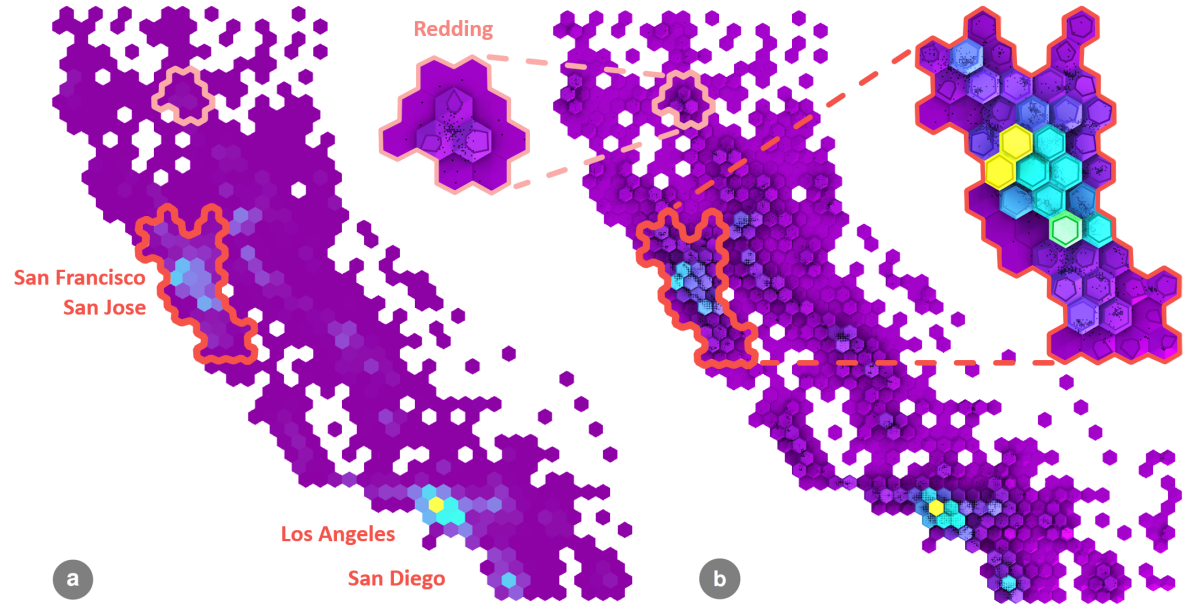

Figure 3.13: Juxtaposition of (a) a heat map and (b) a honeycomb plot of the aggregated locations of houses originating from the 1990 US census [161].

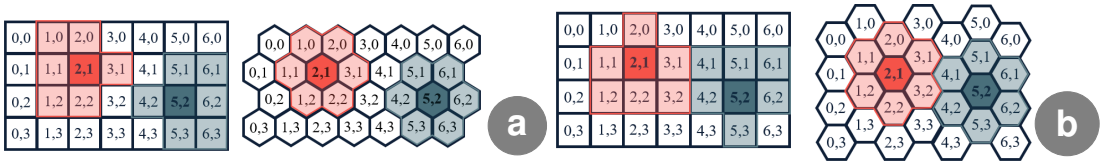

Figure 3.14: Relationship between texture texels and hexagonal offset coordinates for (a) pointy top grids as well as (b) flat top grids, respectively.

offset coordinates. Based on an additional usage example, we generated expressive visualizations and, in a user study, evaluated whether honeycomb plots can be understood within about 15 minutes. We chose a between-subject study design and had 42 participants. They performed 4 tasks derived from typical hexplot analysis, each consisting of 3 questions with varying difficulty. The study required users to find tiles with the highest or lowest density, or the steepest or flattest change in density within a highlighted region or adjacent to a highlighted tile. Our findings suggest that AO supports value estimation tasks, especially when densities are similar and non-neighboring tiles are compared. Furthermore, the results demonstrated that users understood the diamond cut and that the visualization of underlying points provided additional information without negatively affecting the execution of the tasks. Paper **B**—*honeycomb plots* contains our hypotheses, their evaluation, and explains the implementation.

3.2 Line-based Techniques

Line charts are an effective and widely used technique for visualizing series of ordered two-dimensional data points. The relationship between consecutive points is indicated by connecting line segments, revealing potential trends or clusters in the underlying data. Unfortunately, individual lines become harder to distinguish and interpret with an increasing number of lines. A popular remedy to distinguish individual lines is the use of color and standard alpha blending. However, the blending order is currently either ignored or naively used, for example, assuming it is implicitly given by the order in which the data was saved in a file. This is common practice with standard visualization tools such as Tableau. An illustrative example of this is shown in Figure 3.15. This may result in the possible naive assumption that order may not have any essential influence on the resulting visualization. However, due to the non-commutativity of classic alpha blending, this results in misleading and possibly even contradicting visualizations of the same underlying data set, a so-called "hallucinator" [96].

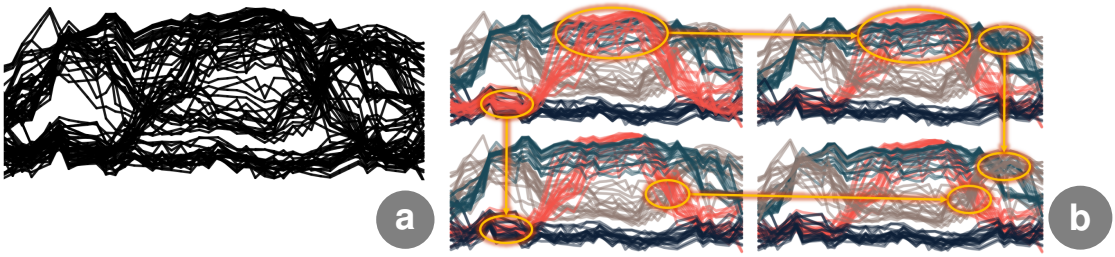

Figure 3.15: Exemplary line charts visualizing (a) the temporal changes of crops [156] and (b) the resulting rendering-order dependency once line bundles correspond to colored clusters.

Figure 3.16: Illustration of Gestalt principles applied to line intersections, making the human visual system derive a three-dimensional cubic rather than individual geometric primitives.

As indicated by the Gestalt principles, rendering order and the resulting occlusion relationship among graphical elements can have a significant impact on their perception. For example, as shown in Figure 3.16 when looking at a wireframe representation of a cube where only the edges connecting the corner points are visible. Our brain will not emphasize the individual primitives such as triangles, rectangles, or trapezoids that arise from intersections of lines but instead, it will tend to recognize a three-dimensional cube, due to an implicitly derived order of the lines.

Hybridization of Line Charts

Hybridization is not just limited to points but can equally be applied to techniques relying on related visual encodings. Line charts, for example, are a natural progression of point data, emphasizing the relationship of consecutive elements as lines. If a large number of overlapping lines form clusters, a heuristic can determine how these have to be (re)arranged according to optimal visibility based on importance, i.e., task relevance. Instead of displaying lines in the order in which they are stored in files, as it is common practice, such an approach considers the number of lines clusters consist of and how compact or wide they are, i.e., how much they might obscure other clusters. Again, a heuristic can be used to either decide globally, i.e., for entire lines or clusters, how they are layered, or locally which, in a figurative sense, creates a weaving pattern. Here, too, we exploit spatialization cues, e.g., global illumination, to highlight bundles.

In our approach, we propose to preserve occlusions and depth cues by using a blending operator that still exhibits occlusion, while providing explicit control over occlusion relationships by introducing an importance function. Our approach is based on the three requirements, shown in Figure 3.17. First, the contributions of individual lines should be independent of their order in the data set or the order in which they are rendered.

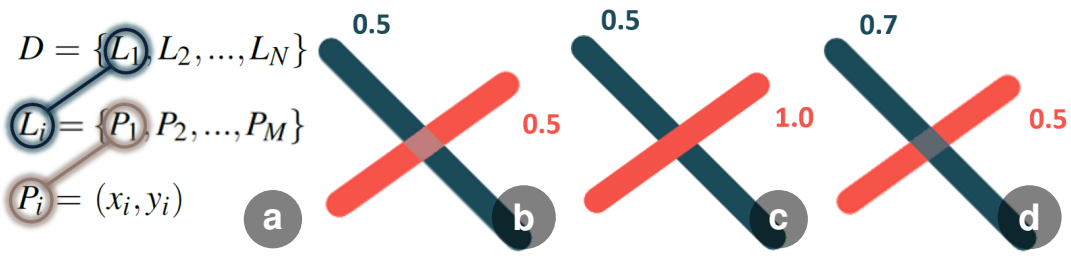

Figure 3.17: Data specification and blending results of lines with different importance.

This prevents visualizations from being dependent on render and blending order and, therefore, possibly misleading. (a) We regard a line set D of N polylines with its members L_1, L_2, until L_N represented as tuples of M ordered two-dimensional points P_1, P_2, until P_M. Second, line segments with (b) similar importance should also contribute similarly to the pixels they cover in the final image. This means that the contribution of line segments to a pixel that they both share are averaged, hence, their order is irrelevant. Third, when the importance of line segments differs significantly (c,d), the line segment with higher importance should occlude the line segment with lower importance.

Our contribution, furthermore, includes an algorithm that maximizes the visibility of the included clusters. Figure 3.18 shows a simplified example depicting (a) a parallel coordinates plot. In case each line has an associated group identifier, our goal is to reduce the amount of overdraw by assigning importance values based on the estimated amount of screen space taken up by all the lines of a group along the dependent axis of the graph. For example, the axis values of a particular group may vary considerably for one dimension but may later on fall within a much narrower range. Another group may exhibit an inverse behavior. In such a case, the importance values for each group should correspond to the estimated amount of screen space, such that groups that take up less space receive higher importance values. For each value along the independent axis of the graph, we first (b) compute the minimum and maximum values of each group on the dependent axis covered by lines belonging to this group. This allows us to determine the envelope of all lines associated with a group by connecting two subsequent intervals along the independent axis forming a trapezoid, and then compute their areas. Two scenarios may appear, (c) there are no overlaps and, therefore, ordering them will not have an impact on the resulting visualization, or (d) there are potential overlaps of the calculated cluster envelopes. In this case, we iterate over the calculated areas with the goal of establishing an ordering of all groups according to a cost function.

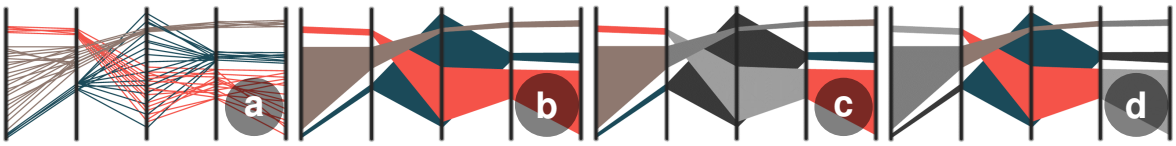

Figure 3.18: Illustration of our weaving algorithm based on (a) colored clustered lines, (b) resulting cluster envelopes, and (c) non-overlapping or (d) overlapping cluster bundles.

An illustration of this *weaving loom* algorithm, to compute the local importance values for such a weaving pattern using a greedy approach, is shown in Figure 3.19. The cost value for a group corresponds to the sum of the intersection areas with all other groups multiplied by the area covered by the group itself.

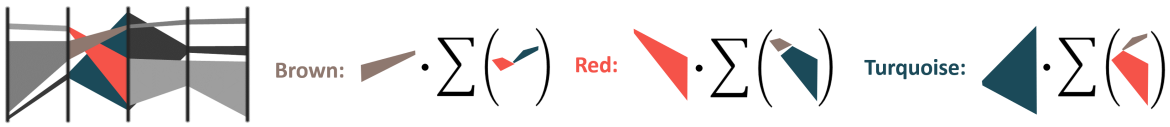

Figure 3.19: Importance calculations based on cluster envelopes and their overlaps.

An example of line weaver exploiting global importance augmenting a benchmark data set [17] for parallel coordinates plots is shown in Figure 3.20. From left to right, we

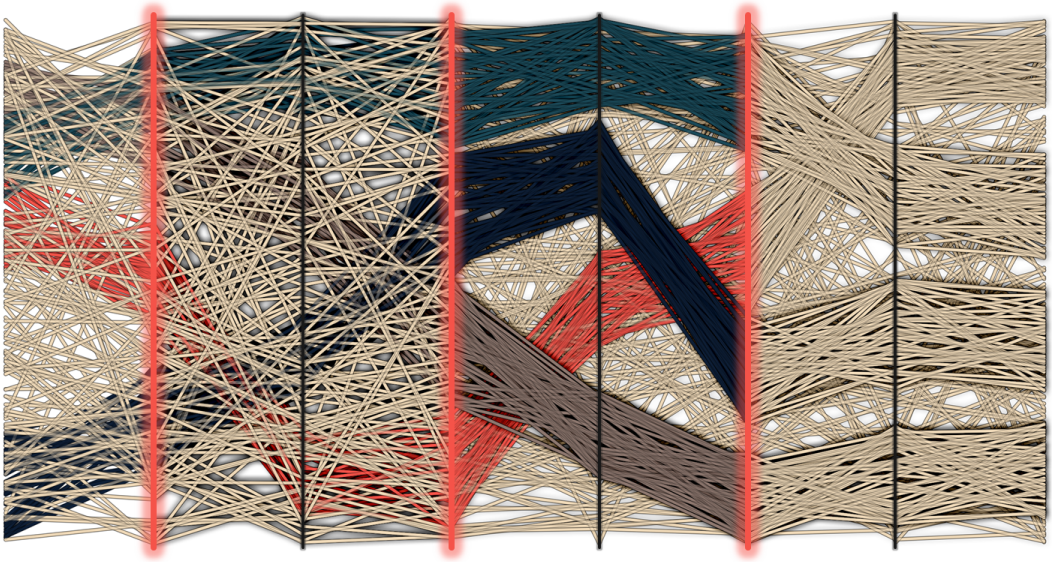

Figure 3.20: Visualization of a benchmark data set [17] consisting of four clusters: brown (50 lines), dark blue (49 lines), turquoise (48 lines), red (46 lines), and beige noise (150 lines).

compare the result of a randomized rendering order with outlines and additional highlights with halos, to our approach using an importance function depending on cluster size and arc length, including a monochrome version of it. As shown, it is difficult to distinguish clusters, even when different colors are assigned to each of them. When using a global ordering, however, the shape of clusters becomes visible even without using color. The importance of a line is influenced by the cluster it belongs to including its arc length, whereby the shortest lines are rendered on top. Another example highlighting local importance is shown in Figure 3.21. It visualizes an Andrews plot [2], sometimes also referred to as smooth parallel coordinates plot, in which each data point is defined as finite Fourier series. Using our approach, line bundles smoothly change locally resulting in a weaving pattern optimizing the visibility of each bundle. Paper **C**—*line weaver* provides all details of our contribution including additional use cases.

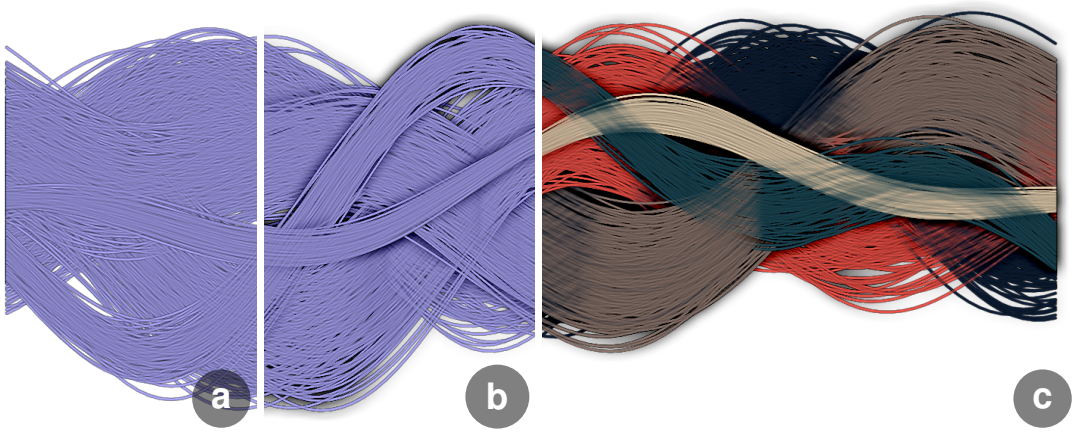

Figure 3.21: Andrews plot [2] of synthetic data (a) without or (b) with outlines and halos as well as (c) colors. The data set shows two compact beige (100 curves) and turquoise (100 curves) clusters rendered over the wider dark blue (300 curves) and brown (250 curves) clusters which are in turn interwoven with the red (250 curves) cluster.

4

Chapter 4

Conclusion and Future Work

We have presented visualization hybridization using spatialization cues for bivariate points and related line data sets. The basis of our techniques are heuristics, i.e., importance functions, that globally or locally influence the hybridization of multiple visual representations. Based on various synthetic as well as real-world data sets, we showed the benefits of hybrid vigors in comparison to standard approaches like common scatter plots, continuous and aggregated density estimations, and classic line charts. Furthermore, we explored user-interaction possibilities in hybrid approaches using importance functions, e.g., focus+context through lenses, and demonstrated that these can be easily implemented. We showed that all our techniques work in real time, as we have developed them efficiently on the GPU. Furthermore, the source code of our publications is either already available open source or we plan to do so in the near future. Nonetheless, a number of interesting research questions remains to be answered in the future.

Future work could further investigate how spatialization cues can help improving the visual perception of other visualizations, either for hybridization or to make individual techniques more preattentive. Our contribution could be expanded to combine other visual representations through hybridization and, thus, generate more widely applicable toolkits. Hybridization could be standardized and built into established tools in order to be easily accessible, also to novice users. It would, furthermore, be interesting to investigate how suitable hybrid visualizations are to be combined with one another. It could be investigated whether line weaving can be applied to spatial trajectories and then be hybridized with sunspot plots or honeycomb plots. Another aspect could be to research if circular visualization techniques such as radar charts or related diamond cuts can be interwoven if they are based on multi-class data. In general, studies could investigate how hybridization of multiple classes are possible.

We have examined heuristics and the importance functions based on them, using different metrics. Nevertheless, it would be interesting to further explore their design space. It would be interesting to examine which other inherently contained data features could be used to potentially serve as importance functions. Furthermore, machine learning could be used to improve hybridization, for example, to adjust parameters such as the choice of the right kernel size of a density estimation, optimize the grid resolution or layout, adapt the intensity of shading and ambient occlusion, find an optimal light position highlighting most surface properties, adapt the used heat map on-the-fly to emphasize data features, enable application-based blending of the individual visualization layers, etc., without having to let users change GUI parameters or slider values.

Finally, we also want to point out that the potential of "spatialization" can be implemented in the fullest sense of the word. The result could be a hybridization of classic visual exploration combined with haptic interaction through touch. This would make visualizations that are currently only available for sighted people accessible to blind users or people with vision deficiencies. In our opinion, the use of spatialization cues, which are applied to 2D information visualization, are ideal to exploit this. For example, Figure 4.1 highlights that honeycomb plots can be three-dimensionalized and fabricated directly with a 3D printer. Future studies could show to what extent tasks that are currently performed purely visually, could be implemented haptically. Furthermore, the hybridization of the material used or the combination of printed materials and tangible interfaces could be examined.

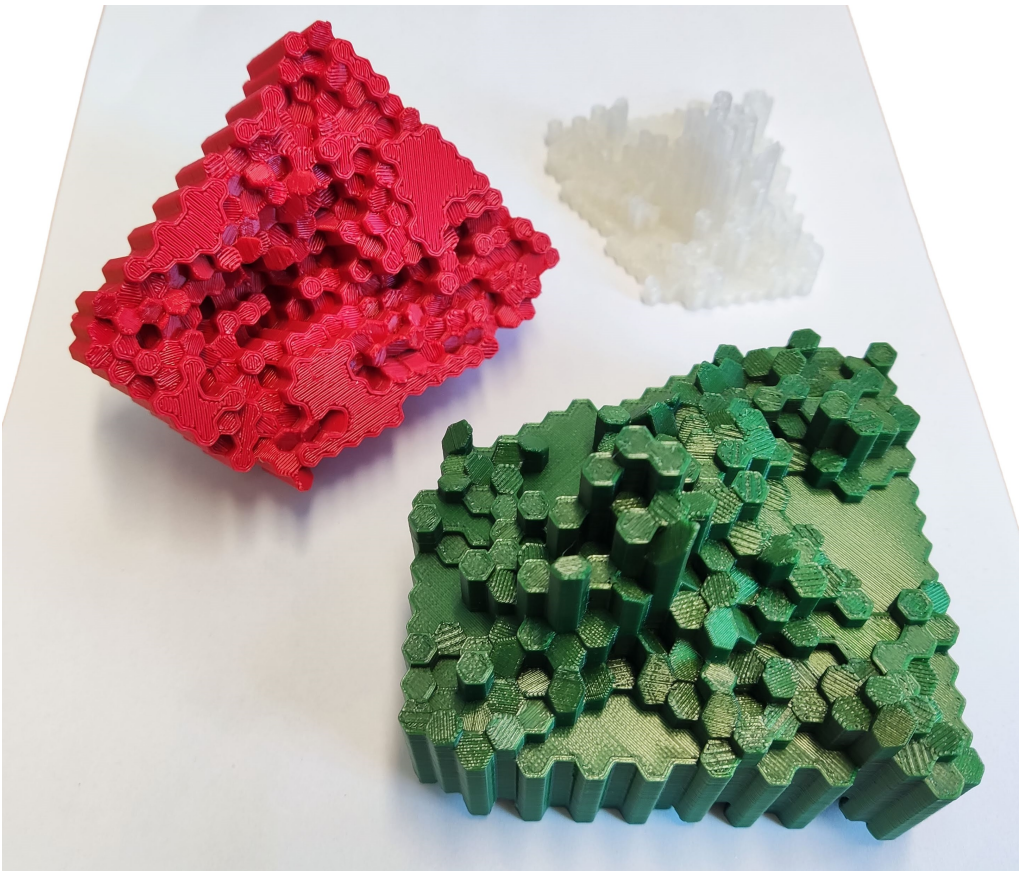

Figure 4.1: Examples of 3D printed honeycomb plots.

Part II

Included Papers

Paper A

Sunspot Plots: Model-based Structure Enhancement for Dense Scatter Plots

Thomas Trautner, Fabian Bolte, Sergej Stoppel,
and Stefan Bruckner

University of Bergen, Norway

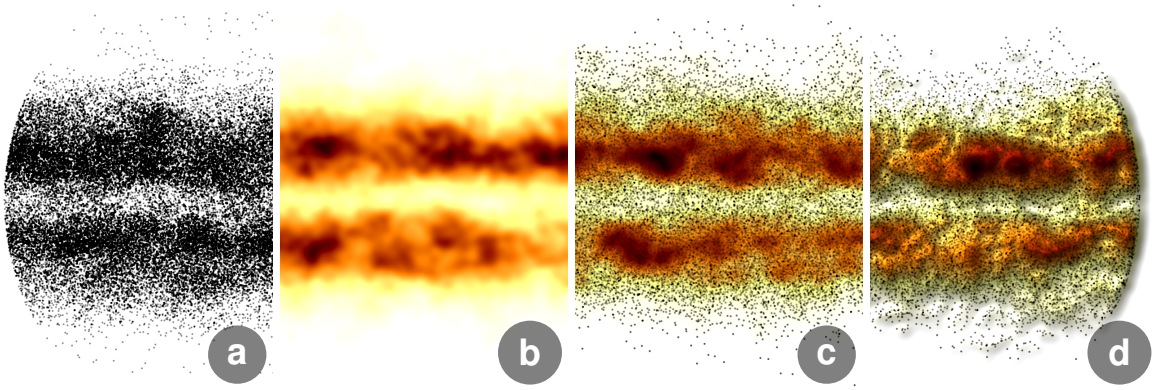

Figure A.1: Sunspot plots were inspired by the natural phenomenon of the same name visible on the sun's surface. These spots are colder and therefore darker, which is why they visually stand out from their bright surroundings. Between 1825-1867, the German astronomer Samuel Heinrich Schwabe documented the position of approximately 135,000 sunspots. When visualizing such dense data sets as scatter plots (a), it is often hard to estimate the density distribution of points due to overplotting. Density-based visualizations (b) are able to convey trends and clusters, but fail to effectively depict individual data samples. Our technique (c) is able to visualize density variations without abolishing the discrete nature of scatter plots. In addition, illumination can further emphasize local density changes, as shown in (d).

Abstract

Scatter plots are a powerful and well-established technique for visualizing the relationships between two variables as a collection of discrete points. However, especially

Computer Graphics Forum (EuroVis 2020), 39(3), pages 551–563, 2020. doi: 10.1111/cgf.14001

when dealing with large and dense data, scatter plots often exhibit problems such as overplotting, making the data interpretation arduous. Density plots are able to overcome these limitations in highly populated regions, but fail to provide accurate information of individual data points. This is particularly problematic in sparse regions where the density estimate may not provide a good representation of the underlying data. In this paper, we present sunspot plots, a visualization technique that communicates dense data as a continuous data distribution, while preserving the discrete nature of data samples in sparsely populated areas. We furthermore demonstrate the advantages of our approach on typical failure cases of scatter plots within synthetic and real-world data sets and validate its effectiveness in a user study.

A.1 Introduction

Scatter plots are a popular means of representation within various research fields. According to Friendly and Denis [61], it can be assumed that scatter plots represent 70-80% of the graphs used in scientific publications. Traditionally, scatter plots encode each data element with a marker placed on two ordered orthogonal dimensions. Simple yet versatile, scatter plots are able to effectively depict global and local regression patterns, clusters, and outliers in the data. However, due to *overplotting*, scatter plots become less effective when dealing with an increasing number of samples. As a result, global and local data variations become increasingly hard to detect. Approaches like opacity regulation, reduction of marker size, or subsampling can increase the readability to some degree. Nevertheless, these approaches still fail when the number of data samples approaches the number of available pixels. Frequency-based visualization techniques such as density estimations offer solutions for such cases. These approaches interpret the data as samples of a continuous phenomenon that can be approximated by a model, e.g., a probability density function. A popular approach to approximate the probability density function is *kernel density estimation* (KDE). However, visualizations based on density functions are not well suited to depict local variations or singular points. Especially in low density regions, discrete encoding can depict singular data points better than density-based approaches.

In this paper we introduce *sunspot plots*, a novel visualization for bivariate scattered data. Sunspot plots are named after the astronomical phenomenon that they bear resemblance with, similar to the visualization of Schwabe's sunspot observations [5], shown in Figure A.1. Our approach aims to alleviate the respective drawbacks of discrete and continuous representations. Instead of dismissing the discrete encoding of data points, we combine advantages of discrete and continuous data metaphors in one unifying visualization model. We interpret the probability density function as a surface over a two-dimensional field. Depending on the density of the underlying data, we create a blend between the continuous surface representation and the discrete point representation. Based on a GPU on-the-fly KDE, we are able to interactively display even large data sets with several thousands of data elements. Finally, using illumination sunspot plots can exploit the human mental model of surfaces to effectively convey the absolute density as well as local and global variations of the data.

The contributions in this paper can be summarized as follows:

- A novel kernel density-based visualization technique that unifies discrete and continuous representation of large bivariate data
- A user study with several hundred participants in which we compared five different visualization techniques based on two representative tasks during the analysis of bivariate data

A.2 Related Work

Sarikaya and Gleicher [141] introduced four scatter plot design classifications depending on task and data characteristics: *point encoding*, *point grouping*, *point position*, and *graph amenities*. For reasons of clarity, the following section is divided into the first three categories, omitting graph amenities.

Point Encoding: Point encoding represents design decisions that affect the appearance of individual markers of data points. For example, color, size, opacity, blurriness, outline, or the selected symbol [141]. Li et al. [105] provide studies concerning lightness, size, radii [103], and contrast [104] of different scatter plot symbols in relation to human visual perception. As part of our user study, we investigated, i.a., the effects of density-dependent color coding of individual points compared to conventional scatter plots and continuous heatmaps.

An example of point encoding are *bubble plots* [130]. They display circles instead of points, whose diameter and color encode additional data-dependent properties. Unfortunately, scaling of circle radii simultaneously increases the chance of occluding neighboring circles. An ad-hoc approach to overcome overplotting is to uniformly downscale all radii in case of occlusions. Woodruff et al. [184] extended this idea to a density-based approach that adjusts the size of each symbol depending on the density of its neighborhood. Mayorga and Gleicher [114] point out that when the data size exceeds the screen resolution, overplotting can no longer be avoided through scaling. This restriction intensifies further if the screen space is narrowed down, for example, by using *scatter plot matrices* [73] to simultaneously compare multiple scatter plots.

Apart from size, other visual properties such as color or brightness can be used to augment scatter plots. However, according to Trumbo [166], color blending is limited by the nonlinear color perception of the human visual system which may vary even among people who are not considered as color blind. This may handicap the interpretation of quantitative data. An alternative would be transparency [112, 176, 182]. *Alpha blending*, for example, represents an implicit density encoding since points are rendered semi-transparently instead of completely opaque [158]. Figures A.2(a) and (b) show a comparison between a scatter plot with completely opaque points and one using alpha blending. Nevertheless, blending is limited to a few layers and it soon becomes difficult to distinguish between varying densities. This inspired us to include a mental model of the density distribution as surface representation in our visualization, in contrast to pure point blending.

Point Grouping: Point grouping is a design concept that aggregates multiple objects and therefore abstracts data into combined local regions. A typical example is

binning [71], which reduces the number of elements to prevent overplotting. On the one hand, it supports the identification of correlations and data characteristics, but on the other hand, it hinders the location and selection of individual data points. Finding an adequate bin size can be seen as an analogue problem to finding suitable KDE kernel parameters.

HexBin scatter plots [29] are a representative example of a point grouping approach. They represent hexagon-shaped bins whose size depends on the enclosed number of observations. Dupont and Plummer [52] emphasize that hexagonal bins are more densely packed and reduce disruptive horizontal and vertical artifacts which could arise from square bins. A hybrid approach visualizing density estimations through binning as well as individual data points are *Varebi plots*[80]. They prevent occlusions using overplotting indices per bin depending on the glyph shape, screen size, and data distribution. Another example are *sunflower plots* [38], a combination of hexagonal bins and sunflower-like glyphs where the total number of flower petals corresponds to the number of binned data points. To summarize even larger numbers of elements, the background color of each bin can indicate a multiplication factor for the petals [52]. The main disadvantage of binning approaches is the actual displacement and distortion of the underlying density locations [45]. Another disadvantage of quantizing density through aggregated bins is the absence of perceptually smooth contours. Therefore, Mayorga and Gleicher [114] introduced *Splatterplots*. They combine smooth polygonal shapes enclosing dense regions, color blending for interrelations, and the explicit representation of subsampled outliers. Unfortunately, color blending could degenerate and synthesize new colors that are difficult to interpret. In contrast to color blending, we want to emphasize the perception of *shape from shading* [135] as a possible alternative. In our study, we therefore examined how typical tasks performed with scatter plots are affected by shading. Similar to Splatterplots, we use an on-the-fly KDE. Point aggregation based on KDE has proven to be a solid and elegant approach which neither destroys the layout of the underlying data nor limits the number of data points. It is therefore used in many related applications, for example, density-based node aggregation used for large graphs [191], visualization of graphs as continuous fields [169], visualization of high-density flows of streaming points [102], or the emphasis on edges using line kernels [44]. KDE is therefore not only a popular tool in statistics but also applicable in visualization.

Point Position: Design decisions regarding point position include rearranging or reprojecting the spatial position of individual points, reducing data through subsampling, or visualizing temporal changes as *animations* [33, 110]. Unfortunately, the time required to play an animation increases with the number of overlapping samples and it may be necessary to play the animation multiple times. Furthermore, Carr et al. [29] highlight that animation can help in detecting changes, but often visualizations of density differences are sufficient.

Another possible approach to prevent overplotting is through *subsampling*, in order to reduce the total number of underlying data points. A detailed analysis of clutter reduction techniques including subsampling is presented by Ellis and Dix [57]. Simple random sampling [48, 56], for example, does not require any prior knowledge of the data set but can be disadvantageous since local or global trends may disappear, sampling artifacts could be introduced, or important outliers may not be preserved.

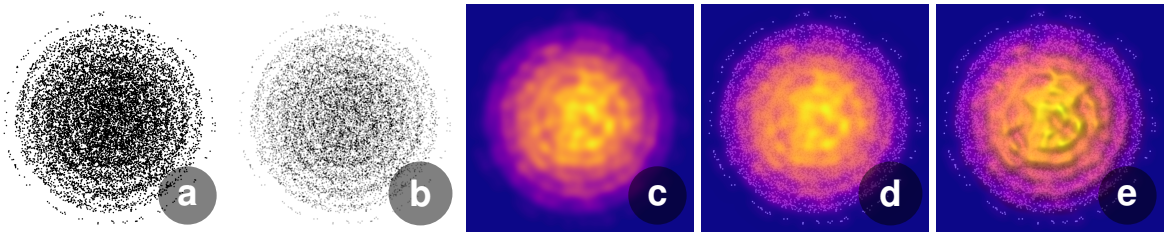

Figure A.2: Overview of currently well-established visualization techniques for bivariate data, which lay the foundation for our sunspot plots: (a) Classical scatter plots that can be improved using (b) alpha blending. (c) A visualization of a KDE using a continuous heat map representing the foundation of (d) sunspot plots depicting density information and singular data points (e) including the possibility to emphasize local variations using additional illumination cues.

More advanced approaches are *adaptive* or *stochastic subsampling* [15, 32]. These approaches aim to reduce the density in highly occluded regions while preventing further decrease in density of sparse regions, which would result in information loss. Examples are *best uniform sampling* and *non-uniform sampling* [15] which are both based on perceptual user studies, and *multi-class sampling* [32, 35] that preserves point distributions. Ellis et al. [55] present a user-controlled sampling lens enabling different sampling rates inside and outside the lens. We consider such *focus+context* functionality to be complementary and provide this functionality through an optional *magic lens* in our visualization. Figure A.2(c) shows the result of a KDE visualized using a continuous heat map, and Figures A.2(d) and (e) depict sunspot plots, without and with illumination cues, respectively.

Other possible approaches to reduce overplotting are *displacement techniques* [90, 92, 134]. Their advantage is that all individual points remain visible to the user. An example is *circular pixel placement* [91]. In case of an occlusion, the circular region around a data point's pixel position is searched and subsequently changed to the next free pixel closest to the original position. This approach has been further developed by Janetzko et al. [86] who introduced *ellipsoid point placement*. It performs an initial clustering and *principal component analysis* to generate ellipses whose orientation and aspect ratio visualize the local correlation of the data. Another possibility is adding *jitter* or marginal *random noise* [31, 168]. Instead of placing data points exactly above each other, minimal variations along the x and y axes are added. Disadvantages of these displacement techniques are the still limited number of available pixels on screen and the falsification of the original point positions which may obscure important data properties. Additionally, arbitrary visual patterns independent of the data, e.g., merging circular structures, are introduced.

Filtering operations [23, 167] aim to reduce the underlying data to the most relevant, meaningful, or significant selections through user input. Filtering already requires a certain basic knowledge of the data, thus is not suitable for untrained users. A representative example is the *magic lens filter* [152]. Similarly, *distortion techniques*, such as zooming operations (uniform) or *fish-eye views* [25] (non-uniform), allow decluttering and a focus on interesting data regions. Tominski et al. [160] provide a detailed survey of visualization techniques using lenses. Carpendale et al. [27] emphasize the importance of informing the user about the spatial distortion, for example, by using su-

perimposed grid lines or adequate shading techniques. Similar to filtering, these techniques are user-dependent and therefore particularly affect inexperienced or untrained users. This is why we prefer either automatic scatter plot designs [117] or visualizations with generally chosen parameters applicable to various use cases.

Finally, we want to discuss multidimensional approaches. Dang et al. [45] present *stacked plots*, a three-dimensional visualization approach that stacks data points in case of occlusions. Unfortunately, adding a third dimension introduces further problems such as occlusions depending on the viewing direction, perspective distortions, and difficulties in interpreting size and distance which may require additional user interaction. A multidimensional focus+context approach is presented by Staib et al. [149]. They introduce implicit filtering through continuous *depth-of-field* techniques. Bachthaler and Weiskopf [8, 9] present *continuous scatter plots*, a parameter-free continuous density plot that is capable of visualizing multidimensional input data defined on continuous domains. Similar to our approach, continuous density plots scale well with increasing data size, preventing problems with overplotting, but do not preserve the discrete data properties in sparse regions. Lastly, we would like to draw attention to the inspirational work of Sprenger et al. [148], which demonstrated the usefulness of hierarchical clustering with the help of implicit surfaces, and the work of van Wijk and Telea [170], which enriched visualizations of scalar functions of two variables by adding ridges.

A

A.3 Sunspot Plots

Overplotting is a typical problem of scatter plots even when displaying only moderate amounts of data, as shown in Figures A.1(a) and A.2(a). A common strategy to increase the readability of scatter plots is opacity reduction of individual points. We show results of this strategy in Figure A.2(b) – although some parts of the data are depicted clearer, overplotting is still present. Frequency-based visualization techniques are a common solution for large scattered data sets. Typically, density distributions are conveyed via *heatmaps* [141]. As shown in Figure A.2(c), these are not very well suited to depict small variations or outliers. Since the applied color has a strong influence on the perceived change of the density function, it is important to choose a perceptually uniform color map such as *Plasma* [82].

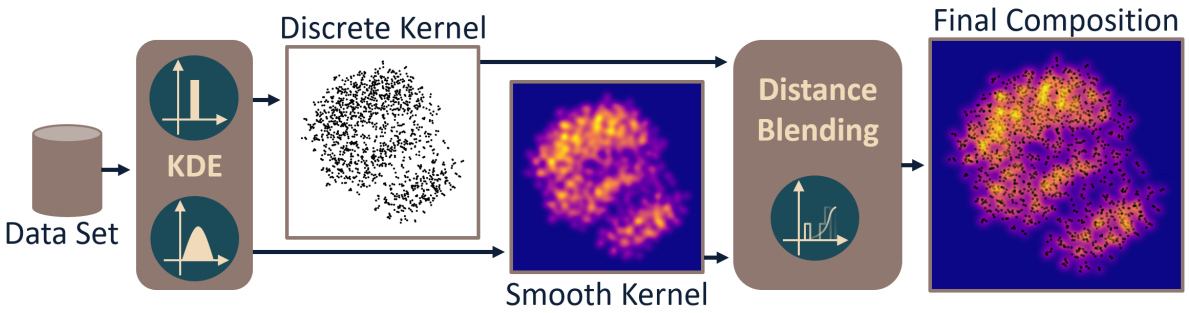

Figure A.3: Overview of our visualization system pipeline: Initially, we compute a discrete and a continuous representation of the data. Subsequently, both representations are composed into a unified visualization emphasizing continuous and discrete phenomena.

The goal of our approach is to support the exploration of bivariate scattered data by providing the best suited visual encoding depending on the local data density. We furthermore aimed for a technique that can be easily integrated into already existing visualization systems. One key aspect of our technique is an importance function α that regulates a smooth transition between a discrete encoding in sparse regions and a density-based surface encoding in data-rich regions. Figure A.3 shows a general overview of our method. We simultaneously interpret the data set as a discrete and a continuous phenomenon. The discrete aspects are depicted via a conventional scatter plot. In order to capture the continuous aspects of the data, we interpret the underlying data distribution as a two-dimensional field that we approximate through a model. While there are countless models to approximate the data distribution, we use an on-the-fly KDE to achieve interactive performance. In the final step of our pipeline, the discrete and continuous models are blended based on the importance function α. In the following, we provide a detailed description of our approach.

A.3.1 Kernel Density Estimation

Bivariate data points can often be seen as samples of a continuous phenomenon, i.e., a *probability density function* (PDF). Assuming the data is described through a PDF, the data distribution can be modeled through *kernel density estimation* (KDE). KDE is a non-parametric approach to approximate the PDF from a finite number of samples, where the PDF model $\hat{f}(x)$ at point x is computed through addition of kernels at the positions of each data sample:

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{A.1}$$

$$K_{disc}(u) = \begin{cases} 1, & \text{if } |u| \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{A.2}$$

$$K_{cont,\sigma}(u) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{u^2}{2\sigma^2}}, & \text{if } |u| \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{A.3}$$

wherein K is the kernel function, h is the kernel bandwidth, n is the number of the individual samples x_i, and σ defines the width of the Gaussian kernel. *Silverman's rule of thumb* [145] suggests a globally optimized bandwidth parameter h, but due to the strong data and task dependency, we nevertheless allow h as well as σ to be adjusted by the user. In addition, it is possible to have different bandwidths h for K_{disc} and K_{cont}. Naturally, the properties of the reconstructed PDF highly depend on the kernel chosen. Conventional scatter plots, for example, can be seen as reconstructed models using discrete kernels with constant intensity, as sketched in Equation A.2. Smooth kernels, on the other hand, create continuous surfaces that are well suited to describe patterns in the data distribution. The normal distribution function, as described in Equation A.3, is a common choice for a continuous kernel. As we illustrate in Figure A.4, not all kernels are equally suited for all data distributions. Smooth kernels, such as normal distribution kernels, are better suited to reconstruct the data distribution within dense data regions [114]. On the other hand, discrete kernels are well suited for low density

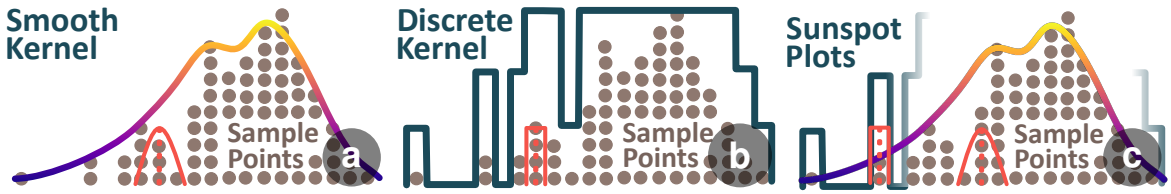

Figure A.4: (a) Smooth surface representations are well suited to depict the general data distribution, but badly suited to depict single data points. (b) Discrete encodings are opportune for the visualization of singular data samples but can reach an opacity threshold and exhibit overplotting. (c) We utilize the strength of both approaches to convey the data probability function in dense and sparse regions.

areas, as they are better able to convey outliers and their position than continuous representations. In our approach, we aim to reduce the visual complexity of the visualization by smoothly changing the kernel choice based on the local data complexity instead of adapting the bandwidth. In contrast to Splatterplots [114], we do not make a strict separation between the discrete and aggregated data interpretation. Instead, sunspot plots smoothly blend between the continuous and discrete representation based on an importance function α. There are various possible candidates for such an importance function. To reduce the computational complexity, we construct α by normalizing the already computed PDF $\hat{f}$ at point x as:

$$\alpha(x) = \frac{\hat{f}(x)}{max\left\{\hat{f}(t) \mid t \in \mathbb{R}\right\} - min\left\{\hat{f}(t) \mid t \in \mathbb{R}\right\}}. \tag{A.4}$$

As a result, α can be directly used as opacity while blending. This way, dense regions are aggregated to a continuous representation, while regions with low density are depicted through a discrete representation. The typical exploration of dense scatter plots requires *panning* and *zooming* in image space. When zooming, the ratio between the data samples and available image space changes. Our approach inherently accounts for this by fixing the kernel bandwidth within the image space. Hence, we smoothly transition the visual encoding from a surface representation to a discrete plot. An example of how the aggregation becomes more detailed as we zoom in on the data is shown in Figure A.5.

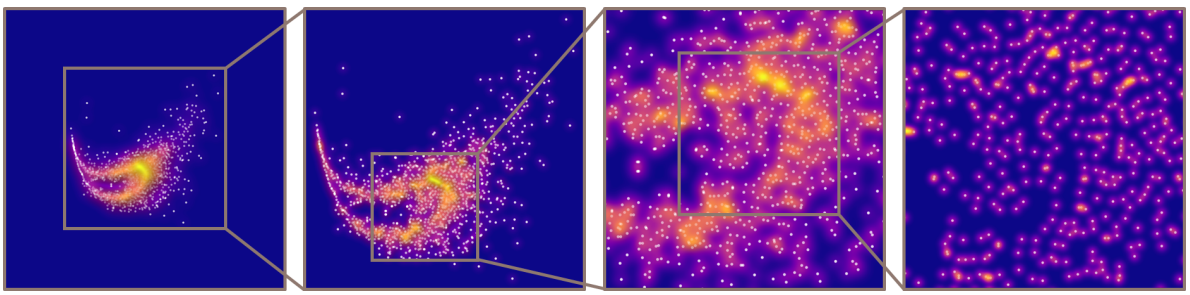

Figure A.5: The aggregation level of sunspot plots automatically adapts to the zoom level. From left to right we zoom into the highlighted areas. It can be seen that the aggregation becomes more and more detailed as the zoom level increases.

A.3.2 Surface Properties

As mentioned earlier, the probability density function over a two-dimensional field can be naturally interpreted as a surface. The field of *computer graphics* has a long tradition in effectively rendering three-dimensional surfaces. It is therefore reasonable to investigate well-established techniques for truthful and easy-to-understand surface representations. Already in 1950, Gibson [63] discussed aspects that affect our shape perception of surfaces. These attributes have a direct relationship to data properties captured with a PDF:

Distance to surface relates directly to the surface height, i.e., absolute values of the probability density function. When using surface representations for bivariate data, a color encoding is the most effective way to convey absolute function values [162].

Quality and orientation of slant, or slope, corresponds to changes of the PDF. Gibson [63] furthermore mentions the shape at a given slant, i.e., the curvature of the surface, as an important visual cue. The curvature depicts the second derivative of the PDF. Both, the slope and curvature, are conveyed effectively using the *Phong illumination model* [129].

Impression of contours arises when surface parts are separated from each other or when surface areas are not well accessible to light. The nature of the PDF does not allow for discontinuity of surfaces, therefore we only need to account for surface accessibility. A measure that is equivalent to surface accessibility is the measure of *ambient obscurance*, which is the amount of ambient light that is stopped from reaching a point. Ambient obscurance, first introduced by Zhukov et al. [190], is a technique for natural-looking lighting effects with a small fraction of computational costs, compared to global illumination. In other words – obscurance measures, for each point on the surface, the degree to which its hemisphere is obscured by neighboring surfaces. In this paper we use scalable ambient obscurance, introduced by McGuire et al. [115], to efficiently approximate the ambient obscurance of the surface.

Naturally, illumination affects the colors of the rendered surface, with slightly darker colors in shadow regions and brighter colors in areas facing light. However, since the human perceptual system is processing global structures and not only the local color information, as for example shown by the Adelson's *Checker-Shadow Illusion*[159], we postulate that using shading with sunspot plots will not have a negative effect on the perception of the surface properties, but might even yield an advantage over solely color-based sunspot plots. This assumption and other hypotheses were tested in an extensive user study which is described in detail in Section A.6.

A.3.3 Blending Function

As already mentioned, color is the most effective way to convey the surface height or, in our case, the absolute PDF values. To emphasise the discrete nature in sparser regions, the discrete kernel is using a constant color. A natural question that arises when combining two visual stimuli is the choice of the blending function. Following the reasoning of the importance function α, as sketched in Equation A.4, we blend the two stimuli using the *Porter-Duff* [132] *over* operator.

The blending of **A over B** is defined as follows:

$$C = \frac{1}{\alpha_C} \left(\alpha_A A + (1 - \alpha_A) \, \alpha_B B \right) \tag{A.5}$$

$$\alpha_C = \alpha_A + (1 - \alpha_A) \, \alpha_B \tag{A.6}$$

wherein C is the final pixel color, A the continuous, and B the discrete stimuli. In our case, α_A corresponds to the importance function α as computed from the normalized PDF and α_B can be considered as constant value 1. The background color can be chosen arbitrarily, but should preferably be either a neutral color, such as white, or the color depicting the lowest value in the density plot. We chose the lowest density value for all of the Figures shown in this paper.

A.4 Implementation

Our aim was to implement sunspot plots as interactively as possible, allowing the user to adjust individual parameters, such as the bandwidth h or the width of the Gaussian kernel σ, in real-time. The underlying data should thus be easy to explore and interpret. In order to achieve this, the presented approach was implemented as a framework based on C++ and OpenGL. Apart from the most necessary operations, such as loading and processing *comma-separated values* (CSV) files as input, all calculations were written in GLSL and performed in parallel on the GPU. An imported CSV file represents a table of data values from which the user can select two columns. This selection is done using *ImGui* [42], a *graphical user interface* (GUI). These columns are then mapped to the x and y axis. In addition, the GUI provides sliders to dynamically adjust parameters such as opacity of the surface model, scale of the samples, or the used bandwidth of the KDE. The visualization is then immediately updated and provides visual feedback based on the changed parameters. This allows the user to analyze diverse data sets and generate expressive visualizations. In the following, we will present our implementation consisting of four *render passes*.

First render pass: We start our visualization pipeline by rendering a scatter plot using the discrete kernel K_{disc}. Therefore, all scattered points are positioned according to their x and y coordinates, scaled and colored depending on user settings. While rendering each point, we additionally perform an implicit opacity modulation using an *additive blend function*. To prevent rasterization artifacts while splatting, we further perform *anti-aliasing* by smoothly fading out the outlines of the sample points.

(Optional) Pilot KDE: This optional computation is performed only if the user prefers an adaptive KDE with local bandwidth instead of a KDE with constant global bandwidth. In case of an adaptive KDE, a comparably smaller kernel is used in regions with higher density than in sparser regions. This adaptability requires a pilot KDE with constant bandwidth which can be used to adapt the kernel depending on the estimated density from the pilot study.

Second render pass: This render pass performs the continuous (adaptive) KDE, which is similar to the computation from the first render pass. They differ, however, in

that now the radii of the points are increased. This is necessary for a smooth evaluation of K_{cont}. As usual, the increased radii represent the bounding geometry of the Gaussian kernels for a fixed cutoff value where their contribution is considered to be zero. Next, the density function is evaluated per pixel. Again we use additive blending, but this time to sum up the density contributions of individual fragments.

Third render pass: Based on the results of the second render pass, we now calculate the actual surface model. Here we take advantage of the fact that the height of the surface directly corresponds to the accumulated density values. During this rendering stage, we additionally store the minimum and maximum values of the calculated PDF which is necessary for its normalization.

Fourth render pass: Here we apply the perceptually uniform Plasma color map from *Matplotlib* [82] to the calculated density range and blend it with the scatter plot using the above described blending and importance function. Optionally, local Phong illumination [129] and ambient occlusion [115] can be additionally applied. When using illumination, a single point light is positioned "top-left" according to the recommendations from Wambecke et al. [178] using an azimuth of 120°, elevation of 45°, and the five-fold distance of the diameter of the object in the center.

We plan to make our stand-alone framework available to the community under a permissive license to allow for extensions and to simplify the integration into existing applications.

A.5 Usage Examples

In this section, we demonstrate the strength of sunspot plots based on three real-world data sets from different scientific fields, illustrating the applicability of our approach for large data sets with varying data density. All examples show possibilities of how global trends or clusters and individual points or outliers can be explored. For an easier comparison, we again used the *Plasma* color map [82] in all our examples.

A.5.1 Boston Marathon

As our first example, we will analyze the 2014 Boston Marathon data set [119] including the results of approximately 32,000 participants. A classical scatter plot using alpha blending including a polynomial trend line is shown in Figure A.6(a). The x-axis of the plot encodes the unique *bib number*, which is assigned to each participant depending on the qualification time. The faster he or she was, the lower the number and the earlier the marathon starting time. The y-axis encodes the total time in hours that the participant needed to finish the complete marathon.

Next, we will visualize the same data set using sunspot plots. The result is shown in Figure A.6(b). Similar to scatter plots but in contrast to classical heatmaps alone, individual sample points from sparse regions are preserved. An essential example is the group of runners with the lowest bib numbers and a race time between 1 and 2 hours. These are the participants of the *wheelchair race* including the South African participant Ernst F. Van Dyk who won the race with a time of 1:20:36. In this example, sunspot plots indirectly show the approximation of the non-linear trend using their den-

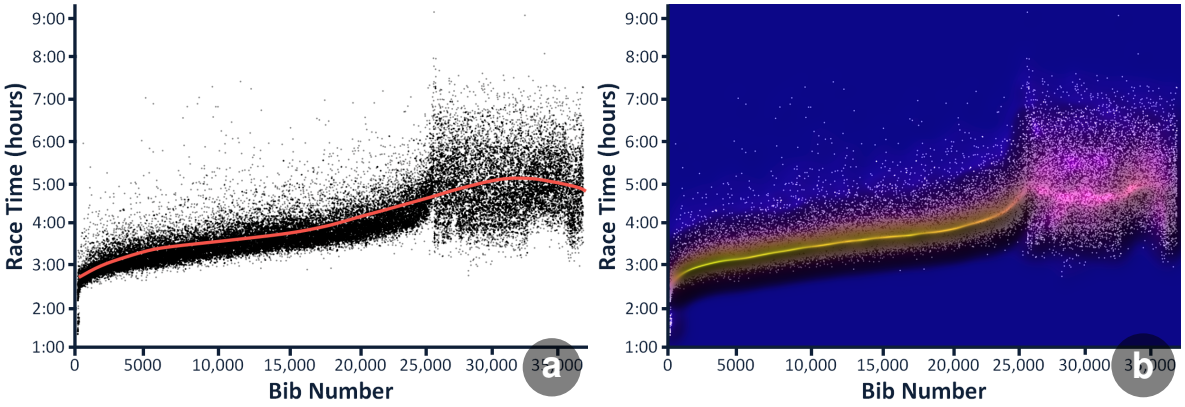

Figure A.6: (a) Scatter plot of the 2014 Boston Marathon including a red trend line of degree four, calculated using the method of least squares. Notice the clearly different point distribution and trend of the last starter wave, beginning with the bib number 27,000. (b) Example of a sunspot plot that indirectly approximates the best fit polynomial visible as ridge line between bib number 0 and 25,000, and a race time between 2 and 5 hours.

sity encoding. Comparing Figure A.6(a) and (b), it becomes visible that sunspot plots clearly show that the trend can be well approximated between a bib number of 0 and 27,000 but is only fuzzy and less meaningful between 27,000 and 36,000.

A

A.5.2 World Cities

As second example, we will analyze a data set obtained from the GeoNames database [19]. This data set contains the geo-spatial locations of cities with more than 500 inhabitants worldwide accounting for 189,280 entries in total. The geo-spatial information can be used to explore the agglomerations of cities and identify areas that may require infrastructure extension or reveal environmental conditions that influence the cities' development, such as mountain chains or valleys. In the following, we will focus on insights that can be revealed using sunspot plots. Figure A.7 shows a close-up of central Europe including several highlighting structures.

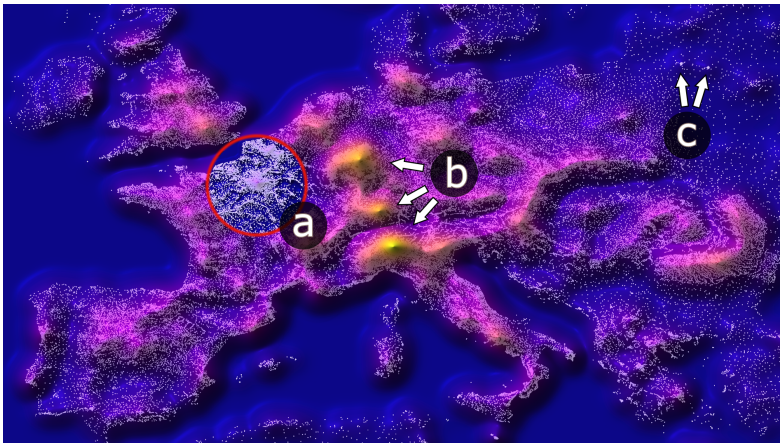

Figure A.7: Geo-spatial data of cities with more than 500 inhabitants in Europe. We have annotated a typical use case of a magic lens (a), densely populated metropolitan areas (b), and examples of sparse features (c) which are all discussed in detail in Section A.5.2.

Next, we will discuss the highlighted examples from Figure A.7. Annotated by (a), a practical use case of the magic lens becomes apparent, revealing the exact catchment areas and suburbs of Paris. It uncovers inhabited areas and the resulting large-scale road networks in northern France. At letter (b), we can observe larger clusters around Frankfurt, Zurich, and Milan. It simultaneously shows that the cluster around Frankfurt is more widely populated than the concentrated area around Zurich. Because of the Alps, the area between Switzerland and Italy is only sparsely populated. The ridge-like cluster in Italy is particularly elongated and extends from Turin in the west to Venice in the east. The two arrows from letter (c) point to the largest cities in Lithuania, Vilnius and Kaunas. Both are, in contrast to pure density encoding, clearly visible although the surrounding country is evenly but very sparsely populated.

A.5.3 t-SNE of Handwritten Digits

As third example, we will analyze the test set of the MINST database of handwritten digits [101]. It contains 10,000 28-by-28 grayscale images of labels from 0 to 9. After an initial dimensional reduction from originally 784 to 50 dimensions, using a *principal component analysis* (PCA), their dimensions are further reduced to two using the *t-SNE Barnes-Hut algorithm*. The resulting sunspot plot of similar point clusters is shown in Figure A.8(a).

In the following, we want to justify our design decisions by investigating how sunspot plots from Figure A.8(a) would change if one were to naively blend the computed heat map with a scatter plot. This comparison including a heat map and scatter plot alone is shown in (b). It seems that there are already occlusions that complicate the interpretation although this data set is comparatively small. It is no longer possible to perceive the density of regions that contain sample points, but only that of regions without. It is therefore no longer possible to interpret the actual underlying data. This negative effect would be even worse if more points were to be visualized.

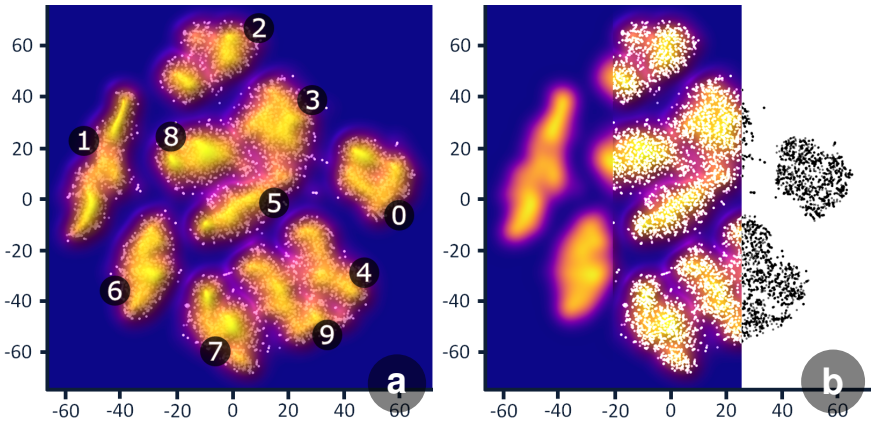

Figure A.8: (a) Visualization of the t-SNE result of handwritten digits using sunspot plots. We have additionally annotated the implicitly created clusters with the correspronding numbers from 0 to 9. (b) Overview of three different visualization techniques: (left) a heat map that neglects individual samples, (middle) naive blending of heat map and scatter plot where overplotting hides the actual density encoding, and (right) a scatter plot alone which complicates the detection of cluster centers.

A

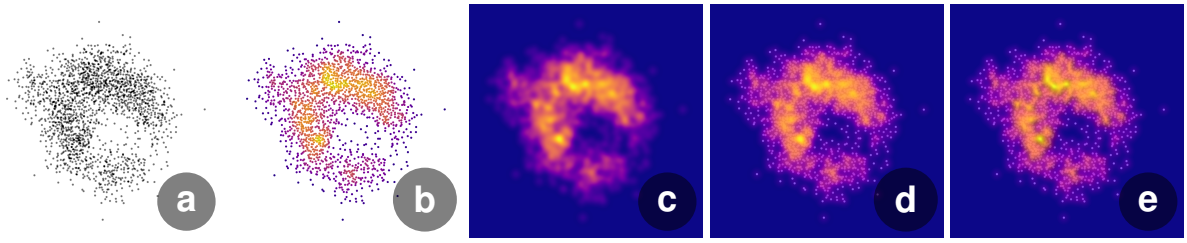

Figure A.9: Overview of the five visualization stimuli used in our user study: (a) A classical scatter plot with opacity modulation to encode overplotting. (b) A scatter plot with density-dependent color coding of individual points. (c) A smooth density representation using a continuous heatmap. (d) Sunspot plots and (e) sunspot plots including Phong illumination [129] and ambient occlusion shading [115].

A.6 User Study

One of the main concerns with sunspot plots is that the blending of individual points and density representations using an importance function might still negatively influence the user perception of absolute and relative density values. We therefore conducted a controlled user study to analyze the human visual perception dealing with dense scatter plots, overplotting, and density estimations. We compared well-established techniques like scatter plots using common opacity modulation or color encoding, continuous heatmaps, and our approach in a regular and an illuminated alternative. In total, we analyzed five different visualization stimuli, as shown in Figure A.9. The used questionnaires and data collected during the study are included as supplemental material.

A.6.1 Hypotheses

We utilize the classical scatter plot with alpha blending as base line for density encoding visualizations. One could argue that the easiest way to encode density would be to color individual sample points by their specific density value. Sarikaya and Gleicher [141] suggest that *point encodings*, as in colored scatter plots, would not improve density comparison tasks. Another approach would be *point grouping* as in continuous heatmaps. According to Sarikaya and Gleicher [141], this would improve the perception of density values and hence support comparisons. We therefore use colored scatter plots to gather empirical evidence for the statement that *point groupings* provide better support than *point encodings*. *Point groupings*, however, hide individual sample points. Since sunspot plots combine scatter plots and heatmaps, we aim to examine whether the blending of discrete points and a continuous heatmap introduces drawbacks in density estimation tasks. Finally, we aim to investigate if the introduction of illumination improves the perception of surface properties or if it leads to undesired side effects. We therefore state the following hypotheses:

H1: Density-dependent color coding of individual scatter plot points increases the users' accuracy of estimated density values compared to scatter plots with alpha blending alone.

H2: Sunspot plots do not perform worse than continuous heatmaps for absolute and relative density estimations.

H3: Applying an illumination model in combination with ambient occlusion shading to sunspot plots improves the users' accuracy when estimating density values.

A.6.2 Experiment Design and Tasks

We designed an experiment in which participants are shown images of a given visualization technique and asked to answer one question regarding the data density per image. The study tested one independent variable, the visualization type, with five levels: *scatter plot* (SP), *colored scatter plot* (CSP), *continuous heat map* (CHM), *sunspot plot* (SSP), and *shaded sunspot plot* (SSSP). We followed a between-subject design to avoid a learning bias by assigning one visualization type to each participant.

The study consisted of two analysis tasks: *assign* and *compare*. We adhered to Sarikaya and Gleicher's list of *abstracted analysis tasks* [141] performed on scatter plots and considered one task from each of their defined task types as most relevant:

- **object-centric:** task 4 - *object comparison*
- **browsing:** task 5 - *explore neighborhood*
- **aggregate-level:** task 11 - *density comparison*

In the *assign* task, users had to estimate the density value from the bounded interval [0,100] in a highlighted region within the visualization. The correct value was computed using regular KDE, where a value of 100 corresponds to the densest point in the data and 0 to the sparsest. The *error* was computed as the difference between the participant's answer and the averaged per-pixel density within the highlighted region. In the *comparison* task, we highlighted three regions (A,B,C) and asked users which region contains either the highest or lowest density. Possible answers were A, B, C, and all three regions had the same density. We assured participants that there was exactly one correct solution. The answers were therefore evaluated as a binary, correct or wrong.

Each participant was asked 12 questions, 6 per task type, each showing an image of a defined visualization type applied to a different data set. In order to minimize data bias, we selected a total of 84 data sets with varying densities from recent publications by Abbas et al. [1] and Aupetit et al. [7], analyzing cluster patterns in scatter plots. In each data set we highlighted 4 different sets of density regions, 2 per task type, leading to a total of 336 images per visualization type. Each participant saw 12 of these images (2 each of sparse, moderately sparse, and dense data set in each of the two tasks). This ensures that on average, all data sets are displayed with similar frequency throughout the study. We randomized both the order of tasks and the order of questions within each task block to reduce potential learning effects. Participants provided answers via numeric text entry, had no time limit, and were told to come up with their best estimate. To assess the user performance we computed the *average error* for the *assign* task and the *average percentage of correct answers* for the *compare* task.

A.6.3 Participants and Procedure

The experiment was performed as an online user study using *Amazon Mechanical Turk*. We assigned 70 participants per visualization type resulting in a total of 350 partici-

pants, of which 260 (SP: 56, CSP: 52, CHM: 54, SSP: 54, and SSSP: 44) passed the attention checks. Participants (169 male, 90 female, and 1 with no answer) ranged from 18 to 68 years. We determined whether participants were wearing glasses (167 without glasses, 92 with glasses, and 1 with no answer) or were suffering from color vision deficiencies (245 without deficiencies, 8 reporting a deficiency, 6 reported that they do not know, and 1 with no answer).

At the beginning of the study, users saw a set of instructions based on their assigned visualization type. Next, they had to carry out two different task blocks (*assign*, *compare*), each consisting of 1 example, 1 practice trial, and 6 questions. When answering the practice trial, users were shown the correct answer including an explanation for how the answer was obtained. Each task block contained a simple attention check to filter out inattentive participants. At the end, participants had to answer general questions about the study and about themselves.

A.6.4 Study Results

We conducted a one-sample *Kolmogorov-Smirnov* test for each task type revealing that the underlying data are not normally distributed at the 5% significance level. We therefore used the *Kruskal-Wallis* non-parametric test indicating that the visualization stimuli differ significantly (*assign*: $\chi^2(4) = 171.35$, $p = 5.37 \times 10^{-36}$ and *compare*: $\chi^2(4) = 68.64$, $p = 4.38 \times 10^{-14}$). We rejected the null hypothesis assuming that all visualizations had similar effects and conducted *Wilcoxon* rank sum tests for pairwise comparisons. The detailed study results of the *assign* task and the *compare* task are shown in Table A.1. Individual significant results from both task types were additionally highlighted.

Stimuli:	SP vs. **CSP**	CSP vs. **CHM**	CHM vs. SSP	SSP vs. SSSP	CHM vs. **SSSP**
Participants:	108 (56, 52)	106 (52, 54)	108 (54, 54)	98 (54, 44)	98 (54, 44)
Assign	$p = 2.81 \times 10^{-4}$ *	$p = 1.15 \times 10^{-13}$ *	$p = 0.161$	$p = 0.251$	$p = 0.014$ *
Posthoc Test	$\widetilde{m}_{SP} = -16$	$\widetilde{m}_{CSP} = -6.5$	$\widetilde{m}_{CHM} = 1$	$\widetilde{m}_{SSP} = 1$	$\widetilde{m}_{CHM} = 1$
(*Wilcoxon*):	$\widetilde{m}_{CSP} = -6.5$	$\widetilde{m}_{CHM} = 1$	$\widetilde{m}_{SSP} = 1$	$\widetilde{m}_{SSSP} = 0$	$\widetilde{m}_{SSSP} = 0$
Compare	$p = 1$	$p = 1.54 \times 10^{-7}$ *	$p = 0.663$	$p = 0.104$	$p = 0.271$
Posthoc Test	$\widetilde{m}_{SP} = 50\%$	$\widetilde{m}_{CSP} = 50\%$	$\widetilde{m}_{CHM} = 83.3\%$	$\widetilde{m}_{SSP} = 83.3\%$	$\widetilde{m}_{CHM} = 83.3\%$
(*Wilcoxon*):	$\widetilde{m}_{CSP} = 50\%$	$\widetilde{m}_{CHM} = 83.3\%$	$\widetilde{m}_{SSP} = 83.3\%$	$\widetilde{m}_{SSSP} = 66.6\%$	$\widetilde{m}_{SSSP} = 66.6\%$

Table A.1: Pairwise comparison of the study results of the assign task as well as the compare task using a Wilcoxon rank sum test. Stimuli names with superior performance in pairwise comparisons are shown in bold and significant p-values are marked with an asterisk (*).

Figure A.10 displays the user *error* in density value estimations for the task type *assign*. The results show that users of CSP made significantly less errors than users of SP. This observation aligns with our hypothesis **H1** that a color coding of scatter plots can already improve the users' accuracy in density estimation tasks. Interestingly, we did not detect a significant difference in the *compare* task. While users of CSP performed better than users of SP, they still performed significantly worse than users of CHM in both the *assign* and *compare* tasks. This finding confirms previous results, for example from Sarikaya and Gleicher [141].

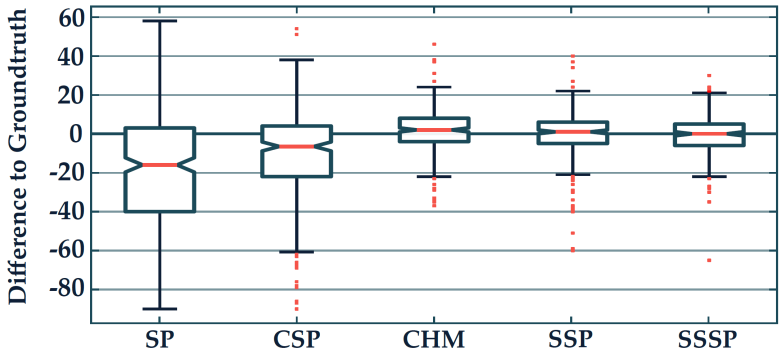

Figure A.10: Overview of the errors during the density value estimations for the assign task in all five visualization stimuli.

As shown in Table A.1, we could not find a significant difference between the user performance in CHM and SSP stimuli, neither for the *assign* task nor for the *compare* task. This aligns with our hypothesis **H2** that sunspot plots do not perform worse than continuous heatmaps for density estimation tasks. This provides us furthermore with an indication that our visualization method can indeed utilize the individual benefits of scatter plots and continuous heatmaps without losing the advantage in density estimations that heatmaps provide.

We could not confirm hypothesis **H3**, that the introduced illumination model and ambient occlusion would improve the users' accuracy in estimation density in sunspot plots. Users of the SSP and SSSP stimuli did not perform significantly differently, neither for the *assign* task nor for the *compare* task. However, as shown in Table A.1, we saw a significant difference between CHM and SSSP, showing that shaded sunspot plots can provide a higher accuracy than continuous heatmaps for the *assign* task.

In summary, we can observe that blending continuous and discrete representations did not have a negative effect on the typical tasks for density-based visualizations, while having the advantage of including additional information of the individual data samples. A comparison of pure heatmaps with the illuminated sunspot plots in the *assign* task showed a significant performance improvement, however this result should be taken with a grain of salt, since no clear improvement could be observed between illuminated and non-illuminated sunspot plots. Hence, for the basic tasks we considered in this study, we can conclude that additional illumination likely does not provide substantial benefits, but we could not find evidence that it was harmful, either. An interesting topic for further investigation is whether this holds true for other types of tasks, e.g., ones that relate to patterns or high-frequency variations in density.

A.7 Performance

The following performance measurements were conducted on a desktop computer equipped with an Intel Core i7-8700K CPU (3.7 Ghz), 16 GB RAM, an NVIDIA GeForce RTX 2080 graphics card with 8 GB of texture memory, and Windows 10 Home 64-bit as operating system. In total, we analyzed nine data sets. Six of them were created artificially with different sampling rates (from 1,000 to 1,000,000 points). The other three data sets correspond to the usage examples from Section A.5. During

these tests, we scaled the data sets so that their bounding box corresponded to the exact viewport size. Data set *World Cities* is the only exception. Here we zoomed in on Europe to generate a more realistic application scenario. In total, we analyzed two different viewport sizes (768x768 and 1280x1280). Table A.2 shows a detailed overview of the test scenarios including the number of sample points and the average number of frames per second (avg_{min}^{max}) per viewport.

Data Set	Points	Viewport 768x768	Viewport 1280x1280
Artificial Data	1,000,000	$11.51_{9.93}^{11.75}$	$4.33_{4.31}^{4.41}$
	200,000	$37.18_{36.90}^{37.32}$	$13.72_{13.31}^{13.95}$
	100,000	$55.70_{55.59}^{55.78}$	$21.45_{21.43}^{21.47}$
	50,000	$83.875_{83.48}^{84.22}$	$31.28_{29.28}^{31.77}$
	10,000	$116.35_{115.47}^{117.25}$	$38.06_{37.85}^{38.12}$
	1,000	$261.44_{257.06}^{266.76}$	$75.30_{74.69}^{75.87}$
World Cities	189,280	$34.01_{33.96}^{34.06}$	$13.60_{12.07}^{13.95}$
Boston Marathon	31,984	$68.31_{65.52}^{69.06}$	$26.14_{23.13}^{26.75}$
t-SNE	10,000	$128.01_{127.58}^{129.35}$	$49.12_{48.92}^{50.03}$

Table A.2: Overview of the analyzed test scenarios including their names, number of sample points, and the average framerate in frames per second (avg_{min}^{max}) using two viewport sizes.

A

A.8 Discussion and Limitations

We have developed sunspot plots as a flexible visualization approach for large scattered bivariate data. Although relatively straight-forward, we are unaware of previous work that smoothly combines discrete and continuous representations of scattered data in this manner. Furthermore, we implemented automatic adjustment of the kernel size to the current zoom level; we also support a non-uniform kernel computation based on the density distribution, and allow for the use of a magic lens that shows more local details by reducing the kernel bandwidth. Each of these three tools can be enabled and disabled on demand. At the moment, our implementation does not support the encoding of additional properties using color or size. Further studies are required to find out how such point encodings affect the human perception when being blended. Additionally, sunspot plots can only be used to visualize single-class scatter plots. In the future, multiple classes could be supported by blending multiple surfaces, each representing a specific class. Similarly, it must be investigated how sunspot plots behave in combination with other visualizations. For example when used as overlay, especially because sunspot plots are already the result of a blending operation.

Throughout the paper, we used the blending function described in Equation A.5. We found that this blending function does not only provide a sensible transition between dense and sparse regions, but also results in most readable visualizations. We experimented with other blending functions, such as curvature or normal-orientation based blending. However, we found that these functions were less predictable, and that the intuitive interpretation of distance-based blending is advantageous.

When using frequency-based methods, the visual quality of the result is highly dependent on kernel properties such as shape and size. We therefore allow users to dynamically adjust these parameters to change the degree of aggregation of the data samples. This can lead to different visualization results within the same data set. However, this is a drawback of all frequency-based methods. In our current implementation, we therefore provide a preset of kernel properties that we derived through empirical trials.

In our experiments, we found that sunspot plots are scalable in terms of data size and most effective for data with high sample density including global and local density variations. Data sets containing only sparse samples are well represented with conventional scatter plots, whereas dense data containing only global variations can be depicted effectively with frequency-based approaches. The same applies to uniformly sampled data sets since sunspot plots will then not encode additional information.

Although illumination of sunspot plots was not shown to significantly improve the assessment of the density information, there might be other tasks benefiting of an illumination model which were not evaluated in our study. In our personal and anecdotal experience, illumination did improve the perception of local density variations, where high-frequency changes in local density tend to manifest themselves as rough vs. smooth surface appearance. However, such differences were not captured in our experiment, as they do not correspond to the most typical tasks in scatter plots, and yet need to be confirmed through further empirical studies covering a wider set of tasks.

Finally, our current GPU-based implementation is capable of rendering several thousands of data samples in real-time. If one were to visualize even larger data sets, for example several millions of sample points, the frame rate would no longer be interactive. To be able to process such large amounts of data, additional pre-processing steps such as subsampling would be required.

A.9 Conclusion

We have presented sunspot plots, a visualization approach for dense scattered bivariate data. Sunspot plots allow a smooth transition between discrete and continuous data representations based on the local sample density in the image space. As such, sunspot plots alleviate the common drawbacks of discrete and continuous frequency-based visualization approaches for scattered data. The smooth blending between the discrete and continuous representations enables the visualization of main trends in dense areas while still preserving outliers in sparse regions. We have demonstrated that our technique is able to handle various data sets while still allowing for interactive exploration. Finally, we have evaluated the effectiveness of our approach in a user study, indicating that blending discrete and continuous information does not impair the estimation of densities while providing additional information of individual data points.

Acknowledgment

The research presented in this paper was supported by the MetaVis project (#250133) funded by the Research Council of Norway. In addition, we want to thank Michaël Aupetit for providing us with a variety of scatter plot data sets for our user study.

A

Paper B

Honeycomb Plots: Visual Enhancements for Hexagonal Maps

Thomas Trautner[1], Maximilian Sbardellati[2], Sergej Stoppel[1,3], and Stefan Bruckner[1]

[1]University of Bergen, Norway
[2]TU Wien, Austria
[3]Wolftech Broadcast Solutions AS

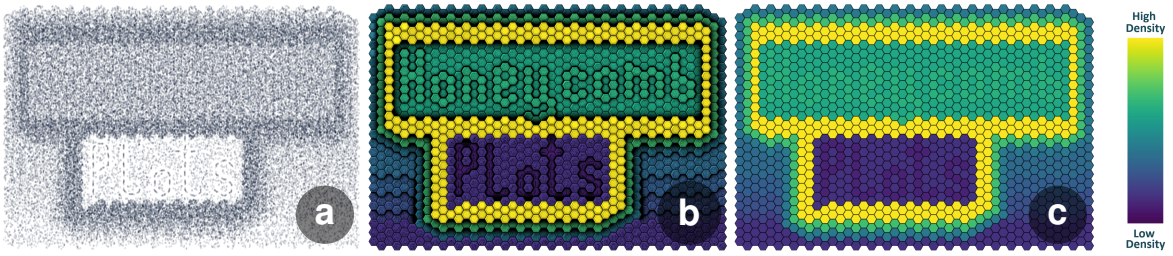

Figure B.1: Honeycomb plots (b) combine (a) scatter plots and (c) heat maps using per-tile densities. Through shading, ambient occlusion, and an implicit point-distribution encoding, the observer can explore features that cannot be captured by either of the techniques alone.

Abstract

Aggregation through binning is a commonly used technique for visualizing large, dense, and overplotted two-dimensional data sets. However, aggregation can hide nuanced data-distribution features and complicates the display of multiple data-dependent variables, since color mapping is the primary means of encoding. In this paper, we present novel techniques for enhancing hexplots with spatialization cues while avoiding common disadvantages of three-dimensional visualizations. In particular, we focus on techniques relying on preattentive features that exploit shading and shape cues to emphasize relative value differences. Furthermore, we introduce a novel visual encoding that conveys information about the data distributions or trends within individual

Accepted for publication at the *Symposium on Vision, Modeling, and Visualization (VMV 2022).*

tiles. Based on multiple usage examples from different domains and real-world scenarios, we generate expressive visualizations that increase the information content of classic hexplots and validate their effectiveness in a user study.

B.1 Introduction

Hexplots represent a form of spatial aggregation usually applied to a large number of two-dimensional points, making them resistant to overplotting. They rely on a subdivision of the plane by a regular space-filling grid of hexagonal polygons. However, Cleveland and McGill [39] emphasize that the visual system may fail to detect quantitative information from geometric aspects of a visualization. For example, aggregation comes at the expense of the perceptibility of individual points, making outliers invisible [49], and potentially obscures trends or clusters. Additionally, being able to distinguish between tile colors with at least just-noticeable difference (JND) can be essential, especially when color describes quantitative data properties [150, 151]. We, therefore, assessed potential disadvantages of current techniques by applying *algebraic visualization design* (AVD) [96]. Our strategy was inspired by McNutt [116], who highlights the advantages of this human operable and interpretable systematic framework. AVD analyzes how data changes affect the resulting visualization. Input for honeycomb plots are tabular 2D point coordinates. If the same data is displayed differently, this is referred to as *hallucinator*. When changes in the data remain invisible, this is referred to as *confuser*. Although both flaws cannot be avoided completely, they should be minimized. Therefore, we initially noted the following *confusers*:

C1: Point-data visualizations often assume equal density ranges. The underlying points are either sparse (scatter plots) or dense (density estimations and aggregations), making techniques reach their limits when data exhibit both simultaneously.

C2: Color-coded tiles containing a similar number of points may be perceived as equally dense, which would falsely correspond to the same visualization of actually different data.

C3: Heat maps only encode the quantity of aggregated points, so that uniform distributions within tiles cannot be distinguished from clusters or trends if their numbers of points match.

C4: Heat maps prevent the analysis of sparse features or outliers, allowing their position or arrangement to change although the aggregation, i.e., tile color, remains the same.

These confusers highlight the need for (C1) a novel hybrid visualization technique that preserves features of highly uneven distributions. Our goal was (C2) a visual encoding that overcomes problems with color coding of the first moment of statistics, i.e., the mean value describing average density as color, but also to (C3) capture statistical moments of higher order describing shape parameters, while (C4) strengthening the embedding of underlying points.

Our contribution can be summarized as follows:

- An interpretation of hexagonal tiles as *relief mosaic* where ambient occlusion serves as a subtle aid to perceive nuanced color differences between neighboring discrete tiles.
- An extension of the visually encoded information content by incorporating the regression plane of the underlying densities per bin in the form of a per-tile *diamond cut*.
- A hybrid approach that blends point data with colored hexagonal tiles corresponding to an *amber inclusions* metaphor, enabling an exploration of trends and clusters also in sparse regions.
- A quantitative user study consisting of four typical hexplot tasks in which we compare our encodings to classic heat maps.

In addition to the listed contributions, we developed an efficient hexagonal aggregation algorithm for two-dimensional data points, based on arbitrary grid sizes, in real time on the GPU.

B.2 Related Work

Hexagonal binning as aggregation technique was first introduced by Carr et al. [29] in 1987. They suggest the use of glyphs in the shape of hexagons whose size encodes the total number of data points within. Another technique called sunflower plots [38] uses flower-like glyphs in which petals represent aggregated points. A hybrid approach that visualizes individual data points as well as density estimations through binning are variable resolution bivariate plots, so-called varebi plots [80]. Over the years, a variety of related techniques spread under similar names: *hexagon plots* or *hexplots*, *hexbins*, *hexagonal binning plots*, *hexagonal tiles*, *hexbin maps*, *hexagonal gridded maps*, and *hexagonal heatmaps*.

For statistical summaries in cartography, Carr et al. [30] emphasize the advantages of hexagon mosaic maps, i.e., hexagon grid-cell choropleth maps, extendable by information layers [131]. Battersby et al. [12] investigate projection distortions of such grid structures resulting in either differently sized geographic areas as the basis for binning, or bin grid overlays of varying size. Example scenarios for spatio-temporal cartographic data contain the analysis of criminal activity [138] or visitor flows in amusement parks when solving the VAST 2015 Mini-Challenge (MC1) [26]. Another recent approach [177] focuses on multivariate game metrics using hexbin maps, Wurman dots [128], and arrow heads as direction indicators. To explore data attributes at specific (geospatial) locations, so-called attribute blocks [120], i.e., dynamically configurable regular arrays of "screen door" lenses, can be used. Possible alternatives are hexagonal cells as windows or magic lenses enabling the comparative visualization of multiple data sets [111]. Other techniques [69] evaluate blending and weaving to encode multivariate information by color. Established guidelines derived from exploring the visual design space of multi-class point data [76] and different representations using similar approaches [88] evaluate blending, weaving, majority-based coloring, embedding of

pie charts or bar charts as glyphs, etc. Furthermore, user studies analyzing user performance when comparing multiple heat maps [98] show that juxtaposed 2D heat maps work best for overview tasks, whereas stacked 3D heat maps (explored using stereo vision) are superior when reading and comparing single values. Although the two-dimensionality of honeycomb plots prevents occlusions and perceptual distortions, our spatialization cues could also be mapped to 3D geometry or digitally fabricated.

B.3 Honeycomb Plots

Hexplots, in contrast to choropleth maps [51], enable a fairer spatial comparison of aggregated values since differently sized landmasses do not affect their interpretation. Although well suited for this, hexplots also limit the parameter space for additional visual encodings since *position*, *area*, *color*, *orientation*, and *shape*, as discussed by Munzner [122], are already occupied. We therefore investigated potential spatialization cues, i.e., three-dimensional shape cues, applicable to 2D visualizations, preventing disadvantages of 3D visualizations such as viewpoint choice and view-dependent occlusion. Spatialization cues have a long tradition, e.g., cushion treemaps [172] and shaded Voronoi diagrams [157], enridged contour maps [170], sunspot plots [163], and line weaver [164]. In all cases, luminance from shading expands the design space.

Perception studies analyze *lightness constancy*, i.e., if color and shadow can be separated by the human visual system. Szafir et al. [154] analyze molecular visualizations and confirm that users can assign shadowed colors to corresponding unshadowed colors. Langer and Bülthoff [100] highlight that viewers assume light from top left and that a (convex) object is likely observed from above. Irani et al. [83, 84] also showed that users prefer shaded visualizations which help them understand structures.

B.3.1 Relief Mosaic

Hexplots are well suited to spatially aggregate data points. However, in cases where minimal color differences are not merely attributable to falling within a margin of error, discrepancies between adjacent tiles are difficult to see. Inspired by locally adjusting heat maps [187], we apply *ambient occlusion* (AO), i.e., encoding accessibility of a surface point [190], to spatial density which is represented by the tile height. We exploit the advantages of a regular hexagonal grid and calculate AO analytically which is, in such a well-defined scenario, most precise and interactive in real time. As known from literature [4, 53], the AO integral of a surface point x occluded by a rectangle Q, defined by four corner points $q_{1,2,3,4}$, is calculated as follows:

$$AO(x,Q) = \frac{1}{2\pi}\sum_{i=1}^{3}\left(cos^{-1}\frac{v_i \cdot v_{i+1}}{\| v_i \| \cdot \| v_{i+1} \|}\right)\left(\frac{v_i \times v_{i+1}}{\| v_i \times v_{i+1} \|}\cdot N\right) \tag{B.1}$$

where $v_i = q_i - x$ corresponds to vectors pointing from x to the corner points of Q, and N is the surface normal of x. According to Quilez [133], the double integral of the occluding surface can be reduced to a line integral of its perimeter, whose individual sides projected on the unit hemisphere correspond to the angle of the arc itself, which can be computed as inverse cosine of two consecutive corner points. The contribution of the

occlusion is then calculated as dot product between the surface normal at the point of occlusion and the normal of the triangle connecting the two corner points to the surface point. Unlike shadow mapping [183], where a light source casts shadows, AO emphasizes the accessibility of a tile to its immediate neighbors. A point x can, therefore, only be occluded by a maximum of six rectangles corresponding to the adjacent side walls of potentially higher tiles. The total AO can, hence, be calculated as the sum of six individual occlusions:

$$AO(x, \mathbf{Q} = \{Q_{1,\cdots,6}\}) = \sum_{j=1}^{6} AO(x, Q_j) \tag{B.2}$$

The summarized AO factor can then be used as per-pixel darkening factor when coloring tiles. Figure B.2 shows how AO can help understanding the spatial interpretation of scalars as a height field.

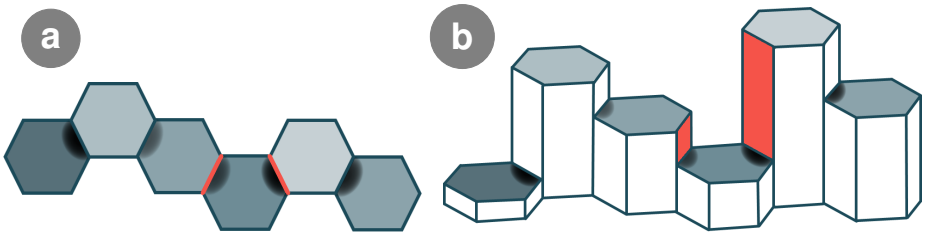

Figure B.2: Illustrations of how AO emphasizes structures using a viewpoint from (a) top-down and (b) "the side", including two red rectangular side walls that cause such an exemplary tile darkening.

Figure B.6(a-c) shows a synthetic example of a density increase from left to right, observable in (a) the scatter plot and (b) the heat map. The vertical trend in the center, however, can only be emphasized by (c) AO. Such an augmentation can be beneficial in two ways: first, to highlight color differences in continuous heat maps, and second, to visually emphasize more nuances than the number of colors that is available in a discrete heat map. For example, cartographers rarely use more than seven colors on choropleth maps [72]. Using a relief mosaic also shows that explaining the semantics of a heat map, i.e., which colors correspond to the maximum/minimum density, is now preattentive as it can be derived from the topological structure.

Since darkening, however, might not always distort colors favorably, we conducted a user study (see Section B.7) to evaluate its impact. We chose *viridis* as the heat map for all our visualizations, as it corresponds to the default in many standard tools and scientific environments, such as the python package matplotlib or R for statistical computing and graphics. It is perceptually uniform even when printed in black-and-white and was developed to improve readability for users with color deficiency or color blindness. Since none of our techniques focuses on absolute numbers, the color legend is only visible in the teaser Figure B.1.

B.3.2 Diamond Cut

Classic hexplots rely on flat tiles, leading to the intuitive assumptions that densities within tiles are either uniform or neglectable. However, non-uniformly distributed

points may form trends that are relevant for analysis. Inspired by hexagonally shaped glyphs [29] and Phoenixmaps [188] as abstraction of data points, we introduce a diamond cut metaphor. Similar to how a rough diamond is cut into a brilliant, we cut hexagonal pyramids so that their cut surface corresponds to the *regression plane* of the contained points, i.e., a fit plane that best approximates the underlying density distribution. This requires the computation of six intersections, one of which called p' can be calculated as follows:

$$p' = l_0 + l\frac{(p_0 - l_0)\cdot N}{l\cdot N} \quad \text{with} \quad l = \frac{l_0 - p_i}{\| l_0 - p_i \|} \tag{B.3}$$

wherein l_0 is the pyramid top, p_0 is the center point of a tile at the height of the average density, N is the normal, i.e., the regression plane orientation, and l is the normalized direction pointing from one of the six hexagonal grid corner points p_i to the pyramid top l_0. An illustration of this calculation is shown in Figure B.3.

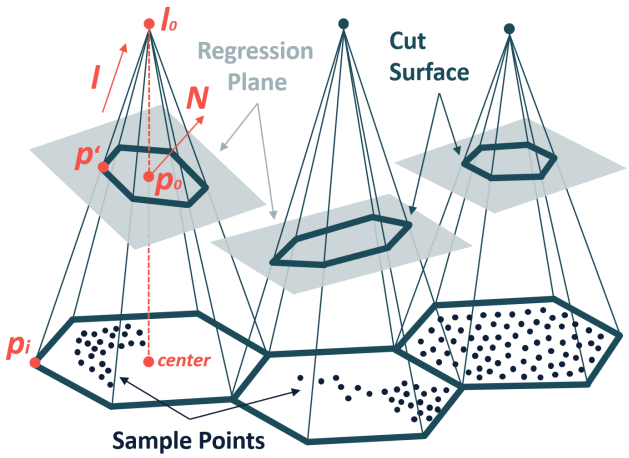

Figure B.3: Pyramid intersections with planes (tilted to the left/right and horizontally) that best approximate the underlying point density.

These slopes can be interpreted as diamonds pointing towards the steepest descent. The smaller and narrower a glyph gets and the more it is attracted to one side of the hexagon, i.e., the more it is repelled from another side, the steeper and more extreme the trend. An illustration of this is shown in Figure B.4. Alternatively, inverting the pyramids, i.e., using concave cavities rather than convex spikes, without changing the orientation of the regression planes would instead produce glyphs reminiscent of *radar charts* where attraction in one direction points towards the steepest ascent.

Figure B.6(d-f) shows a data set containing three regions with different densities visible using (e) a heat map, but not in (d) the scatter plot. Using (f) a diamond cut, the regression plane within the tile can be inferred from the shape, shading, and outline of the truncated pyramid. To evaluate whether users understand this concept within 15-20 minutes, we conducted a user study (see Section B.7).

B.3.3 Amber Inclusions

Regression planes are insightful but, unfortunately, cannot capture outliers. Similar to how Novotny and Hauser [125] preserved outliers in parallel coordinate plots (PCP),

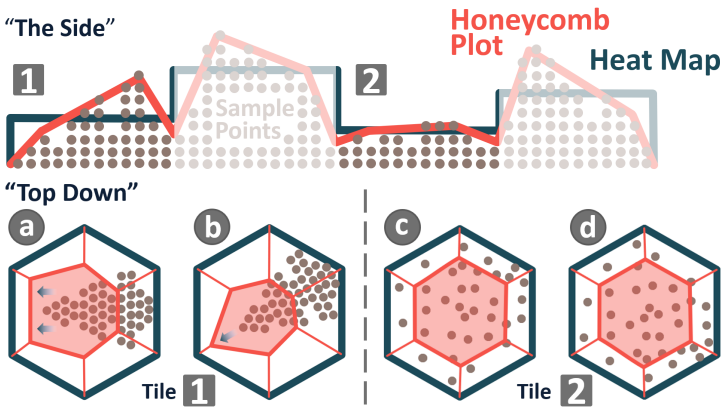

Figure B.4: Comparison of four neighboring tiles aggregating points viewed from "the side". The heat map (blue line) corresponds to the average density, whereas the diamond cut (red line) provides per-tile density distributions. When viewed "top down" as usual, tile (1) shows two possible decreases from right to left (a,b) and tile (2) exemplifies two feasible, almost uniform, distributions (c,d).

points within sparse regions could be analyzed as in scatter plots. Therefore, we follow approaches like sunspot plots [163] or Splatterplots [114] and combine hexplots with the underlying points based on the relative densities of tiles. We employ a per-tile opacity modulation mapping the transparency of a tile (encoding a relief mosaic and a diamond cut) to the interval $[0, \textit{density of the densest tile}]$. This results in a metaphor where flat and transparent amber shows enclosed insects rather than thick cloudy amber. We use the Porter-Duff [132] *over* operator which defines **A over B** as:

$$C = \frac{\alpha_A A + (1-\alpha_A)\,\alpha_B B}{\alpha_A + (1-\alpha_A)\,\alpha_B} \tag{B.4}$$

wherein C is the final pixel color, A the hexplot, B the scatter plot, α_A the normalized density/opacity, and α_B the scatter plot opacity. Figure B.5 shows tiles with different densities using (a) amber inclusions, (b) their combination with diamond cut spatialization cues, and (c) additional 2D contour polygons emphasizing the cut surface. Figure B.6(g-i) shows a synthetic data set with particularly distinctive sparse features in the form of letters from A to P, which can be clearly seen in (g) the scatter plot. The (h) heat map captures the density increase in the center but obscures the distinctive letters. Using (i) amber inclusions, however, both the sparse features and the dense data characteristics are preserved. We, again, evaluated whether blending complicates the interpretation of the other encodings as part of our user study (see Section B.7).

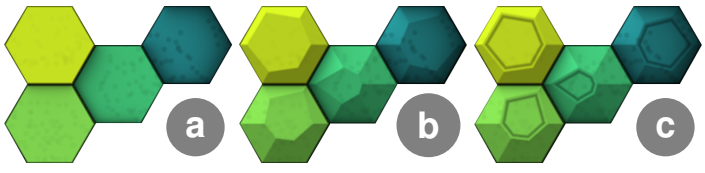

Figure B.5: (a) Blended tiles with (b) diamond cuts and (c) contours.

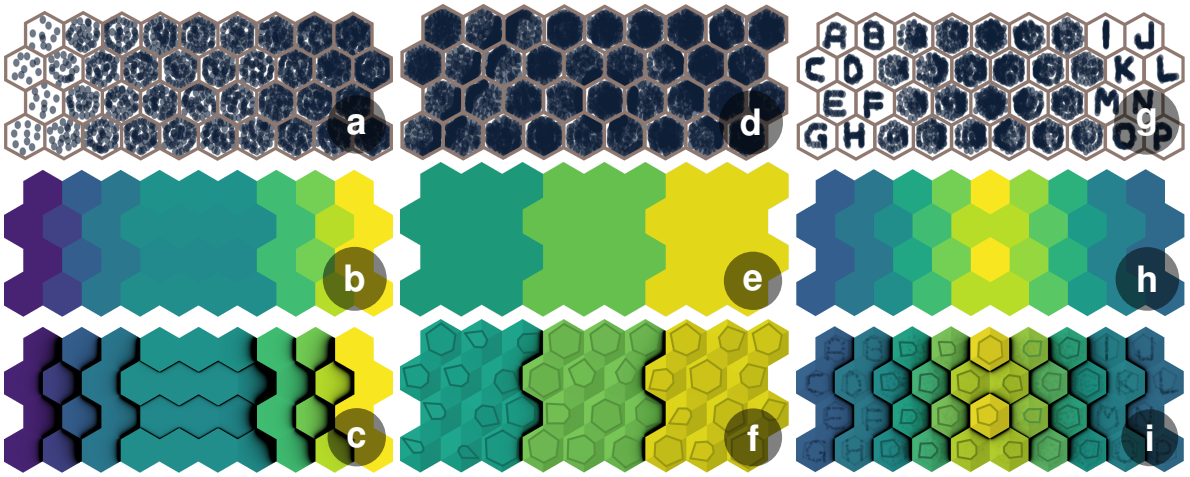

Figure B.6: Three scenarios comparing scatter plots and heat maps to our techniques: scenario one (a-c) shows the applicability of the relief mosaic, scenario two (d-f) highlights advantages of the diamond cut, and scenario three (g-i) illustrates amber inclusions.

B.4 Usage Examples

In this section, we demonstrate benefits of honeycomb plots using real-world usage examples. In each one, we show how classic hexplots can be enhanced to increase their information content.

B.4.1 US Tornadoes (1950 - 2019)

The NOAA's National Weather Service - Storm Prediction Center has been collecting tornado data [124] since 1950. Figure B.7 shows a honeycomb plot aggregating points corresponding to the starting position of a tornado between 1950 and 2019. In the center of the US, there is the so-called "Tornado Alley". The visualization, furthermore, reveals that most of Florida's tornadoes form along the west coast between Tampa Bay and Forth Myers. Nevertheless, there is a large number of tornadoes visible as continuous black dotted line along the east coast. This phenomenon describes so-called "waterspouts", which are extraordinary tornadoes forming over water. Additional indicators for this are the diamond cuts along the coast pointing towards the open sea, illustrating the highly decreasing number of tornadoes from this direction.

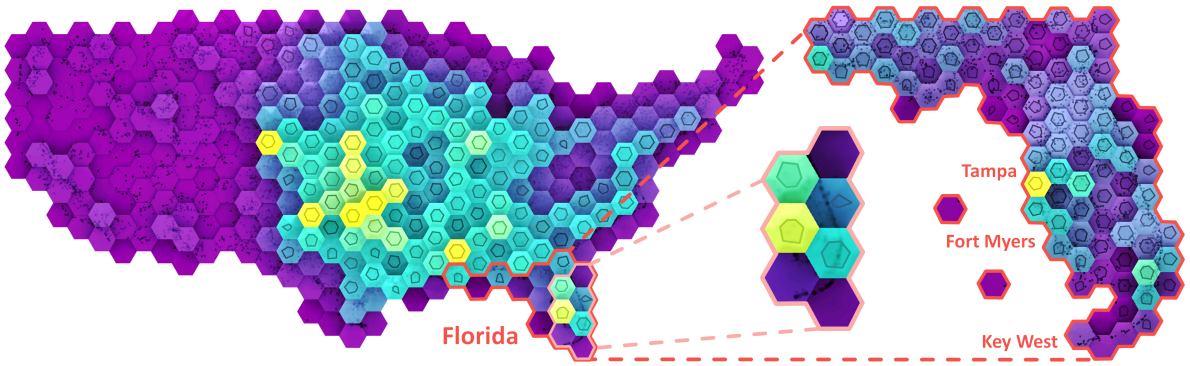

Figure B.7: Honeycomb plot of US tornadoes between 1950 and 2019.

B.4.2 California Housing Data (1990)

A well-known machine-learning data set is the California Housing data [161] containing the longitude and latitude of houses, their age, the number of rooms, etc., collected during the 1990 US census. Figure B.8 shows a hexagonal aggregation of their locations, where one point corresponds to the geographical position of a house. As highlighted by the heat map, the four largest metropolitan areas are located along the west coast: Los Angeles (4 M), San Diego (1.4 M), San Jose (1 M), and San Francisco (0.9 M). A heat map without blended points, however, fails to encode how houses are arranged, such as in San Francisco along the San Francisco Bay, and may lack emphasis of regions where density only varies slightly corresponding to minor changes in colors, such as in Redding (0.09 M), still being the largest city in California north of Sacramento. Using honeycomb plots, however, both data properties relevant for analysis and exploration are subtly highlighted.

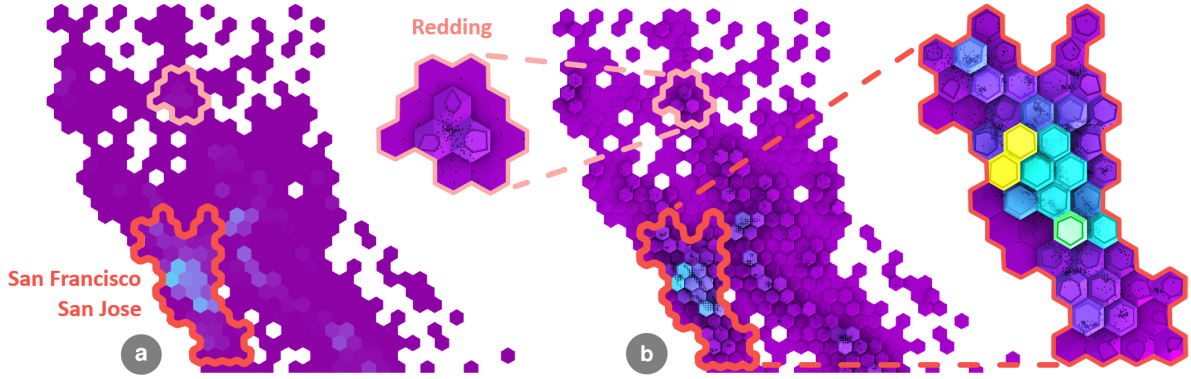

Figure B.8: Juxtaposition of (a) a heat map and (b) a honeycomb plot of the aggregated locations of houses originating from the 1990 US census. Both visualizations show agglomerated metropolitan areas along the west coast. Heat maps (a) alone, however, do not show underlying data points and the detection of minimal color differences depends on the user or hardware. Using (b) honeycomb plots, both are visualized as shown in the two enlarged very dense as well as sparse regions using an updated zoom-dependent heat map.

B.4.3 Gender Equality Index EU-28 (2020)

This example illustrates a relief mosaic of data without underlying points. In many domains, e.g., cartography [12, 30, 46, 131], it is common practice to rely on discretization. Figure B.9 shows a simplified map of Europe with equally sized hexagonal countries, which supports a comparison regardless of landmasses. The color of each tile corresponds to the gender equality index, except purple countries which only provide geographical context. Additionally, each hexagon contains a picture of the prime minister (as of August 2021). The resulting visualization highlights a north-south disparity, as well as a west-east divide within Europe. Due to the coloring, countries like Sweden (SE), Denmark (DK), the Netherlands (NL), and France (FR) stand out. Unfortunately, when using the heat map alone, no difference can be seen between Estonia (EE), Latvia (LV), Lithuania (LT), Poland (PL), Czech Republic (CZ), Slovakia (SK), Hungary (HU), Romania (RO) and Croatia (HR). They appear to form a light-blue plateau of equal gender indices although they vary between 60.7 (EE) and 53.0 (HU),

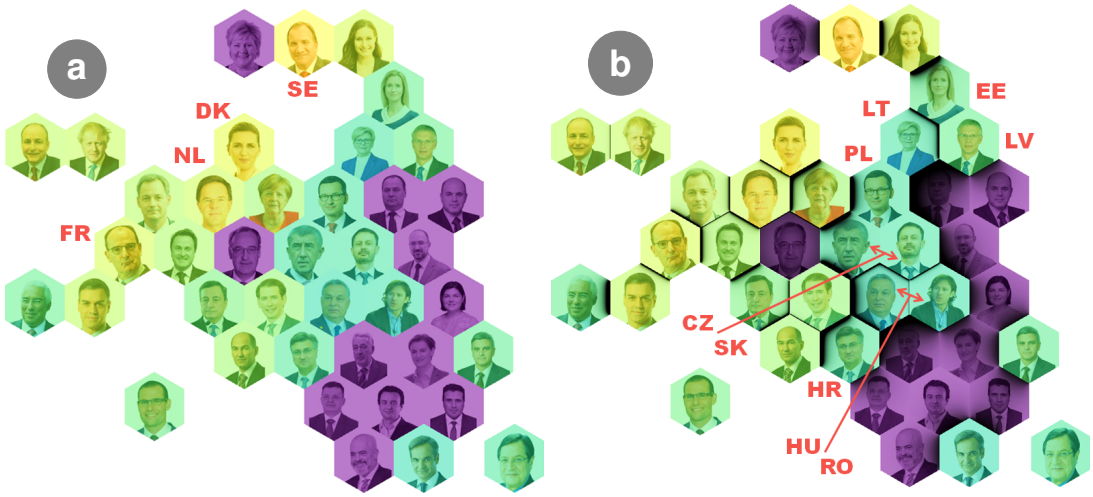

Figure B.9: Juxtaposition of (a) a heat map and (b) a relief mosaic visualizing a pseudo-spatially arranged hexbin map of Europe.

approximately 10 percentage points. Using a relief mosaic, three groups can be differentiated: Estonia and Latvia; Lithuania, Poland, Czech Republic, and Slovakia; and Hungary and Romania.

We focused primarily on hexagonal grids, as these are among the most common types. A relief mosaic is, nevertheless, not limited to hexagons. Other shapes would equally be possible, e.g., deltoids, trapezoids, parallelograms, rectangles, triangles, etc.

B

B.5 Implementation

Our approach was implemented in C++ and OpenGL, and all calculations were performed in parallel on the GPU. Using ImGui [42], all user-dependent parameters can be adjusted in real time. The five implementation steps necessary for honeycomb plots are listed below.

Scatter Plot: First, we create a scatter plot by rendering all points as semitransparent disks with additive blending. To prevent aliasing, the opacity of points decreases from their centers and their color as well as their radii can be modified using a GUI.

Kernel Density Estimation: Second, the density of the data set is calculated using a kernel density estimation (KDE). As previously, we render points additively as disks, but now evaluate a two-dimensional Gaussian bell function centered at each sample point. This time, however, the point disks are enlarged to cover the Gaussian bell curves until their contributions are considered to be zero. Again, the GUI can be used to change the sigma of the Gaussian function depending on the granularity of the analysis.

Hexagonal Aggregation: Third, we aggregate all points by rendering them into a texture whose dimensions correspond to the number of rows and columns of the hexagon grid. After determining the tile in which a data point falls, a counter in the associated texture texel is increased using additive blending. The grid layout is adjustable using a GUI slider allowing for various grid resolutions.

Regression Plane: Fourth, using the central difference of per-pixel densities within each tile, originating from the KDE, we reconstruct per-pixel normals and sum them up in per-tile shader storage buffer objects (SSBO). Their normalized sum then corresponds to the regression-plane normal. Similarly, we average per-pixel densities as they define the height of a tile pyramid.

Compositing: Fifth, depending on the total number of points per tile, a heat map color is assigned. Next, optionally, the AO is calculated according to Equations B.1 and B.2 using the density, i.e., height, of neighboring tiles. If enabled, the diamond cut is calculated according to Equation B.3 using the regression plane normal and the average density. Finally, the initial scatter plot is blended with the honeycomb plot using the *over* operator from Equation B.4.

The source code of honeycomb plots is available at GitHub:
`https://github.com/TTrautner/HoneycombPlots.git`

B.6 Performance

The performance measurements listed in Table B.1 were conducted on a desktop computer with an Intel Core i7-8700K CPU (3.7 GHz), 16 GB RAM, an NVIDIA GeForce RTX 2080 graphics card with 8 GB of texture memory, and a Windows 10 Home 64-bit operating system. Our analysis was based on the seven data sets used as paper figures ranging between 1,831 and 68,306 sample points aggregated into 36 to 1,242 hexagonal tiles. They correspond to four synthetically generated data sets and three real-world data sets. Table B.1 additionally shows which of our techniques were enabled: relief mosaic (RM), diamond cut (DC), and amber inclusions (AI), indicated by check marks. Our focus was on the analysis of achieved frames per second (FPS) and has shown that all data sets can be rendered as well as interacted with in real time.

Figure	Points/ Tiles	RM	DC	AI	FPS $_{1280\times720}$	FPS $_{1920\times1080}$
B.1	68,306/ 1,242	✓	✓	✓	$80.11^{80.29}_{77.66}$	$26.29^{26.40}_{25.05}$
B.6c	2,024/ 38	✓			$629.24^{643.37}_{603.80}$	$433.79^{435.68}_{429.22}$
B.6f	3,244/ 36	✓	✓		$414.99^{417.94}_{412.62}$	$220.72^{222.26}_{218.78}$
B.6i	2,244/ 38	✓	✓	✓	$406.59^{407.78}_{401.27}$	$219.68^{220.35}_{216.23}$
B.7	66,388/ 329	✓	✓	✓	$36.79^{37.25}_{33.48}$	$18.73^{19.14}_{15.35}$
B.8	20,640/ 637	✓	✓	✓	$493.32^{497.22}_{483.75}$	$250.72^{251.28}_{247.38}$
B.9b	1,831/ 44	✓			$641.34^{649.28}_{625.18}$	$298.61^{303.63}_{296.84}$

Table B.1: Performance analysis of all seven visualized data sets.

B.7 User Study

A main concern about honeycomb plots is their interpretability and the required learning effort. We therefore conducted a user study and compared standard heat maps (HM) to our three contributions, i.e., relief mosaic (RM), diamond cut (DC), and amber inclusions (AI), all exemplified in Figure B.10. The website of the online survey as well as the collected results are included in the supplemental material. We investigated the following hypotheses:

H1: Relief Mosaic (RM) - ambient occlusion improves performance, i.e., correctness, in value estimation tasks.

H2: Diamond Cut (DC) - regression planes improve performance, i.e., correctness, in slope estimation tasks.

H3: Amber Inclusions (AI) - blending of points with tiles does not negatively affect other encodings, i.e., RM and DC.

B.7.1 Experiment Design and Questions

Our study, evaluating whether honeycomb plots can be understood within a short time, took 15 to 20 minutes. We chose a between-subject design and randomly assigned visualization techniques. Inspired by Padilla et al. [126] and Kraus et al. [98], we chose Brehmer and Munzner's [22] visualization-tasks typology as basis for four questions:

Q1: *"Which tile has the highest/lowest density value within the highlighted box?"*

Q2: *"Which tile has the steepest/flattest change in density within the highlighted box?"*

Q3: *"Which tile has the highest/lowest density value, adjacent to the tile with the red asterisk (∗), within the highlighted box?"*

Q4: *"Which tile has the steepest/flattest change in density, adjacent to the tile with the red asterisk (∗), within the highlighted box?"*

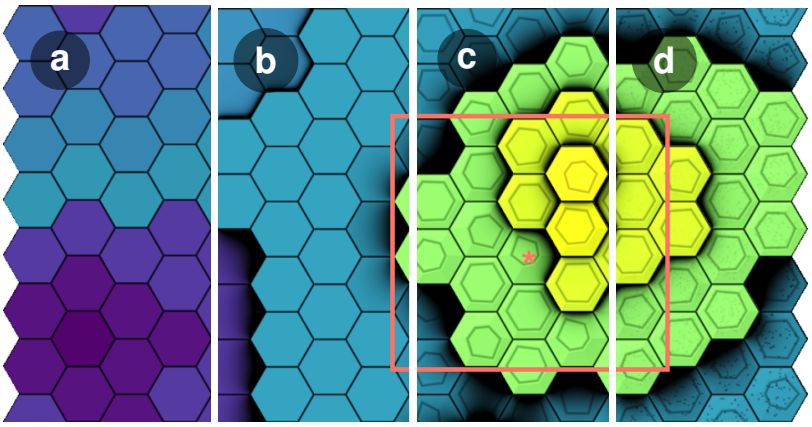

Figure B.10: Comparison of four stimuli of a sample question used in our study. From left to right, the techniques expand on the previous stimuli: (a) HM, (b) RM, (c) DC, and (d) AI.

We avoided asking questions about overall patterns in the data, as our techniques primarily improve the perception of individual tiles. Questions Q1 and Q2 focus on a global level, i.e., a larger highlighted region whose size varies. Questions Q3 and Q4 correspond to a local level, i.e., within the immediate vicinity of the tile with the red asterisk. Furthermore, two questions relate to density value estimations (Q1, Q3) and two to trends or density distributions (Q2, Q4). In order to answer the questions, participants may proceed differently, which then in turn corresponds to different tasks according to Brehmer and Munzner [22]. If, for question Q1, a participant first interprets the heat map and then searches for the corresponding value within the visualization, this represents a *locate* (target known, location unknown) and identify task. If the visualization is first examined and only afterwards compared to the heat map, this corresponds to an *explore* (target unknown, location unknown) and identify task. Since question Q2 requires a diamond glyph interpretation, it corresponds to an *explore* and identify task. Both Q3 and Q4 specify the location using a red asterisk. Again, Q3 may correspond to a *lookup* (target known, location known) or *browse* (target unknown, location known) and compare task, and Q4 again to a browse and compare task. Each question had to be answered using three different synthetic data sets, created by us, with varying difficulty, i.e., $4 \times 3 = 12$ questions. Finally, we evaluated binarily whether the correct tile was found or not.

B.7.2 Participants and Procedure

The online user study included 42 participants (11 HM, 10 RM, 9 DC, and 12 AI). Participants (32 male, 10 female) were aged from 16 to 69 ($\widetilde{m}_{age} = 32$ years) and stated whether they were wearing glasses (19 yes, 23 no) or suffered from color vision deficiencies (1 with, 40 without, and 1 that did not know). They rated their familiarity with scatter plots ($\widetilde{m}_{sp} = 4$), density estimations ($\widetilde{m}_{de} = 2$), and heat maps ($\widetilde{m}_{hm} = 4$) on a scale from $1 =$ *not familiar* to $5 =$ *very familiar*. At the beginning, the assigned visualization type was introduced, followed by a technical explanation of the survey. Subsequently, participants had to work through the 12 questions and finally answer general questions about themselves, were given the opportunity to leave comments, and asked to rate from $1 =$ *strongly disagree* to $5 =$ *strongly agree* how aesthetically pleasing they found their assigned technique ($\bar{m}_{sp} = 3.23$, $\bar{m}_{hm} = 4.09$, $\bar{m}_{rm} = 4.7$, $\bar{m}_{dc} = 4.55$, $\bar{m}_{ai} = 4.5$).

B.7.3 Study Results

At the beginning, we evaluated whether the collected results were normally distributed, which was rejected by a Kolmogorov-Smirnov test at a 5% significance level. We, therefore, conducted pairwise comparisons of all techniques per question, shown in Table B.2, using Kruskal-Wallis tests. Furthermore, Table B.3 shows the mean error made by participants, including histograms (with bins from left to right for 0, 1, 2, or 3 mistakes) showing the number of errors made. Although we also measured times, these varied widely, which is why we did not analyze them further. For **H1**, we observed that the error rates for value estimation were generally low for RM, DC, and AI. On a global level (Q1), RM (6.6%) and AI (13.8%) were significantly lower in comparison to HM (39.3%). We also see that for question Q3, where a much smaller neighborhood

Stimuli:	HM vs. **RM**	HM vs. **DC**	HM vs. **AI**	RM vs. **DC**	RM vs. **AI**	DC vs. AI
Participants:	21 (11, 10)	20 (11, 9)	23 (11, 12)	19 (10, 9)	22 (10, 12)	21 (9,12)
Q1:	$\chi^2 = 8.10$ $\mathbf{p < .05}^*$	$\chi^2 = 0.31$ $p = .57$	$\chi^2 = 5.09$ $\mathbf{p < .05}^*$	$\chi^2 = 3.25$ $p = .07$	$\chi^2 = 0.20$ $p = .65$	$\chi^2 = 1.93$ $p = .16$
Q2:	$\chi^2 = 0.42$ $p = .51$	$\chi^2 = 15.55$ $\mathbf{p < .05}^*$	$\chi^2 = 19.90$ $\mathbf{p < .05}^*$	$\chi^2 = 14.48$ $\mathbf{p < .05}^*$	$\chi^2 = 19.14$ $\mathbf{p < .05}^*$	$\chi^2 = 1.33$ $p = .24$
Q3:	$\chi^2 = 3.03$ $p = .08$	$\chi^2 = 0.06$ $p = .80$	$\chi^2 = 0.11$ $p = .73$	$\chi^2 = 2.35$ $p = .12$	$\chi^2 = 1.74$ $p = .18$	$\chi^2 = 0.01$ $p = .91$
Q4:	$\chi^2 = 1.16$ $p = .28$	$\chi^2 = 17.27$ $\mathbf{p < .05}^*$	$\chi^2 = 19.84$ $\mathbf{p < .05}^*$	$\chi^2 = 15.78$ $\mathbf{p < .05}^*$	$\chi^2 = 18.40$ $\mathbf{p < .05}^*$	$\chi^2 = 0.75$ $p = .38$

Table B.2: Pairwise comparisons using a Kruskal-Wallis test with superior stimuli in bold and marked (*) significant p-values.

	Q1:		**Q2:**		**Q3:**		**Q4:**	
	mean error	hist.	mean error	hist.	mean error	hist.	mean error	hist.
HM:	$\bar{m} = 39.3\%$		$\bar{m} = 72.7\%$		$\bar{m} = 9.0\%$		$\bar{m} = 93.9\%$	
RM:	$\bar{m} = 6.6\%$		$\bar{m} = 66.6\%$		$\bar{m} = 0.0\%$		$\bar{m} = 86.6\%$	
DC:	$\bar{m} = 33.3\%$		$\bar{m} = 3.7\%$		$\bar{m} = 7.4\%$		$\bar{m} = 0.0\%$	
AI:	$\bar{m} = 13.8\%$		$\bar{m} = 0.0\%$		$\bar{m} = 13.8\%$		$\bar{m} = 2.7\%$	

Table B.3: Overview of the mean errors and corresponding histograms with bins of 0, 1, 2, or 3 mistakes (from left to right).

had to be considered (comparing a tile to its six neighbors), there are no significant differences between the techniques, providing support for the assumption that color encoding alone is sufficient. In case of DC (33.3%), the results were not significant. We suspect that this could be rooted in the fact that the introduction of a visual encoding that did not have any benefit for the task at hand may have caused confusion in some participants. For **H2**, we found strong evidence that our diamond cut metaphor is effective in facilitating slope encoding tasks, as both DC and AI performed significantly better than HM and RM on a global (Q2) as well as a local (Q4) level. In both cases, the participants were able to solve the corresponding tasks with very low error rates (DC: 3.7% for Q2 and 0.0% for Q4, AI: 0.0% for Q2 and 2.7% for Q4). With respect to **H3**, there were no significant differences between DC and AI in any of the questions (Q1-Q4), which supports our hypothesis that additional point encoding does not introduce negative side effects.

These findings demonstrate that while AO can provide substantial benefits for value estimation tasks, care must be taken when mixing it with other encodings that are not relevant to the task at hand. The results also suggest that participants were generally able to understand our diamond cuts based on the explanation in the study. We believe that this initial study provides support for the beneficial effects of incorporating spatialization cues into binning plots, while also demonstrating the clear need for particular care when combining multiple encodings based on the task at hand.

B.8 Discussion and Limitations

The focus of a relief mosaic is emphasizing color differences, making it unsuitable if these are unimportant or within a margin of error. Using a diamond cut for shape pa-

rameters also requires caution, especially if a grid layout is not implicitly defined since features may change if the grid is scaled, shifted, or rotated. Additionally, while we found it intuitive to adjust AO with a slider, future work could focus on automatic selections depending on, e.g., data variety. It would also be interesting to investigate to what extent AO alone is able to encode quantitative data. Future studies could examine if the sign of quantitative data, comparable to the neutral point of a classic heat map, can be encoded using AO or the pyramid geometry, i.e., tiles could have a convex or concave shape. We, furthermore, have not tested the effectiveness of amber inclusions but rather analyzed whether they negatively impact other encodings. Future research could quantitatively evaluate the analysis of outliers.

All our plots use the classic *over* operator, wherein the aggregated density influences the blending. Although this is well suited to highlighting sparse features, we have also experimented with other influence factors, e.g., point discrepancy. This would render tiles opaque when contained points are nearly evenly distributed, whereas tiles containing points close to each other or large empty areas are rendered transparently. In this work, we augmented hexplots with spatialization cues leading to $2\frac{1}{2}$D visualizations. We are, nevertheless, confident that honeycomb plots would also be well suited for 3D physicalization such as digital fabrication. It would be interesting to explore how tasks that are usually performed on hexagonal aggregation plots could be carried out by visually impaired or blind people using 3D prints of honeycomb plots.

B.9 Conclusion

In this work, we have presented novel visualization techniques for hexagonal binning plots. We proposed a relief mosaic that exploits ambient occlusion highlighting differences between similarly colored tiles, introduced a regression plane as diamond cut that reveals point distributions, and emphasized trends or clusters in sparse regions using amber inclusions. We showed that our techniques can be implemented in real time on the GPU. Based on three usage examples, we generated expressive visualizations that increase the information content of hexplots, and showed, supported by a user study, that honeycomb plots can be understood in short time.

B

Paper C

Line Weaver: Importance-Driven Order Enhanced Rendering of Dense Line Charts

Thomas Trautner and Stefan Bruckner

University of Bergen, Norway

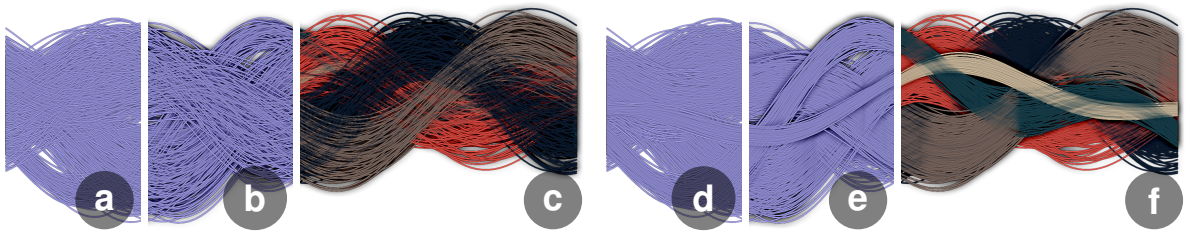

Figure C.1: Line weaver was inspired by techniques from textile production where multiple threads are interwoven to form fabrics. If the blending order is (a) ignored or naively used, essential visual information is lost, even if (b) outlines and halos are added or (c) clusters are colored. If, however, the ordering of clusters is (d) optimized, both (e) outlines and halos as well as (f) colors can help perceiving clusters.

C

Abstract

Line charts are an effective and widely used technique for visualizing series of ordered two-dimensional data points. The relationship between consecutive points is indicated by connecting line segments, revealing potential trends or clusters in the underlying data. However, when dealing with an increasing number of lines, the render order substantially influences the resulting visualization. Rendering transparent lines can help but unfortunately the blending order is currently either ignored or naively used, for example, assuming it is implicitly given by the order in which the data was saved in a file. Due to the non-commutativity of classic alpha blending, this results in contradicting visualizations of the same underlying data set, so-called "hallucinators". In this paper, we therefore present line weaver, a novel visualization technique for dense line charts. Using an importance function, we developed an approach that correctly considers the

Computer Graphics Forum (EuroVis 2021), 40(3), pages 399–410, 2021. doi: 10.1111/cgf.14316

blending order independently of the render order and without any prior sorting of the data. We allow for importance functions which are either explicitly given or implicitly derived from the geometric properties of the data if no external data is available. The importance can then be applied globally to entire lines, or locally per pixel which simultaneously supports various types of user interaction. Finally, we discuss the potential of our contribution based on different synthetic and real-world data sets where classic or naive approaches would fail.

C.1 Introduction

Line charts, also known as *line graphs*, *line plots*, or *curve charts*, are among the most frequently used forms of visual representation in statistics [146]. In contrast to scatter plots, where data elements are shown as a set of two-dimensional points in space, line charts visually encode the relations between data elements. Therefore, individual points have to be ordered, for example chronologically, along an axis. Unfortunately, line charts have two major weaknesses: First, the individual lines become harder to distinguish and interpret with an increasing number of lines. Second, when using standard alpha blending, occlusion between lines means that the resulting image is dependent on the rendering order. A popular remedy to distinguish individual lines is the use of color. Unfortunately, this prevents the color channel from being used to encode any other data attributes. Considering the second problem regarding rendering order, it seems that this has not received much attention in the literature so far. In Tableau, for example, one of the most widespread visualization tools, *"the drawing order is driven by the order of the members in the data source* [155]*"*.

However, rendering order and the resulting occlusion relationships among graphical elements can have a significant impact on their perception, as indicated by the *Gestalt principles*. The principle of *Symmetry and Order*, for instance, sometimes also referred to as *Law of Prägnanz* or the *Law of Simplicity* describes that the human brain tends to interpret visual elements in the simplest possible way to avoid an overflow of visual stimuli. For example, when looking at a wireframe representation of a cube, where only the edges connecting the corner points are visible, our brain will tend to recognize the three-dimensional cube instead of individual primitives such as triangles, rectangles, or trapezoids that arise from intersections of lines.

In this paper, we present a novel technique for the visualization of line charts that avoids the issues caused by a global rendering order by introducing the notion of a quantitative importance function that can vary locally. Using a per-pixel blending approach controlled by this importance function, we provide explicit control over occlusion relationships, allowing us to make more efficient use of available screen space and present features such as clusters in a more coherent manner. The main contributions of our work can be summarized as follows:

- We introduce a new approach for displaying line-based data that supports the use of a quantitative importance function.
- We demonstrate how this method can be used with different types of importance functions.

- We present a simple algorithm for deriving an importance function for grouped line data.
- We show that our technique can be efficiently implemented on modern GPU architectures for high-quality rendering of line data.

C.2 Related Work

Visualization of line sets plays an important role in various domains and scientific fields, for example, in their elementary form as classic line charts, when exploring networks or graphs to better understand their structure, when visualizing streamlines or pathlines to analyze fluid flow, when interpreting temporal changes of time series, when displaying multi-dimensional data as parallel coordinates, or directly when researching how lines can be rendered as efficiently as possible. All research fields study different challenges but one they have in common is that visual clutter increases when more lines are displayed. For reasons of clarity, this section is divided into two categories with different clutter-reduction approaches. Section C.2.1 focuses on features derived from line data sets and how they can be visually encoded, and Section C.2.2 presents advanced and optimized rendering techniques for dense line and curve data.

C.2.1 Feature Encoding

One possibility for reducing visual clutter could be to visualize a density estimate of underlying lines, instead of visualizing the entirety of individual lines. Lampe and Hauser [44] introduce an approach based on kernel density estimation (KDE), using line kernels defined by a start and an end point. In a subsequent step, the estimated density can then be color-coded using a perceptually uniform heatmap. A similar approach can be used to estimate the density of curves in a continuous [43] or discrete manner [121]. Unfortunately, density representations in general are not well suited when analyzing individual lines, especially in sparse regions. Recent work by Trautner et al. [163] presents sunspot plots, an approach focusing on this challenge when visualizing scatter plots. In case of line charts, however, the problem remains unsolved.

Another visualization technique that suffers from visual clutter are parallel coordinate plots (PCPs). Assuming the multi-dimensional data originates from a continuous domain, continuous parallel coordinates [181] represent a related approach to KDEs and, therefore, benefit from the same advantages while also suffering from the same disadvantages. Instead of changing the visual representation of parallel coordinates in advance, an initial step can be reordering the axes. Blumenschein et al. [17] recommend that, especially with highly cluttered data sets, axes with dissimilar data dimensions should be arranged next to each other, whereas for data sets with low clutter, similar axes should be displayed next to each other. Fua et al. [62] introduce a hierarchical cluster-based enhancement for PCPs. They propose to visually encode individual clusters as variable-width opacity bands in combination with proximity-based coloring. The width of such a band represents the extent of the cluster. The center of each band is fully opaque while transparency linearly decreases towards the top and

bottom edges. Unfortunately, they do not provide further specification on the blending operator or blending order used. Novotny and Hauser [125] introduce a technique specifically targeted at outlier and trend detection within PCPs. The authors propose to detect outliers first, providing them with a separate visual representation, and then applying aggregation techniques to the underlying data to prevent outliers from being smoothed away. Related work by Artero et al. [6] uses image processing techniques to detect clusters. Work by Johansson et al. [89] introduces transfer functions to highlight different properties of clusters in PCPs. In the end, however, lines are rendered in a given order mostly using the non-commutative Porter-Duff [132] *over* operator for blending, neglecting that it is not order independent.

Under the assumption that lines can be viewed as three-dimensional trajectories or networks, Kwon et al. [99] introduce edge bundling using a spherical graph layout and depth routing for edges to improve legibility of graph visualizations. Subsequently, line bundles are emphasized by global illumination using real-time ambient occlusion approaches similar to the work by Eichelbaum et al. [54]. Here, it is important to mention that we do not consider edge bundling as a competing approach but as a possible pre-processing step. The resulting line bundles could subsequently be displayed using our technique.

Additional inspiration comes from the work of Nakayana and Yano [123] who combine classic space-time cubes with KDEs by using a spatio-temporal kernel for 3D point data to emphasize both the temporal duration as well as the spatial extent, visualized using volume rendering. Subsequently, Demšar and Virrantaus [47] built upon this idea and applied 3D density estimations to three-dimensional polylines. The interpretation of line sets as volumes [143] allows for the use of transfer functions, which inspired us to use importance functions without having to convert the underlying data to a volume. Moritz and Danyel [121], for example, suggest normalizing the contribution of a curve to the density estimation by its arc length, correcting for the higher numbers of pixels that are needed to render strongly fluctuating curves. Instead, we use the arc length as a derived geometric property, describing curvature and frequency. This enables us to optimize overall visibility and reduce visual clutter by displaying curves with more variability in the back and less fluctuating curves in front. However, many other properties such as visual complexity, as described by Ryan et al. [140], geometric features derived from families of curves by Konyha et al. [97], or statistical features could be used instead.

C.2.2 Line Rendering

Before the development of more advanced line rendering techniques, Spear [146] recommended creating multiple graphs using the same unit scale and comparing them, for example, using juxtaposition in case the data set was too cluttered. This is still a common practice when current techniques reach their limits. Building on that, Cleveland et al. [37] recommended that an average angle of 45 degrees should be kept between adjacent line segments. In the literature, this is often referred to as "banking to 45 degrees". Furthermore, line smoothing techniques can be applied, for example as proposed by Rosen and Quadri [137], to initially remove high frequencies or noise from the data, which we consider an optional pre-processing step to our approach.

Rendering smooth lines in OpenGL is a basic functionality that has yet to be supported by all graphics cards. Kilgard's [94] polar stroking technique is a recent approach which uses the arc length of a curve for texturing or dashing. A general overview of different CPU and GPU rendering techniques for transparent 3D line sets is provided by Kern et al. [93]. Similar to our method, an A-buffer [28] combined with multi-layer alpha blending can be used to approximate transmittance and color of fragments. While rendering, A-buffers store additional fragment information, for example, by using per-pixel linked lists together with a global atomic counter, as suggested by Yang et al. [185]. A modification of A-buffers are K-buffers, introduced by Bavoil et al. [13]. Instead of blending all fragments, only a fixed number of fragments are stored and blended. An enhancement of such a K-buffer is used by Groß and Gumhold [64] who introduce a GPU-based ray casting approach supporting ambient occlusion and transparency by using a billboard proxy geometry.

In their work, Hagh-Shenas et al. [69] compare the accuracy of two distinct strategies for visualizing multivariate data using different colors, namely weaving and blending. Weaving, as presented in the work by Luboschik et al. [107], refers to the selection of one colored item, such as a part of a line, which is then exclusively rendered on top at a given location. Blending, on the other hand, refers to the mixing of multiple colors, for example assigned to multiple lines which all overlap. Our approach can be seen as a hybrid between both, as we use blending to combine contributions of lines with similar importances, but since importances may vary along lines, their overall appearance will be reminiscent of a "weaving" pattern. Our method is related to approaches for smooth composition by Luft and Deussen [109] and Bruckner et al. [24]. In addition, we want to highlight the work of Everts et al. [59] on depth-dependent halos to emphasize line bundles. Their work inspired us to use halos, implemented using the unsharp masking approach proposed by Luft et al. [108].

Other related strategies consider the rendering of lines as global optimization problem. Günther et al. [65] suggest rendering only a globally optimized selection of lines that are indispensable when visualizing important features, thus preventing visual clutter that would arise from rending all lines. This approach has later on been refined ensuring coherence in 3D time-dependent flow visualizations [66] and to support not only lines but transparency optimization in combination with points and surfaces [67].

C.3 Line Weaver

Our approach addresses the fact that in many cases, not much attention is paid to the order in which lines are drawn in a chart. When rendering only solid single-colored and fully opaque lines, this does not make a significant difference, but as soon as transparency is introduced or more advanced stylization approaches such as halos are applied, it becomes important – depending on the blending operator – to also consider the effects of the rendering order.

The *algebraic model for visualization design* proposed by Kindlmann and Scheidegger [96] explicitly refers to such considerations. They illustrate this by using a plot of taxi pick-up and drop-off locations, which they consider as a set of points without an inherent order, and argue that using an order-dependent blending operator (such as

the common *over* operator) in this case constitutes a failure of representation invariance, which they refer to as "hallucinator", i.e., a deficiency where differences in the images may arise from representational or algorithmic choices (such as the rendering order) without reflecting changes in the underlying data. Order-independent operators will, by definition, always generate the same result irrespective of the rendering order, and in their example Kindlmann and Scheidegger propose to use additive blending as a possible resolution. In additive blending, the contributions of all elements to a pixel are averaged and hence their order is irrelevant. The drawback of this approach is that it completely eliminates occlusion which in fact can be a powerful cue. As already mentioned, we have learned from the Gestalt principles such as *continuity*, *closure*, and *figure-ground* that human perception attempts to complete missing or occluded image regions.

In our approach, we therefore propose to preserve these cues by still using a blending operator that exhibits occlusion, but instead providing explicit control over occlusion relationships by introducing an importance function. While this may at first glance look like a minor semantic distinction (replacing the term order by the term importance), it opens up several interesting and, as we will demonstrate in the remainder of this paper, advantageous avenues for improving the visualization of line-based data.

We regard line data as set $D = \{L_1, L_2, ..., L_N\}$ of N polylines with its members $L_i = (P_1, P_2, ..., P_M)$ represented as tuples of M ordered two-dimensional points $P_i = (x_i, y_i)$. The resulting parametric curve $l_i(u)$ of each member is normally a polyline generated by linear interpolation between its associated points $P_1, P_2, ..., P_M$, but of course other interpolation functions are equally possible. We choose this formulation since it is general enough to represent common visualizations such as time-series charts and parallel coordinate plots, which simply constitute different mappings between the underlying data set and the x and y coordinates of the individual points. Our importance function $\beta_i(u) \in [0,1]$ now associates a scalar importance value with every position along each curve $l_i(u)$ and has two major properties that distinguish it from order. First, our importance function is quantitative in nature, not just ordinal. This means that it is possible that two lines may have very similar or even the same importance value. Second, importance does not need to be constant along a line, but it may vary. These attributes should be represented in the resulting visualization. Specifically, our approach is based on the following requirements:

- The contributions of individual lines should be independent of their order in the data set or the order in which they are rendered.
- Line segments with similar importance contributions should contribute similarly to the pixels they cover in the final image.
- When the importance differs significantly, line segments with higher importance should occlude line segments with lower importance.

We can draw an analogy to the textile industry, for example, when comparing weaving or knitting techniques as illustrated in Figure C.2. Instead of a global ordering of individual threads, they are locally woven to interleave forming an intricate pattern that is discernible to human observers. In the same way, our aim is to thread lines in a

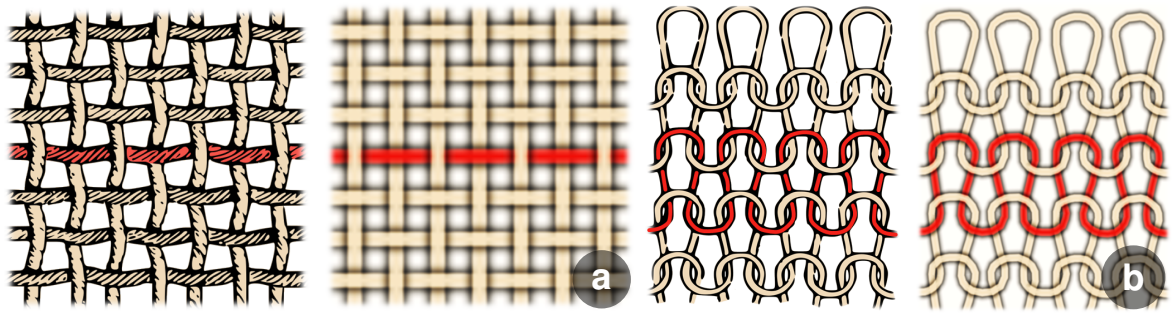

Figure C.2: Juxtaposition of illustrations of (a) plain weaving and (b) weft knitting in comparison to visualizations retrieved with our approach (right column). Note how both examples cannot be reproduced using classic alpha blending and a globally defined render order, i.e., depth per thread.

meaningful pattern, instead of simply pasting them on top of each other. Our approach, therefore, enables the importance of individual lines to vary along their trajectory, but without imposing an ordinal relationship among them.

C.3.1 Importance-Based Blending

In computer graphics, the term order-independent transparency commonly refers to techniques that aim to enable the rendering of transparent polygons using the Porter-Duff [132] *over* operator in a way that it does not require prior sorting of the primitives. This can occur in an exact manner producing the same results as primitive sorting, e.g., by still performing sorting but rather at the fragment level (using data structures such as an A-buffer [185]) or in an approximate manner, e.g., by using different (typically depth-based [14]) blending operators that attempt to mimic the results of the over operator as closely as possible. Our goal here, however, is different. We are seeking a blending operator that allows us to visually reflect the properties of our importance function, i.e., by conveying its quantitative nature as well as its variability over polyline primitives. The latter can be resolved by changing the granularity at which the blending is performed. Switching from a per-primitive to a per-pixel resolution, as it is done in A-buffer based approaches, allows us to freely vary the importance function across a primitive. The former, however, is more challenging as the decision of whether one element is in front of another in the classic *over* operator is inherently binary and therefore prone to artifacts.

A good illustration for this is z-fighting, caused by two primitives with differences in depth that are close to the numerical precision of the z-buffer, resulting in visible artifact patterns. As stated previously, we instead want our blending to have a well-defined and meaningful behavior when importance values are equal or similar. When blending together multiple elements with equal importance values, the result should correspond to the average value of all contributing elements, while when importance values are far apart, we want to indicate higher importance values using occlusions. As importance values can vary continuously, we further want to avoid abrupt changes between these behaviors. Such smooth transitions can be achieved using a blending approach inspired by the works of Luft and Deussen [109], and Bruckner et al. [24].

Conceptually, this works by having the color contribution of one element (e.g., a pixel color) influencing the contributions of other elements whenever they are within a certain importance range, using a continuous falloff function such that the contributions are equal when the respective importance values are the same. The individual contributions (which then include those of other elements within the influence range) are next blended in sequence of their importance using the conventional over operator. The adjusted color of an element c'_i is computed as follows:

$$c'_i = \frac{\sum_{\Delta(\beta_i,\beta_j)>0} c_j \Delta(\beta_i,\beta_j)}{\sum_{\Delta(\beta_i,\beta_j)>0} \Delta(\beta_i,\beta_j)}, \tag{C.1}$$

wherein c_j are the opacity-weighted color contributions of the other elements, β_i, β_j are the respective importance values, and Δ is a falloff function that gradually decreases to zero with the absolute difference of its arguments. In practice, we define Δ based on a Hermite polynomial, similar to the common *smoothstep* function, such that it is 1 if the two importance values are the same and becomes zero as their difference exceeds a threshold value:

$$\Delta(a,b) = \begin{cases} 0 & \text{if } |a-b| \geq t \\ 2\left(\frac{|a-b|}{t}\right)^3 - 3\left(\frac{|a-b|}{t}\right)^2 + 1 & \text{otherwise} \end{cases}, \tag{C.2}$$

wherein t is the user-defined threshold that specifies the range of contributions to be considered.

This approach now allows us to flexibly vary the importance function along individual lines. If the respective importances vary continuously, two polylines may pass through each other without abrupt changes. In regions where the importances are equal, each contributing element will contribute equally. The behavior of our blending approach with different importance functions is illustrated in Figure C.3. We show how the blending result of two completely overlapping lines varies based on their importance functions (a *step* function, a *sine* function, and a *tent* function).

C.3.2 Rendering and Stylization

For high-quality rendering of our polyline sets with variable thicknesses, we use a method that combines a geometric approach with signed distance functions (SDFs). This means that instead of generating geometry to define the exact outline, which can be difficult and expensive both due to the potentially high geometric complexity (e.g., when rounded joins between line segments are desired) as well as issues with numerical precision, we instead rasterize the line geometry conservatively and then evaluate an SDF for each fragment in order to determine the exact coverage.

In this method, we create a polygonal scaffolding for each individual polyline by extruding it on the fly into a triangle strip such that all potential pixels of the line are covered, similar to the proxy billboards used by Groß and Gumhold [64]. To avoid gaps in the pixel coverage, it is important to ensure that neighboring segments are extended such that the endpoints of their outline meet at the bisector between the two segments.

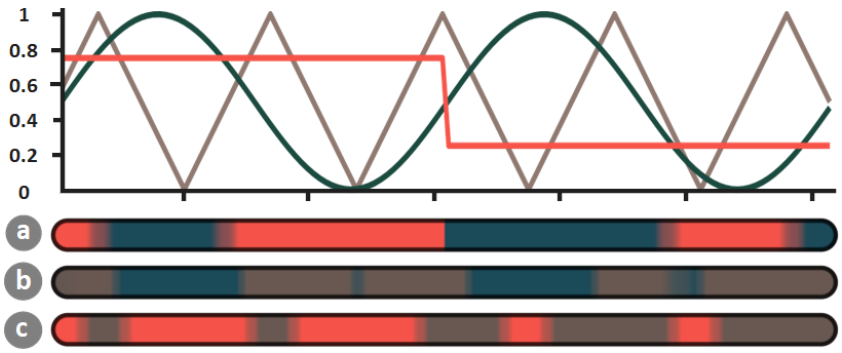

Figure C.3: Pairwise comparison of two lines, each with varying importance, blended on top of each other: (a) red and turquoise, (b) brown and turquoise, and (c) red and brown, wherein red has a step *function, turquoise a* sine *function, and brown a* tent *function as importance, using a smoothness of* $t = 0.15$*. Note that the visual result depends purely on the blending and not the render order.*

Then, for each fragment of the resulting geometry, we evaluate an SDF determining inside and outside of a line segment. The distance from a point P to a line segment connecting points A and B can be written as:

$$d(P,A,B) = \left|\vec{AP} - h \cdot \vec{AB}\right| - w, \text{ with} \tag{C.3}$$

$$h = \text{clamp}\left(\frac{\vec{AP} \cdot \vec{AB}}{\vec{AB} \cdot \vec{AB}}, 0, 1\right), \tag{C.4}$$

wherein w is the thickness of the line segment. Note that the boundary of the line is located at the zero level set of d and negative values correspond to positions inside the line segment. Figure C.4 shows an illustration of the resulting SDF. To ensure proper handling of joins, we need to take into account the distance to the previous, current, and next line segment. This can be achieved by using different operators combining the respective distances. Per default, we use the *minimum* operator which results in *rounded* joins, but other types such as *miter* and *bevel* are equally possible. Thus, our distance function d_i for the i-th segment of a polyline is defined as follows:

$$d_i(P) = \min\left(d(P,P_{i-1},P_i), d(P,P_i,P_{i+1}), d(P,P_{i+1},P_{i+2})\right). \tag{C.5}$$

At the same time, this approach enables easy stylization of our lines, as the distance can be flexibly mapped to different color and/or opacity profiles. Furthermore, per-fragment derivative functions available in common APIs such as OpenGL or DirectX can be used to perform anti-aliasing. For all the images in this paper, we use solid lines with a darkened outline, generated in the described manner. In addition, we use the unsharp masking approach by Luft et al. [108] to generate halos.

Once the color and opacity of a line fragment has been determined, we store it in a per-pixel linked list. To determine the final pixel color, the list is subsequently sorted based on importance and blended using the approach described in Section C.3.1.

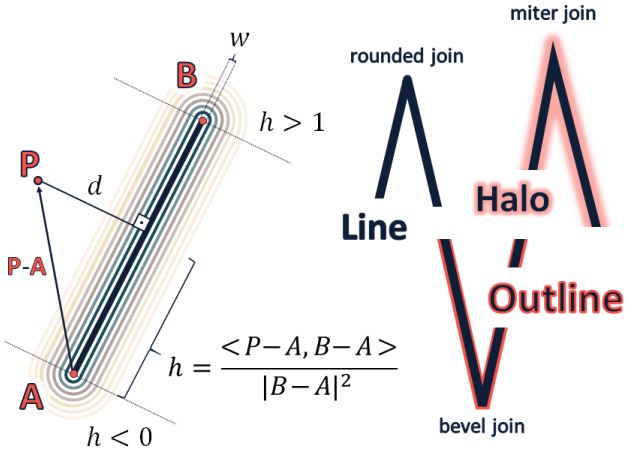

Figure C.4: Calculating the distance from all points P in space to a line defined by the two points A and B results in an SDF. Note how its contour lines converge to a circle as distance d increases. The SDF can then be utilized to render anti-aliased lines with variable thicknesses, as well as additional outlines, halos, or join types.

C.3.3 Importance Functions

In principle, there are many different ways of how to define an importance. First, and most naturally, the importance value itself may be part of the underlying data. For instance, in case of time-series data there may be a confidence measure associated with each data point. If the resulting visualization should then prioritize high confidence values, this would be an appropriate choice. Another example could be the numerical result of a feature detection algorithm. In many cases, however, such domain-specific, explicit importance measures may unfortunately not be available.

In this case, we may instead want to use more fundamental properties of our lines themselves. Our goal is to minimize the amount of overdraw in a heuristic manner, by assigning lower importance values to those lines that take up screen space. A simple way to achieve this is by specifying the importance based on the arc length of each line. The impact of this simple yet powerful geometric property is shown in Figure C.5. Extending this idea and exploiting the fact that importance values may vary locally, we propose an approach to generate importance values for the common scenario where lines grouped into different sets are depicted in a single chart. Examples include the depiction of time series grouped by a categorical variable, but our method equally applies to cases where the grouping has been established by other means, such as the application of a clustering method. In this case, each line has an associated group identifier and our goal is to reduce the amount of overdraw by assigning importances based on an estimate of the amount of screen space taken up by all the lines of a group along the dependent axis of the graph. For instance, if our chart depicts time series, the values for a particular group may vary considerably for one time step but may fall within a much narrower range as time progresses. Another group may exhibit an inverse behavior. In such a case, the importance values for each group should correspond to the estimated amount of screen space taken up by its lines at each time step, such that groups that take up less space receive higher importance values. Our blending method is well-suited for this purpose as it allows us to interpolate between the assigned importance values along the ordinate of our graph in order to avoid discontinuities.

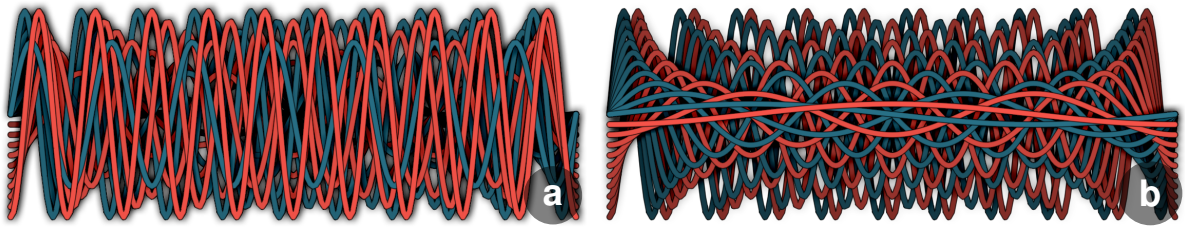

Figure C.5: Juxtaposition of a synthetic data set containing sine *as well as* cosine *functions with varying amplitude and frequency. The line charts are rendered using (a) the order in which the individual lines are specified within the data set, and (b) a global ordering dependent on the overall length of the lines implicitly encoding curvature and complexity. It can be seen that (a) high-frequency curves can easily obscure low-frequency curves with a low amplitude, whereas (b) an optimized order allows for all curves to remain visible.*

We propose a simple algorithm to compute the importance values for such a weaving pattern using a greedy approach (see Algorithm 1). For each value along the independent axis of the graph, we first compute the minimum and maximum values of each group, i.e., the interval on the dependent axis covered by lines belonging to this group. This allows us to determine the envelope of all lines associated with a group by connecting two subsequent intervals along the independent axis forming a trapezoid (for the last value on the axis, we simply duplicate the interval), and compute their areas. We then iterate over the calculated areas along the independent axis with the goal of establishing an ordering of all groups according to a cost function. The cost value for a group corresponds to the sum of the intersection areas with all other groups multiplied by the area covered by the group itself. Initially, all groups are marked as *active*. Using the computed envelope areas, we next determine the area of intersection between all other active groups. Then, we select the group with the lowest cost value among the active groups and mark it as *inactive*. We then repeat recomputing the cost values and select the group with the lowest cost until no active group remains. The importance values are then assigned based on this sequence of selection, i.e., the first selected group receives the highest importance, etc.

C.4 Implementation

Our approach was implemented in C++ and OpenGL. Loading as well as preprocessing the data, such as either deriving the importance based on arc length if no external importance is provided, or executing Algorithm 1 from Section C.3.3, are performed on the CPU while the rendering itself runs in parallel on the GPU. However, it would be equally possible to calculate or modify importance values on the GPU allowing for dynamic data. The selection of the data set to be visualized as well as additional user-dependent parameters, such as line color or width, were implemented using an *ImGui* [42] user interface. Our approach consists of the following phases:

Line rasterization: The input of our rendering pipeline are three buffers containing the x and y coordinates, and the importance per tuple. It is, however, unimportant in which order the lines are allocated within the buffers as they are sorted in the blending phase. In a geometry shader, based on the desired line thickness, we construct the

line scaffolding as triangle strips by using the GL_LINE_ADJACENCY primitive type, which provides the shader with access to neighboring vertices without requiring duplication. In the fragment shader, we evaluate the SDF from Equation C.5 in order to determine the fragments covered by the line segment and evaluate their colors. The result, together with its importance value, is then added to a per-pixel linked list. We use an image object to store the index of the last list entry for each pixel, as well as a shader storage buffer object (SSBO) to store the fragment data. This buffer contains a single counter for the total number of allocated entries, as well as an array of the list entries.

Fragment blending: We render a screen-filling quad to traverse the linked list for each pixel, performing blending as described in Section C.3.1. We use bubble sort to sort the list entries based on their importance and then use our blending operator to determine the final combined pixel color. If halos based on the method of Luft et al. [108] are enabled, this step is preceded by an additional blurring pass which is then used as secondary input to darken surrounding pixel regions.

Our complete source code is available at:
https://github.com/TTrautner/LineWeaver.git

input : A set of groups $G = \{G_1, G_2, ..., G_K\}$
output: Ordering for each value on the dependent axis
for *all values x on the independent axis* **do**
  compute group coverage areas $A(G_i)\ \forall G_i \in G$
  compute group intersection areas $I(G_i, G_j)\ \forall G_i, G_j \in G$
  mark all groups as active: $P \leftarrow G$
  initialize order to zero: $o \leftarrow 0$
  while $P \neq \emptyset$ **do**
    compute cost values:
    $C(G_i) \leftarrow A(G_i) \sum_{i \neq j} I(G_i, G_j)\ \forall G_i, G_j \in P$
    select group with minimal cost:
    $S \leftarrow \arg\min_{G_i \in P} C(G_i)$
    output and increment order, update active groups:
    $O_x(S) = o$
    $o \leftarrow o + 1$
    $P \leftarrow P \setminus S$
  end
end

Algorithm 1: Weaving Loom

C.5 Usage Examples

In this section, we now demonstrate the strength and versatility of line weaver based on three real-world data sets and one artificially generated data set. Each data set has a different number of lines (between 20 and 1,200) and a varying number of clusters (between 3 and 5). We further illustrate the properties of our approach by using diverse types of charts such as parallel coordinate plots (PCPs), Andrews plots [2], and time

series plots. Finally, we show that even user interaction can be efficiently implemented by presenting a usage example of a magic lens [152] for focus+context [74] exploration and angular brushing [75].

C.5.1 Global Importance

To the best of our knowledge, there are no studies that analyze the blending order and its impact on line charts. Blumenschein el at. [17] evaluated different axis reordering strategies in parallel coordinates and found that intersections of line bundles can help identifying clusters in cluttered data sets. Their benchmark data set therefore serves as a reasonable first test case for our technique. It allows us to compare their suggestions to our insights, e.g., that the z-ordering has a significant impact and that both orderings (axis and depth) are not independent. Figure C.6 shows data set *4C.6-150N-Sim* from their study. The authors concluded that this similarity-based axis arrangement tends to be less appropriate when identifying clusters. As shown, it is difficult to distinguish clusters, even when different colors are assigned to each of them. This also does not change when lines are additionally highlighted with halos. When using a global importance, i.e., per line, the exact shapes of the clusters become visible even without using colors. There are four clusters within a relatively noisy data set. The importance of each line is determined by the cluster it belongs to. Apart from noise, the cluster with the most lines is considered as most important. Within each cluster, the importance of the lines is then determined by their arc length, whereby the shortest line is rendered on top. Although the analysis of clusters using this similarity-based axes arrangement has so far been considered unsuitable, it can now be done expressively using line weaver.

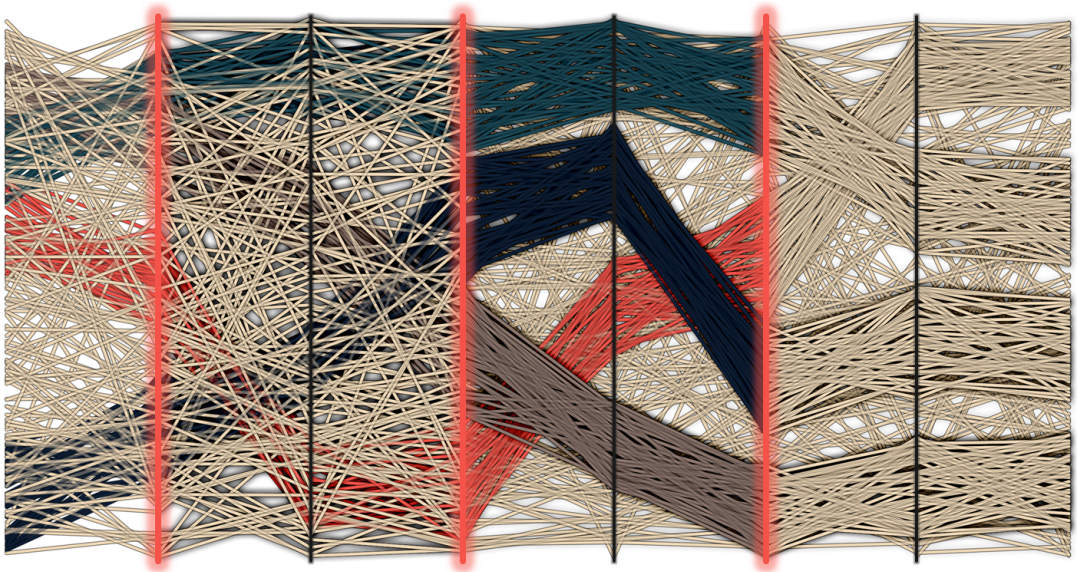

Figure C.6: Visualization of benchmark data set 4C.6-150N-Sim *[17] consisting of four clusters: brown (50 lines), dark blue (49 lines), turquoise (48 lines), red (46 lines), and beige noise (150 lines). From left to right, we compare the result of a randomized rendering order with outlines and additional highlights with halos, to our approach using an importance function depending on cluster size and arc length, including a monochrome version of it.*

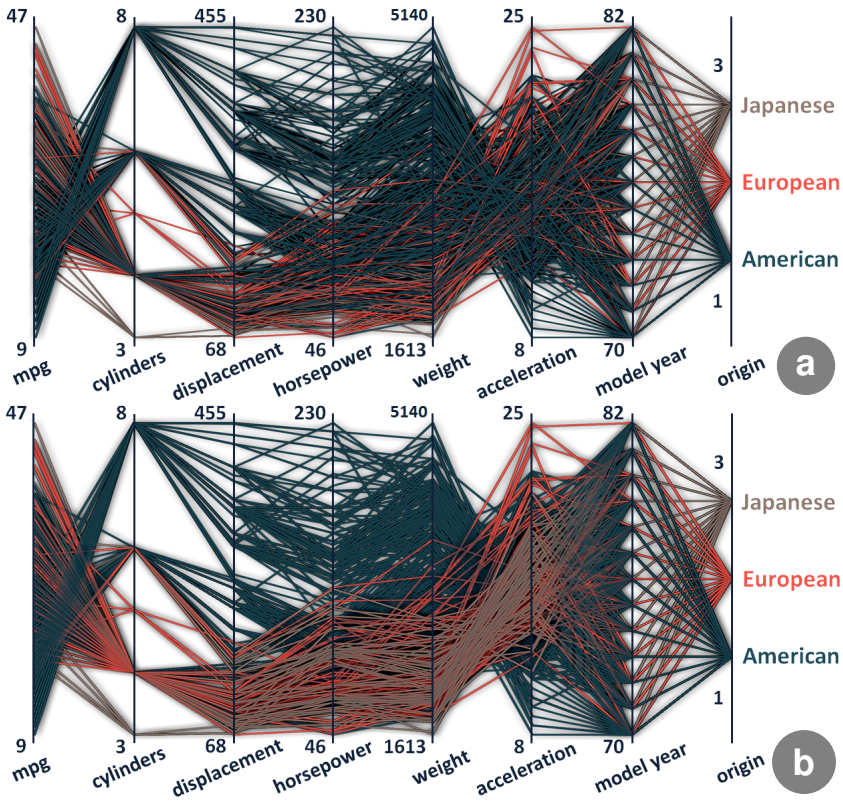

Figure C.7: Juxtaposition of parallel coordinates plots using (a) a randomized order and (b) line weaver of the Auto MPG *[50] data set consisting of three clusters: brown (79 lines) Japanese cars, red (68 lines) European cars, and turquoise (245 lines) American cars.*

C.5.2 Local Importance

To demonstrate our technique for locally varying importance functions, we first use a PCP of the MPG [50] data set from the UCI Machine Learning Repository containing properties of cars, such as origin, number of cylinders, weight, horsepower, etc. from cars produced between 1970-1982. We apply our weaving loom algorithm from Section C.3.3 using the origin variable to define three groups. As shown in Figure C.7, Japanese and European car models have similarly low displacement, horsepower, and weight, and therefore longer acceleration times, whereas American cars cover a much wider spectrum. In contrast to the rather compact and therefore stricter ordering, looking at model year, for example, all three clusters are equally broad and therefore averaged instead. As can be seen by comparing Figure C.7(a) and (b), the randomized order results in the almost complete occlusion of the more compact group of Japanese cars, while they remain clearly visible using our approach.

Apart from classic PCPs, there are various other visualization techniques for high-dimensional data. Another well-established form of visualization are Andrews plots [2] which are also referred to as *smooth PCPs* in the literature. Each n-dimensional data point $x = \{x_1, x_2, \ldots, x_n\}$ is defined as a finite Fourier series, visualized as curve $f_x(t)$ within the interval $[-\pi < t < \pi]$:

$$f_x(t) = \frac{x_1}{\sqrt{2}} + x_2 \sin(t) + x_3 \cos(t) + x_4 \sin(2t) + x_5 \cos(2t) + \ldots \qquad \text{(C.6)}$$

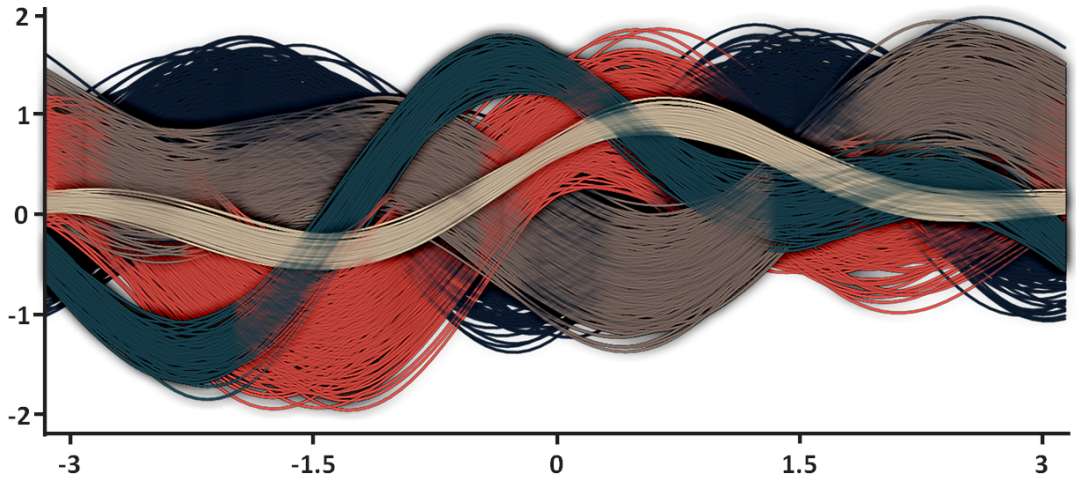

Figure C.8: Andrews plot of synthetic data showing two compact beige (100 curves) and turquoise (100 curves) clusters rendered over the wider dark blue (300 curves) and brown (250 curves) clusters which are in turn interwoven with the red (250 curves) cluster.

To display different facets of the plot and our technique, Figure C.8 shows such a transformation using a synthetically generated data set, which was also used as our paper teaser in Figure C.1. In contrast to the global importance, occlusion relationships between line bundles may change, resulting in a weaving pattern. The determining factor is the area of each bundle and how much it locally occludes other clusters. Due to this optimization, the most compact *beige* cluster becomes visible. Notice how its color and opacity are averaged when it overlaps the *turquoise* cluster with almost identical width. A similar phenomenon can be seen with the *dark blue* and *brown* clusters. In contrast, the less compact *red* cluster is alternatively woven from front to back and vice versa. Each of these changes is smooth which avoids hard cuts and discontinuities, enabling the viewer to better understand the entirety of the visualized data set.

As third example, we present time series data from Tan et al. [156] derived from high-resolution satellite images, each containing 1 million pixels, with one pixel corresponding to a geographic area of 64 m^2. In total, 46 images were taken over time and corrected so that each pixel corresponds to the same geographic region. Next, the temporal change of each area was analyzed considering 24 different category classes including corn, wheat, water, sunflower, etc. We selected four classes for our visualization: soy (class 10), grassland (class 12), poplar (class 21), and mineral surface (class 22), each containing 300 time series which in turn consist of 46 time steps each. The result comparing three different rendering orders is shown in Figure C.9: (a) random, (b) as stored in the file, and (c) line weaver using local importance. With random order, chance decides which lines are rendered last and are therefore best visible to the user, or how well individual outliers are preserved. Using an order defined by the data set, classes are perceived as individual bundles since they have been saved one after the other. Unfortunately, the class that was rendered first will be the least visible and the class rendered last will be visible best. Line weaver, on the contrary, works independently of the storage and rendering order, providing the same visual result even when these orderings are changed. Note that this time, the comparably high smoothness of the blending, i.e., averaging of clusters, guarantees that individual bundles are easy to identify (similarly to using a randomized order) while preserving outliers.

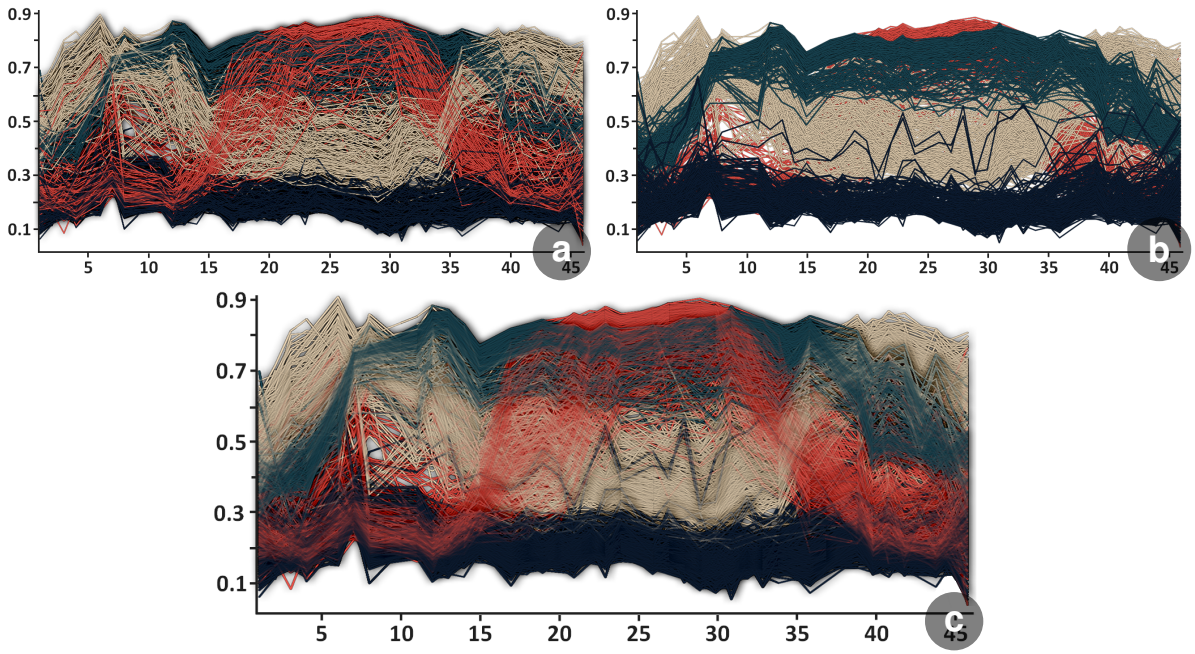

Figure C.9: Comparison of four classes of the Crop *data set from Tan et al. [156] rendered using (a) random line order, (b) the sequence individual lines were stored, and (c) line weaver using local importance. When using (a) random order, the outliers of the dark blue cluster, otherwise in the middle of the beige cluster, are lost. An ordering that is dependent on (b) the data set results in coherent cluster bundles rendered from back to front starting with class 10 (red), followed by class 12 (beige), class 21 (turquoise), and finally class 22 (blue). This time, the blue outliers remain visible by chance, but the red cluster bundle substantially disappears. Using (c) line weaver, both the red and turquoise clusters are interwoven with the beige cluster and the most compact dark blue cluster is rendered on top, preserving its outliers.*

C.5.3 Highlighting and Focus Enhancement

While the previous examples demonstrate how our technique behaves using different types of importance functions derived from the data, our approach also supports dynamically changing importance values. This enables various types of user interactions that demand variable local or global changes with arbitrary granularity. Interactions that would be encoded using an attribute such as color can instead be used to modulate importance values. To illustrate this, we show a simple implementation of two well-established techniques serving as representatives of common line chart interactions: a magic lens [152] that enables focus+context [74] exploration and angular brushing [75]. The lens, positioned at the mouse cursor, has local influence on the importance of all lines within and allows us to see lines that pass through a region even though they are covered by other lines, for instance to highlight lines with a specific axis value. Similarly, angular brushing can help to locally pull a bundle of lines with a certain angle forward – like a rubber band metaphor. As our approach supports quantitative importance functions, this can be done in a non-binary manner based, as is common, on smooth brushing operations.

For this example, we use the smooth subspace data set from Huang et al. [81] for k-means clustering of time series data. It consists of 3 clusters, each containing 50 time series, which in turn consist of 15 time steps per series. A distinct property of this data

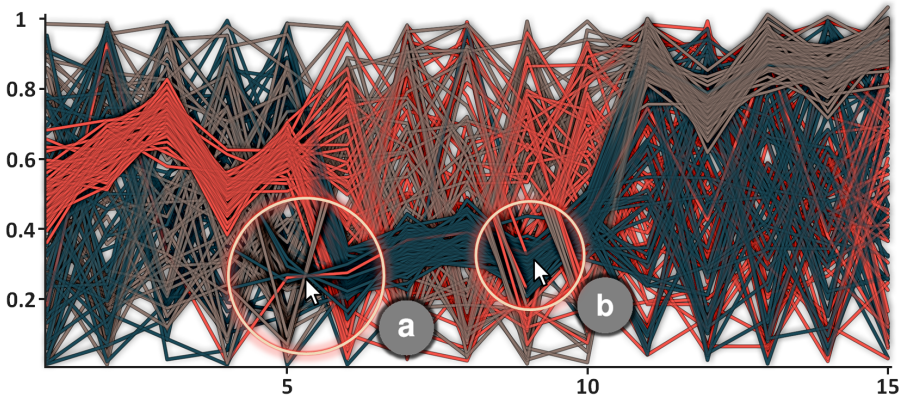

Figure C.10: Two examples of user interaction on a time series data set [81], using a magic lens for (a) focus+context and (b) angular brushing. The lens, centered at the mouse cursor, can be used to, e.g., trace outliers, highlight lines passing through a specific point, or locally change the ordering of bundles by emphasizing lines with certain angle, e.g. $-80°$.

set is that each cluster has a similar and compact pattern for 5 consecutive time steps, whereas all other steps are randomly distributed. For cluster 1 these are time steps 1-5, cluster 2: time steps 6-10, and cluster 3: time steps 11-15. The resulting visualization, where each compact pattern of a cluster is locally woven to the front once, is shown in Figure C.10. Note how it additionally provides examples of (a) a focus+context lens and (b) angular brushing.

C.6 Performance

We conducted performance measurements using a desktop computer equipped with an Intel Core i7-8700K CPU (3.7 GHz), 16 GB RAM, a NVIDIA GeForce RTX 2080 graphics card with 8 GB of texture memory, and a Windows 10 Home 64-bit operating system. Overall, we analyzed two artificially generated data sets and four real-world data sets, corresponding to the figures presented in this paper. In addition, we made sure that all data sets have different densities and varying numbers of lines (between 20 and 1,200). We used two representative screen resolutions (1280×720 and 1920×1080) and scaled each data sets so that its bounding box filled the viewport.

In addition to measuring the rendering performance, we also attempted to quantify the degree of overplotting using the following measure:

$$Overplotting = 1 - \frac{1}{|D|} \sum_{L_i \varepsilon D} \frac{\#visible\,Pixels\,(L_i)}{\#total\,Pixels\,(L_i)}, \tag{C.7}$$

where D is the data set the lines originate from and L_i represents a single line of which the ratio of visible to the total number of pixels that make up this line is calculated. These are then summed up and normalized. The calculated $Overplotting$ measure produces a value from the interval $[0,1]$, where 0 refers to no overdraw. For example, if two polylines are drawn exactly on top of one another, this corresponds to a value of 0.5, with three lines $0.6\dot{6}$, etc.

A detailed overview of all test cases is shown in Table C.1. It reveals that our technique provides interactivity for data sets containing thousands of lines. It is therefore easily possible to interact with the visualization, for example by changing individual parameters at runtime. The analysis furthermore shows that line weaver increases information content by reducing clutter and overplotting and therefore helps to display line data in a more expressive way.

Figure	Lines	FPS 1280×720	FPS 1920×1080	Overplotting Random/ Figure
Sin Cos - C.5	20	$556.34^{562.68}_{546.85}$	$290.53^{293.77}_{289.64}$	0.79/ 0.35
PCP 1 - C.6	493	$135.52^{136.12}_{134.87}$	$109.34^{110.72}_{108.25}$	0.76/ 0.74
PCP 2 - C.7	392	$36.06^{37.01}_{35.63}$	$25.15^{26.42}_{24.77}$	0.85/ 0.84
Andrews - C.8	1,000	$27.82^{29.65}_{27.58}$	$21.91^{22.27}_{20.63}$	0.91/ 0.88
Crop - C.9	1,200	$18.43^{19.41}_{17.99}$	$12.63^{13.48}_{11.98}$	0.93/ 0.84
Lens - C.10	150	$234.93^{241.47}_{221.16}$	$156^{158.39}_{152.60}$	0.75/ 0.73

Table C.1: Summary table of the analyzed test scenarios including name and figure number, the number of lines in the data set, the average frames per second (avg^{max}_{min}) for two common 16:9 screen resolutions, and the result of the overplotting measure.

C.7 Discussion and Limitations

Line charts are a well-established form of representation, not only in the field of visualization but also within a variety of other research fields. Although we have focused on traditional line charts such as time series plots or parallel coordinate plots in this paper, our basic approach is also suitable for other types of two-dimensional line and trajectory data. In particular, we believe that it would be interesting to examine its application to geospatial data, as well as circular visualizations such as radar charts. Our blending and rendering approach can already be applied to such scenarios in a straight forward way, but the line weaving algorithm, as presented in Section C.3.3, would have to be adapted to support general parametric curves.

Although edge bundling and line weaver are fundamentally different approaches (edge bundling reduces clutter by increasing overplotting and line weaver reduces clutter by decreasing overplotting), it would be interesting to conduct studies on how both techniques harmonize with each other. For example, by applying edge bundling as preprocessing step to initially reduce the overall number of lines, similar to techniques such as filtering or subsampling, and then smoothly blending the fewer resulting bundles using an importance function that optimizes their visibility.

We have provided examples of data sets containing up to 5 clusters. As soon as the number of clusters approaches the number of lines, for example, if lines cannot meaningfully be clustered into bundles, the ordering degenerates to an optimization of arc length only. In addition, it is recommended that clusters correspond to relatively compact bundles. Furthermore, as is valid for all techniques that rely on color mixing, the combination of several different colors may not always be easy to interpret. However, as our approach represents a hybrid between weaving and pure blending, the

visual continuity due to occlusion can help to better resolve such ambiguities. To further compensate for this, alternative blending operators such as hue-preserving color blending [36] could be used instead. In addition, it would be interesting to conduct studies in which various data properties are compared, in order to find an optimal and versatile importance function derived from data properties from various domains.

Throughout the paper we have used data sets with various numbers of lines, the largest of which consists of more than thousand lines. Currently, all lines are treated equally. They are blended according to the importance function and emphasized using halos. Nonetheless, it may be advantageous to introduce an additional level of detail for even larger data sets consisting of millions of lines. For example, halos and blending could only be applied to the most important clusters and less relevant ones could be averaged, aggregated, subsampled, or only displayed as confidence bands. This would simultaneously compensate for current limitations in computing performance on even larger data sets.

C.8 Conclusion

We have presented line weaver, a novel visualization technique for dense two-dimensional line data. Line weaver allows for an optimized blending, independent of the rendering order and without costly initial sorting. The basis of line weaver is formed by a quantitative importance function which either originates from external data or is derived from (geometric) properties of the lines such as arc length, and may vary locally. Using various synthetic as well as real-world data sets, we have shown the advantages of "woven" line bundles over visualizations that ignore the non-commutativity of blending or naively assume the blending order depends on how lines were stored in a file. Furthermore, we have shown that the use of quantitative importance functions can offer additional degrees of freedom for representing focus+context information in interactive scenarios. Finally, we have demonstrated that our approach can be implemented on modern GPU architectures to provide interactive, high-quality visualizations of line-based data sets.

Acknowledgments

The research presented in this paper was supported by the MetaVis project (#250133) funded by the Research Council of Norway.

Bibliography

[1] ABBAS, M. M., AUPETIT, M., SEDLMAIR, M., AND BENSMAIL, H. ClustMe: A Visual Quality Measure for Ranking Monochrome Scatterplots based on Cluster Patterns. *Computer Graphics Forum 38*, 3 (2019), 225–236, doi: 10.1111/cgf.13684. A.6.2

[2] ANDREWS, D. F. Plots of High-Dimensional Data. *Biometrics 28*, 1 (1972), 125–136, doi: 10.2307/2528964. 3.2, 3.21, C.5, C.5.2

[3] ANGORI, L., DIDIMO, W., MONTECCHIANI, F., PAGLIUCA, D., AND TAPPINI, A. *ChordLink: A New Hybrid Visualization Model*. Springer, 2019, pp. 276–290. 2.2

[4] APOSTU, O., MORA, F., GHAZANFARPOUR, D., AND AVENEAU, L. Analytic ambient occlusion using exact from-polygon visibility. *Computers & Graphics 36*, 6 (2012), 727–739, doi: 10.1016/j.cag.2012.04.008. B.3.1

[5] ARLT, R., LEUSSU, R., GIESE, N., MURSULA, K., AND USOSKIN, I. Sunspot positions and sizes for 1825-1867 from the observations by Samuel Heinrich Schwabe. *Monthly Notices of the Royal Astronomical Society 433*, 4 (2013), 3165—-3172, doi: 10.1093/mnras/stt961. A.1

[6] ARTERO, A. O., DE OLIVEIRA, M. C. F., AND LEVKOWITZ, H. Uncovering Clusters in Crowded Parallel Coordinates Visualizations. In *Proc. IEEE Symposium on Information Visualization* (2004), pp. 81–88, doi: 10.1109/INFVIS.2004.68. C.2.1

[7] AUPETIT, M., SEDLMAIR, M., ABBAS, M. M., AND BAGGAG, A. Toward Perception-Based Evaluation of Clustering Techniques for Visual Analytics. In *Proc. IEEE Visualization* (2019), pp. 141–145, doi: 10.1109/VISUAL.2019.8933620. A.6.2

[8] BACHTHALER, S., AND WEISKOPF, D. Continuous Scatterplots. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1428–1435, doi: 10.1109/TVCG.2008.119. A.2

[9] BACHTHALER, S., AND WEISKOPF, D. Efficient and Adaptive Rendering of 2-D Continuous Scatterplots. *Computer Graphics Forum 28*, 3 (2009), 743–750, doi: 10.1111/j.1467-8659.2009.01478.x. A.2

[10] BAIR, A., AND HOUSE, D. Grid With a View: Optimal Texturing for Perception of Layered Surface Shape. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1656–1663, doi: 10.1109/TVCG.2007.70559. 2.2, 2.12

[11] BALABANIAN, J.-P. *Multi-Aspect Visualization: Going from Linked Views to Integrated Views*. Department of Informatics, University of Bergen, Norway, 2009, pp. 1–141. 2.2

[12] BATTERSBY, S., STREBE, D., AND FINN, M. Shapes on a plane: Evaluating the impact of projection distortion on spatial binning. *Cartography and Geographic Information Science 44*, 5 (2016), 1–12, doi: 10.1080/15230406.2016.1180263. B.2, B.4.3

[13] BAVOIL, L., CALLAHAN, S. P., LEFOHN, A., COMBA, J. A. L. D., AND SILVA, C. T. Multi-Fragment Effects on the GPU Using the k-Buffer. In *Proc. Symposium on Interactive 3D Graphics and Games* (2007), pp. 97–104, doi: 10.1145/1230100.1230117. C.2.2

[14] BAVOIL, L., AND MYERS, K. Order Independent Transparency with Dual Depth Peeling. *NVIDIA* (2008). C.3.1

[15] BERTINI, E., AND SANTUCCI, G. Give Chance a Chance: Modeling Density to Enhance Scatter Plot Quality through Random Data Sampling. *Information Visualization 5*, 2 (2006), 95–110, doi: 10.1057/palgrave.ivs.9500122. A.2

[16] BLINN, J. F. Simulation of Wrinkled Surfaces. In *Proc. ACM SIGGRAPH* (1978), pp. 286–292, doi: 10.1145/800248.507101. 2.3

[17] BLUMENSCHEIN, M., ZHANG, X., POMERENKE, D., KEIM, D. A., AND FUCHS, J. Evaluating Reordering Strategies for Cluster Identification in Parallel Coordinates. *Computer Graphics Forum 39*, 3 (2020), 537–549, doi: 10.1111/cgf.14000. 3.2, 3.20, C.2.1, C.5.1, C.6

[18] BOLTE, F., NOURANI, M., RAGAN, E. D., AND BRUCKNER, S. SplitStreams: A Visual Metaphor for Evolving Hierarchies. *IEEE Transactions on Visualization and Computer Graphics 27*, 8 (2021), 3571–3584, doi: 10.1109/TVCG.2020.2973564. 2.3, 2.16

[19] BOUTREUX, C. The GeoNames geographical database. `http://www.geonames.org/`, Accessed: March 2019. A.5.2

[20] BRATH, R. 3D InfoVis is here to stay: Deal with it. In *Proc. IEEE VIS International Workshop on 3DVis* (2014), pp. 25–31, doi: 10.1109/3DVis.2014.7160096. 2.3

[21] BRAUN, J. Shape-from-shading is independent of visual attention and may be a 'texton'. *Spatial Vision 7*, 4 (1993), 311–322, doi: 10.1163/156856893X00469. 2.4, 2.26

[22] BREHMER, M., AND MUNZNER, T. A Multi-Level Typology of Abstract Visualization Tasks. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2376–2385, doi: 10.1109/TVCG.2013.124. 8, 3.1, B.7.1, B.7.1

[23] BRODBECK, D., CHALMERS, M., LUNZER, A., AND COTTURE, P. Domesticating Bead: Adapting an Information Visualization System to a Financial Institution. In *Proc. IEEE InfoVis* (1997), pp. 73–80, doi: 10.1109/INFVIS.1997.636789. A.2

[24] BRUCKNER, S., RAUTEK, P., VIOLA, I., ROBERTS, M., SOUSA, M. C., AND GRÖLLER, E. Hybrid visibility compositing and masking for illustrative rendering. *Computers & Graphics 34*, 4 (2010), 361–369, doi: 10.1016/j.cag.2010.04.003. C.2.2, C.3.1

[25] BÜRING, T., GERKEN, J., AND REITERER, H. User Interaction with Scatterplots on Small Screens - A Comparative Evaluation of Geometric-Semantic Zoom and Fisheye Distortion. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 829–836, doi: 10.1109/TVCG.2006.187. A.2

[26] CAKMAK, E., GÄRTNER, A., HEPP, T., BUCHMÜLLER, J., FISCHER, F., AND KEIM, D. A. Applying Visual Analytics to Explore and Analyze Movement Data. In *Proc. IEEE Conference on Visual Analytics Science and Technology* (2015), pp. 127–128, doi: 10.1109/VAST.2015.7347643. B.2

[27] CARPENDALE, M. S. T., COWPERTHWAITE, D. J., AND FRACCHIA, F. D. 3-Dimensional Pliable Surfaces: For the Effective Presentation of Visual Information. In *Proc. ACM UIST* (1995), pp. 217–226, doi: 10.1145/215585.215978. A.2

[28] CARPENTER, L. The A -Buffer, an Antialiased Hidden Surface Method. *ACM SIGGRAPH Computer Graphic 18*, 3 (1984), 103–108, doi: 10.1145/964965.808585. C.2.2

[29] CARR, D., LITTLEFIELD, R. J., NICHOLSON, W. L., AND LITTLEFIELD, J. S. Scatterplot Matrix Techniques for Large N. *Journal of the American Statistical Association 82*, 398 (1987), 424–436, doi: 10.2307/2289444. A.2, B.2, B.3.2

[30] CARR, D. B., OLSEN, A. R., AND WHITE, D. Hexagon Mosaic Maps for Display of Univariate and Bivariate Geographical Data. *Cartography and Geographic Information Systems 19*, 4 (1992), 228–236, doi: 10.1559/152304092783721231. B.2, B.4.3

[31] CHAMBERS, J. M., CLEVELAND, W. S., KLEINER, B., AND TUKEY, P. A. *Graphical Methods for Data Analysis*. Chapman and Hall/Cole Publishing Company, 1983, p. 107. A.2

[32] CHEN, H., CHEN, W., MEI, H., LIU, Z., ZHOU, K., CHEN, W., GU, W., AND MA, K.-L. Visual Abstraction and Exploration of Multi-class Scatterplots. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1683–1692, doi: 10.1109/TVCG.2014.2346594. A.2

[33] CHEN, H., ENGLE, S., JOSHI, A., RAGAN, E. D., YUKSEL, B. F., AND HARRISON, L. Using Animation to Alleviate Overdraw in Multiclass Scatterplot Matrices. In *Proc. ACM CHI* (2018), pp. 417:1–417:12, doi: 10.1145/3173574.3173991. A.2

[34] CHEN, M., WALTON, S., BERGER, K., THIYAGALINGAM, J., DUFFY, B., FANG, H., HOLLOWAY, C., AND TREFETHEN, A. E. Visual Multiplexing. *Computer Graphics Forum 33*, 3 (2014), 241–250, doi: 10.1111/cgf.12380. 2.3, 2.1

[35] CHEN, X., GE, T., ZHAN, J., CHEN, B., FU, C.-W., DEUSSEN, O., AND WANG, Y. A Recursive Subdivision Technique for Sampling Multi-class Scatterplots. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2019), 729–738, doi: 10.1109/TVCG.2019.2934541. A.2

[36] CHUANG, J., WEISKOPF, D., AND MÖLLER, T. Hue-Preserving Color Blending. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1275–1282, doi: 10.1109/TVCG.2009.150. C.7

[37] CLEVELAND, W., MCGILL, M. E., AND MCGILL, R. The Shape Parameter of a Two-Variable Graph. *Journal of the American Statistical Association 83*, 402 (1988), 289–300, doi: 10.1080/01621459.1988.10478598. C.2.2

[38] CLEVELAND, W. S., AND MCGILL, R. The Many Faces of a Scatterplot. *Journal of the American Statistical Association 79*, 338 (1984), 807–822, doi: 10.2307/2288711. A.2, B.2

[39] CLEVELAND, W. S., AND MCGILL, R. Graphical Perception and Graphical Methods for Analyzing Scientific Data. *Science 229*, 4716 (1985), 828–833, doi: 10.1126/science.229.4716.828. B.1

[40] COLLINS, C., AND CARPENDALE, S. VisLink: Revealing Relationships Amongst Visualizations. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1192–1199, doi: 10.1109/TVCG.2007.70521. 2.5, 2.2

[41] COLLINS, C., PENN, G., AND CARPENDALE, S. Bubble Sets: Revealing Set Relations with Isocontours over Existing Visualizations. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1009–1016, doi: 10.1109/TVCG.2009.122. 2.9, 2.2

[42] CORNUT, O. Dear ImGui. `https://github.com/ocornut/imgui`, 2021. Accessed: August. A.4, B.5, C.4

[43] DAAE LAMPE, O., AND HAUSER, H. Curve Density Estimates. *Computer Graphics Forum 30*, 3 (2011), 633–642, doi: 10.1111/j.1467-8659.2011.01912.x. C.2.1

[44] DAAE LAMPE, O., AND HAUSER, H. Interactive Visualization of Streaming Data with Kernel Density Estimation. In *Proc. IEEE PacificVis* (2011), pp. 171–178, doi: 10.1109/PACIFICVIS.2011.5742387. A.2, C.2.1

[45] DANG, T. N., WILKINSON, L., AND ANAND, A. Stacking Graphic Elements to Avoid Over-Plotting. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1044–1052, doi: 10.1109/TVCG.2010.197. A.2, A.2

[46] DE SOUSA, L. M., AND LEITÃO, J. P. HexASCII: A file format for cartographical hexagonal rasters. *Transactions in GIS. 22*, 1 (2018), 217–232, doi: 10.1111/tgis.12304. B.4.3

[47] DEMŠAR, U., AND VIRRANTAUS, K. Space–time density of trajectories: exploring spatio-temporal patterns in movement data. *International Journal of Geographical Information Science 24*, 10 (2010), 1527–1542, doi: 10.1080/13658816.2010.511223. C.2.1

[48] DIX, A., AND ELLIS, G. By Chance Enhancing Interaction with Large Data Sets Through Statistical Sampling. In *Proc. Working Conference on Advanced Visual Interfaces* (2002), pp. 167–176, doi: 10.1145/1556262.1556289. A.2

[49] DOWNEY, A. *Think Stats: Exploratory Data Analysis*. O'Reilly Media, 2014, pp. 91–95. B.1

[50] DUA, D., AND GRAFF, C. UCI Machine Learning Repository. `http://archive.ics.uci.edu/ml`, 2021. Accessed: March. C.7, C.5.2

[51] DUPIN, C. *Tableau des arts et métiers et des beaux-arts: présenté pour servir à propager l'institution des cours de géométrie et de méchanique appliquées aux arts, dans les villes de la France*. Bachelier, Paris, 1826. B.3

[52] DUPONT, W. S., AND PLUMMER, W. D. J. Density Distribution Sunflower Plots. *Journal of Statistical Software 8*, 3 (2003), 1–5, doi: 10.18637/jss.v008.i03. A.2

[53] DUTRÉ, P. Global Illumination Compendium - The Concise Guide to Global Illumination Algorithms. `https://people.cs.kuleuven.be/~philip.dutre/GI/`, 2021. Accessed: November. B.3.1

[54] EICHELBAUM, S., HLAWITSCHKA, M., AND SCHEUERMANN, G. LineAO–Improved Three-Dimensional Line Rendering. *IEEE Transactions on Visualization and Computer Graphics 19*, 3 (2013), 433–445, doi: 10.1109/TVCG.2012.142. C.2.1

[55] ELLIS, G., BERTINI, E., AND DIX, A. The Sampling Lens: Making Sense of Saturated Visualisations. In *Proc. ACM CHI Extended Abstracts* (2005), pp. 1351–1354, doi: 10.1145/1056808.1056914. A.2

[56] ELLIS, G., AND DIX, A. Density Control Through Random Sampling: an Architectural Perspective. In *Proc. International Conference on Information Visualisation* (2002), pp. 82–90, doi: 10.1109/IV.2002.1028760. A.2

[57] ELLIS, G., AND DIX, A. A Taxonomy of Clutter Reduction for Information Visualisation. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1216–1223, doi: 10.1109/TVCG.2007.70535. A.2

[58] ENNS, J. T., AND RENSINK, R. A. Influence of Scene-Based Properties on Visual Search. *Science 247*, 4943 (1990), 721–723, doi: 10.1126/science.2300824. 2.4, 2.28, 2.4

[59] EVERTS, M. H., BEKKER, H., ROERDINK, J. B. T. M., AND ISENBERG, T. Depth-Dependent Halos: Illustrative Rendering of Dense Line Data. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1299–1306, doi: 10.1109/TVCG.2009.138. 2.3, 2.22, C.2.2

[60] FEKETE, J.-D., WANG, D., DANG, N., AND PLAISANT, C. Overlaying Graph Links on Treemaps. *Information Visualization* (2003). 2.7, 2.2

[61] FRIENDLY, M., AND DENIS, D. The early origins and development of the scatterplot. *Journal of the History of the Behavioral Sciences 41*, 2 (2005), 103–130, doi: 10.1002/jhbs.20078. A.1

[62] FUA, Y.-H., WARD, M. O., AND RUNDENSTEINER, E. A. Hierarchical Parallel Coordinates for Exploration of Large Datasets. In *Proc. IEEE Visualization* (1999), pp. 43–50, doi: 10.1109/VISUAL.1999.809866. C.2.1

[63] GIBSON, J. J. The Perception of Visual Surfaces. *The American Journal of Psychology 63*, 3 (1950), 367–384, doi: 10.2307/1418003. A.3.2

[64] GROSS, D., AND GUMHOLD, S. Advanced Rendering of Line Data with Ambient Occlusion and Transparency. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2021), 614–624, doi: 10.1109/TVCG.2020.3028954. C.2.2, C.3.2

[65] GÜNTHER, T., RÖSSL, C., AND THEISEL, H. Opacity Optimization for 3D Line Fields. *ACM Transactions on Graphics 32*, 4 (2013), 120:1–120:8, doi: 10.1145/2461912.2461930. C.2.2

[66] GÜNTHER, T., RÖSSL, C., AND THEISEL, H. Hierarchical opacity optimization for sets of 3D line fields. *Computer Graphics Forum 33*, 2 (2014), 507–516, doi: 10.1111/cgf.12336. C.2.2

[67] GÜNTHER, T., THEISEL, H., AND GROSS, M. Decoupled Opacity Optimization for Points, Lines and Surfaces. *Computer Graphics Forum 36*, 2 (2017), 153–162, doi: 10.1111/cgf.13115. C.2.2

[68] GUO, H., XIAO, H., AND YUAN, X. Scalable Multivariate Volume Visualization and Analysis Based on Dimension Projection and Parallel Coordinates. *IEEE Transactions on Visualization and Computer Graphics 18*, 9 (2012), 1397–1410, doi: 10.1109/TVCG.2012.80. 2.9, 2.2

[69] HAGH-SHENAS, H., KIM, S., INTERRANTE, V., AND HEALEY, C. Weaving Versus Blending: a quantitative assessment of the information carrying capacities of two alternative methods for conveying multivariate data with color. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1270–1277, doi: 10.1109/TVCG.2007.70623. B.2, C.2.2

[70] HANAZAWA, A., AND KOMATSU, H. Influence of the Direction of Elemental Luminance Gradients on the Responses of V4 Cells to Textured Surfaces. *The Journal of Neuroscience 21*, 12 (2001), 4490–4497, doi: 10.1523/JNEUROSCI.21-12-04490.2001. 2.4, 2.27

[71] HAO, M., DAYAL, U., SHARMA, R., KEIM, D. A., AND JANETZKO, H. Variable Binned Scatter Plots. *Information Visualization 9*, 3 (2010), 194–203, doi: 10.1057/ivs.2010.4. A.2

[72] HARROWER, M., AND BREWER, C. ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. *The Cartographic Journal 40*, 1 (2003), 27–37, doi: 10.1179/000870403235002042. B.3.1

[73] HARTIGAN, J. A. Printer graphics for clustering. *Journal of Statistical Computation and Simulation 4*, 3 (1975), 187–213, doi: 10.1080/00949657508810123. A.2

[74] HAUSER, H. *Generalizing Focus+Context Visualization*. Springer, 2006, pp. 305–327. C.5, C.5.3

[75] HAUSER, H., LEDERMANN, F., AND DOLEISCH, H. Angular Brushing of Extended Parallel Coordinates. In *Proc. IEEE Symposium on Information Visualization* (2002), pp. 127–130, doi: 10.1109/INFVIS.2002.1173157. C.5, C.5.3

[76] HEIMERL, F., CHANG, C., SARIKAYA, A., AND GLEICHER, M. Visual Designs for Binned Aggregation of Multi-Class Scatterplots. *CoRR abs/1810.02445* (2018). B.2

[77] HENRY, N., AND FEKETE, J.-D. MatrixExplorer: a Dual-Representation System to Explore Social Networks. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 677–684, doi: 10.1109/TVCG.2006.160. 2.2, 2.6

[78] HENRY, N., FEKETE, J.-D., AND MCGUFFIN, M. J. NodeTrix: a Hybrid Visualization of Social Networks. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1302–1309, doi: 10.1109/TVCG.2007.70582. 2.6, 2.2

[79] HOLTEN, D., VLIEGEN, R., AND VAN WIJK, J. Visual realism for the visualization of software metrics. In *Proc. IEEE International Workshop on Visualizing Software for Understanding and Analysis* (2005), pp. 1–6, doi: 10.1109/VISSOF.2005.1684299. 2.3, 2.14

[80] HUANG, C., MCDONALD, J. A., AND STUETZLE, W. Variable Resolution Bivariate Plots. *Journal of Computational and Graphical Statistics 6*, 4 (1997), 383–396, doi: 10.1080/10618600.1997.10474749. A.2, B.2

[81] HUANG, X., YE, Y., XIONG, L., LAU, R. Y., JIANG, N., AND WANG, S. Time series k-means: A new k-means type smooth subspace clustering for time series data. *Information Sciences 367-368* (2016), 1–13, doi: 10.1016/j.ins.2016.05.040. C.5.3, C.10

[82] HUNTER, J. D. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering 9*, 3 (2007), 90–95, doi: 10.1109/MCSE.2007.55. A.3, A.4, A.5

[83] IRANI, P., SLONOWSKY, D., AND SHAJAHAN, P. The Effect of Shading in Extracting Structure from Space-Filling Visualizations. In *Proc. International Conference on Information Visualisation* (2004), pp. 209–216, doi: 10.1109/IV.2004.1320146. B.3

[84] IRANI, P., SLONOWSKY, D., AND SHAJAHAN, P. Human Perception of Structure in Shaded Space-Filling Visualizations. *Information Visualization 5*, 1 (2006), 47–61, doi: 10.1057/palgrave.ivs.9500113. 2.3, 2.20, B.3

[85] IRANI, P., AND WARE, C. Diagramming Information Structures Using 3D Perceptual Primitives. *ACM CHI 10*, 1 (2003), 1–19, doi: 10.1145/606658.606659. 2.3, 2.19

[86] JANETZKO, H., HAO, M. C., MITTELSTÄDT, S., DAYAL, U., AND KEIM, D. A. Enhancing Scatter Plots Using Ellipsoid Pixel Placement and Shading. In *Proc. Annual Hawaii International Conference on System Sciences* (2013), pp. 1522–1531, doi: 10.1109/HICSS.2013.197. A.2

[87] JAVED, W., AND ELMQVIST, N. Exploring the design space of composite visualization. In *Proc. IEEE PacificVis* (2012), pp. 1–8, doi: 10.1109/PacificVis.2012.6183556. 2.1, 2.1

[88] JO, J., VERNIER, F., DRAGICEVIC, P., AND FEKETE, J.-D. A Declarative Rendering Model for Multiclass Density Maps. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 470–480, doi: 10.1109/TVCG.2018.2865141. B.2

[89] JOHANSSON, J., LJUNG, P., JERN, M., AND COOPER, M. Revealing Structure in Visualizations of Dense 2D and 3D Parallel Coordinates. *Information Visualization 5*, 2 (2006), 125–136, doi: 10.1057/palgrave.ivs.9500117. C.2.1

[90] KEIM, D. A. Designing Pixel-oriented Visualization Techniques : Theory and Applications. *IEEE Transactions on Visualization and Computer Graphics 6*, 1 (2000), 59–78, doi: 10.1109/2945.841121. A.2

[91] KEIM, D. A., HAO, M. C., DAYAL, U., JANETZKO, H., AND BAK, P. Generalized Scatter Plots. *Information Visualization 9*, 4 (2009), 301–311, doi: 10.1057/ivs.2009.34. A.2

[92] KEIM, D. A., AND HERRMANN, A. The Gridfit Algorithm: An Efficient and Effective Approach to Visualizing Large Amounts of Spatial Data. In *Proc. IEEE Visualization* (1998), pp. 181–188, doi: 10.1109/VISUAL.1998.745301. A.2

[93] KERN, M., NEUHAUSER, C., MAACK, T., HAN, M., USHER, W., AND WESTERMANN, R. A Comparison of Rendering Techniques for 3D Line Sets with Transparency. *IEEE Transactions on Visualization and Computer Graphics 27*, 8 (2021), 3361–3376, doi: 10.1109/TVCG.2020.2975795. C.2.2

[94] KILGARD, M. J. Polar Stroking: New Theory and Methods for Stroking Paths. *ACM Transactions on Graphics 39*, 4 (2020), 145:1–145:15, doi: 10.1145/3386569.3392458. C.2.2

[95] KIM, S., MACIEJEWSKI, R., MALIK, A., JANG, Y., EBERT, D. S., AND ISENBERG, T. Bristle Maps: A Multivariate Abstraction Technique for Geovisualization. *IEEE Transactions on Visualization and Computer Graphics 19*, 9 (2013), 1438–1454, doi: 10.1109/TVCG.2013.66. 2.3, 2.16

[96] KINDLMANN, G., AND SCHEIDEGGER, C. An Algebraic Process for Visualization Design. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2181–2190, doi: 10.1109/TVCG.2014.2346325. 3.2, B.1, C.3

[97] KONYHA, Z., LEŽ, A., MATKOVIĆ, K., JELOVIĆ, M., AND HAUSER, H. Interactive Visual Analysis of Families of Curves Using Data Aggregation and Derivation. In *Proc. Conference on Knowledge Management and Knowledge Technologies* (2012), pp. 24:1–24:8, doi: 10.1145/2362456.2362487. C.2.1

[98] KRAUS, M., ANGERBAUER, K., BUCHMÜLLER, J., SCHWEITZER, D., KEIM, D. A., SEDLMAIR, M., AND FUCHS, J. Assessing 2D and 3D Heatmaps for Comparative Analysis: An Empirical Study. In *Proc. Conference on Human Factors in Computing Systems* (2020), pp. 1–14, doi: 10.1145/3313831.3376675. B.2, B.7.1

[99] KWON, O.-H., MUELDER, C., LEE, K., AND MA, K.-L. A Study of Layout, Rendering, and Interaction Methods for Immersive Graph Visualization. *IEEE Transactions on Visualization and Computer Graphics 22*, 7 (2016), 1802–1815, doi: 10.1109/TVCG.2016.2520921. C.2.1

[100] LANGER, M. S., AND BÜLTHOFF, H. H. A Prior for Global Convexity in Local Shape-from-Shading. *Perception 30*, 4 (2001), 403–410, doi: 10.1068/p3178. B.3

[101] LECUN, Y., CORTES, C., AND CHRISTOPHER J.C., B. The MINST Database for Handwritten digits. http://yann.lecun.com/exdb/mnist/, Accessed: February 2020. A.5.3

[102] LI, C., BACIU, G., AND HAN, Z. StreamMap: Smooth Dynamic Visualization of High-Density Streaming Points. *IEEE Transactions on Visualization and Computer Graphics 24*, 3 (2018), 1381–1393, doi: 10.1109/TVCG.2017.2668409. A.2

[103] LI, J., MARTENS, J.-B., AND VAN WIJK, J. J. A Model of Symbol Size Discrimination in Scatterplots. In *Proc. ACM CHI* (2010), pp. 2553–2562, doi: 10.1145/1753326.1753714. A.2

[104] LI, J., VAN WIJK, J. J., AND MARTENS, J.-B. Evaluation of Symbol Contrast in Scatterplots. In *Proc. IEEE PacificVis* (2009), pp. 97–104, doi: 10.1109/PACIFICVIS.2009.4906843. A.2

[105] LI, J., VAN WIJK, J. J., AND MARTENS, J.-B. A Model of Symbol Lightness Discrimination in Sparse Scatterplots. In *Proc. IEEE PacificVis* (2010), pp. 105–112, doi: 10.1109/PACIFICVIS.2010.5429604. A.2

[106] LOKUGE, I., AND ISHIZAKI, S. GeoSpace: An Interactive Visualization System for Exploring Complex Information Spaces. In *Proc. ACM CHI* (1995), pp. 409–414, doi: 10.1145/223904.223959. 2.2, 2.8

[107] LUBOSCHIK, M., RADLOFF, A., AND SCHUMANN, H. A New Weaving Technique for Handling Overlapping Regions. In *Proc. International Conference on Advanced Visual Interfaces* (2010), pp. 25–32, doi: 10.1145/1842993.1842999. C.2.2

[108] LUFT, T., COLDITZ, C., AND DEUSSEN, O. Image Enhancement by Unsharp Masking the Depth Buffer. *ACM SIGGRAPH Computer Graphic 25*, 3 (2006), 1206–1213, doi: 10.1145/1179352.1142016. 2.3, 2.21, C.2.2, C.3.2, C.4

[109] LUFT, T., AND DEUSSEN, O. Real-Time Watercolor Illustrations of Plants Using a Blurred Depth Test. In *Proc. Symposium on Non-Photorealistic Animation and Rendering* (2006), pp. 11–20, doi: 10.1145/1124728.1124732. C.2.2, C.3.1

[110] MAHMOOD, S., AND MUELLER, K. An Exploded View Paradigm to Disambiguate Scatterplots. *Computers & Graphics 73* (2018), 37–46, doi: 10.1016/j.cag.2018.02.008. A.2

[111] MALIK, M. M., HEINZL, C., AND GRÖLLER, E. Comparative Visualization for Parameter Studies of Dataset Series. *IEEE Transactions on Visualization and Computer Graphics 16*, 5 (2010), 829–840, doi: 10.1109/TVCG.2010.20. B.2

[112] MATEJKA, J., ANDERSON, F., AND FITZMAURICE, G. Dynamic Opacity Optimization for Scatter Plots. In *Proc. ACM CHI* (2015), pp. 2707–2710, doi: 10.1145/2702123.2702585. A.2

[113] MATKOVIC, K., FREILER, W., GRACANIN, D., AND HAUSER, H. ComVis: A Coordinated Multiple Views System for Prototyping New Visualization Technology. In *Proc. International Conference Information Visualisation* (2008), pp. 215–220, doi: 10.1109/IV.2008.87. 2.2, 2.4

[114] MAYORGA, A., AND GLEICHER, M. Splatterplots: Overcoming Overdraw in Scatter Plots. *IEEE Transactions on Visualization and Computer Graphics 19*, 9 (2013), 1526–1538, doi: 10.1109/TVCG.2013.65. A.2, A.3.1, B.3.3

[115] MCGUIRE, M., MARA, M., AND LUEBKE, D. Scalable Ambient Obscurance. In *Proc. ACM High-Performance Graphics* (2012), pp. 97–103, doi: 10.2312/EGGH/HPG12/097-103. A.3.2, A.4, A.9

[116] MCNUTT, A. What are Table Cartograms Good for Anyway? An Algebraic Analysis. *Computer Graphics Forum 40*, 3 (2021), 61–73, doi: 10.1111/cgf.14289. B.1

[117] MICALLEF, L., PALMAS, G., OULASVIRTA, A., AND WEINKAUF, T. Towards Perceptual Optimization of the Visual Design of Scatterplots. *IEEE Transactions on Visualization and Computer Graphics 23*, 6 (2017), 1588–1599, doi: 10.1109/TVCG.2017.2674978. A.2

[118] MIKSCH, S., AND AIGNER, W. A matter of time: Applying a data–users–tasks design triangle to visual analytics of time-oriented data. *Computers & Graphics 38* (2014), 286–290, doi: 10.1016/j.cag.2013.11.002. 1

[119] MILL, B. Boston Marathon Raw Data. https://github.com/llimllib, Accessed: February 2020. 3.1.1, 3.6, A.5.1

[120] MILLER, J. R. Attribute Blocks: Visualizing Multiple Continuously Defined Attributes. *IEEE Computer Graphics and Applications 27*, 3 (2007), 57–69, doi: 10.1109/MCG.2007.54. B.2

[121] MORITZ, D., AND FISHER, D. Visualizing a Million Time Series with the Density Line Chart. *CoRR abs/1808.06019* (2018). C.2.1

[122] MUNZNER, T. *Visualization Analysis and Design*. CRC Press, 2015, pp. 95–116. B.3

[123] NAKAYA, T., AND YANO, K. Visualising Crime Clusters in a Space-time Cube: An Exploratory Data-analysis Approach Using Space-time Kernel Density Estimation and Scan Statistics. *Transactions in GIS 14*, 3 (2010), 223–239, doi: 10.1111/j.1467-9671.2010.01194.x. C.2.1

[124] NOAA'S NATIONAL WEATHER SERVICE - STORM PREDICTION CENTER. Severe Weather Database Files (1950-2019). https://www.spc.noaa.gov/wcm/, 2021. Accessed: August. 3.1.2, 3.12, B.4.1

[125] NOVOTNY, M., AND HAUSER, H. Outlier-Preserving Focus+Context Visualization in Parallel Coordinates. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 893–900, doi: 10.1109/TVCG.2006.170. B.3.3, C.2.1

[126] PADILLA, L., QUINAN, P., MEYER, M., AND CREEM-REGEHR, S. H. Evaluating the Impact of Binning 2D Scalar Fields. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 431–440, doi: 10.1109/TVCG.2016.2599106. B.7.1

[127] PALKE, D., LIN, Z., CHEN, G., YEH, H., VINCENT, P., LARAMEE, R., AND ZHANG, E. Asymmetric Tensor Field Visualization for Surfaces. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 1979–1988, doi: 10.1109/TVCG.2011.170. 2.10, 2.2

[128] PASSONNEAU, J. R., AND WURMAN, R. S. *Urban Atlas : 20 American Cities*. (Boston) M.I.T. Press, 1966. B.2

[129] PHONG, B. T. Illumination for Computer Generated Pictures. *Communications of the ACM 18*, 6 (1975), 311–317, doi: 10.1145/360825.360839. A.3.2, A.4, A.9

[130] PLAYFAIR, W. *The Commercial and Political Atlas and Statistical Breviary*. Cambridge University Press, 2005. A.2

[131] POLISCIUC, E., MACAS, C., ASSUNCAO, F., AND MACHADO, P. Hexagonal Gridded Maps and Information Layers: A Novel Approach for the Exploration and Analysis of Retail Data. In *Proc. SIGGRAPH ASIA Symposium on Visualization* (2016), pp. 1–8, doi: 10.1145/3002151.3002160. B.2, B.4.3

[132] PORTER, T., AND DUFF, T. Compositing Digital Images. *ACM SIGGRAPH Computer Graphic 18*, 3 (1984), 253–259, doi: 10.1145/964965.808606. A.3.3, B.3.3, C.2.1, C.3.1

[133] QUILEZ, I. Useful maths - Box ambient occlusions. https://www.iquilezles.org/www/articles/boxocclusion/boxocclusion.htm, 2021. Accessed: November. B.3.1

[134] RAIDOU, R. G., GRÖLLER, E., AND EISEMANN, M. Relaxing Dense Scatter Plots with Pixel-Based Mappings. *IEEE Transactions on Visualization and Computer Graphics 25*, 6 (2019), 2205–2216, doi: 10.1109/TVCG.2019.2903956. A.2

[135] RAMACHANDRAN, V. Perception of Shape from Shading. *Nature 331* (1988), 163–166, doi: 10.1038/331163a0. 2.4, 2.25, A.2

[136] ROBERTSON, G. G., MACKINLAY, J. D., AND CARD, S. K. Cone Trees: Animated 3D Visualizations of Hierarchical Information. In *Proc. ACM CHI* (1991), pp. 189–194, doi: 10.1145/108844.108883. 2.3, 2.18

[137] ROSEN, P., AND QUADRI, G. J. LineSmooth: An Analytical Framework for Evaluating the Effectiveness of Smoothing Techniques on Line Charts. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2021), 1536–1546, doi: 10.1109/TVCG.2020.3030421. C.2.2

[138] ROTH, R., ROSS, K., AND MACEACHREN, A. User-Centered Design for Interactive Maps: A Case Study in Crime Analysis. *International Journal of GeoInformation 4* (2015), 262–301, doi: 10.3390/ijgi4010262. B.2

[139] RUSINKIEWICZ, S., BURNS, M., AND DECARLO, D. Exaggerated Shading for Depicting Shape and Detail. In *Proc. ACM SIGGRAPH* (2006), pp. 1199–1205, doi: 10.1145/1179352.1142015. 2.3, 2.23

[140] RYAN, G., MOSCA, A., CHANG, R., AND WU, E. At a Glance: Pixel Approximate Entropy as a Measure of Line Chart Complexity. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2018), 872–881, doi: 10.1109/TVCG.2018.2865264. C.2.1

[141] SARIKAYA, A., AND GLEICHER, M. Scatterplots: Tasks, Data, and Designs. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 402–412, doi: 10.1109/TVCG.2017.2744184. 3.1, A.2, A.3, A.6.1, A.6.2, A.6.4

[142] SCHULZ, H.-J., AND HADLAK, S. Preset-based generation and exploration of visualization designs. *Journal of Visual Languages & Computing 31* (2015), 9–29, doi: 10.1016/j.jvlc.2015.09.004. 2.1, 2.2

[143] SCHUSSMAN, G., AND MA, K.-L. Anisotropic volume rendering for extremely dense, thin line data. In *Proc. IEEE Visualization* (2004), pp. 107–114, doi: 10.1109/VISUAL.2004.5. C.2.1

[144] SHNEIDERMAN, B., AND ARIS, A. Network Visualization by Semantic Substrates. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 733–740, doi: 10.1109/TVCG.2006.166. 2.5, 2.2

[145] SILVERMAN, B. W. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall/CRC, 1986. A.3.1

[146] SPEAR, M. E. *Charting Statistics*. McGraw-Hill, 1952, pp. 39–95. C.1, C.2.2

[147] SPENCE, I. Visual Psychophysics of Simple Graphical Elements. *Journal of Experimental Psychology: Human Perception and Performance 16*, 4 (1990), 683–92, doi: 10.1037/0096-1523.16.4.683. 2.4, 2.31

[148] SPRENGER, T. C., BRUNELLA, R., AND GROSS, M. H. H-BLOB: A Hierarchical Visual Clustering Method Using Implicit Surface. In *Proc. IEEE Visualization* (2000), pp. 61–68, doi: 10.1109/VISUAL.2000.885677. A.2

[149] STAIB, J., GROTTEL, S., AND GUMHOLD, S. Enhancing Scatterplots with Multi-Dimensional Focal Blur. *Computer Graphics Forum 35*, 3 (2016), 11–20, doi: 10.1111/cgf.12877. A.2

[150] STONE, M. In Color Perception, Size Matters. *IEEE Computer Graphics and Applications 32*, 2 (2012), 8–13, doi: 10.1109/MCG.2012.37. B.1

[151] STONE, M., ALBERS SZAFIR, D., AND SETLUR, V. An Engineering Model for Color Difference as a Function of Size. In *Proc. Color and Imaging Conference* (2014), vol. 6, pp. 253–258. B.1

[152] STONE, M. C., FISHKIN, K., AND BIER, E. A. The Movable Filter As a User Interface Tool. In *Proc. ACM CHI* (1994), pp. 306–312, doi: 10.1145/191666.191774. A.2, C.5, C.5.3

[153] SUN, J., AND PERONA, P. Preattentive Perception of Elementary Three Dimensional Shapes. *Vision Research 36*, 16 (1996), 2515–2529, doi: 10.1016/0042-6989(95)00336-3. 2.4, 2.29

[154] SZAFIR, D. A., SARIKAYA, A., AND GLEICHER, M. Lightness Constancy in Surface Visualization. *IEEE Transactions on Visualization and Computer Graphics 22*, 9 (2016), 2107–2121, doi: 10.1109/TVCG.2015.2500240. B.3

[155] TABLEAU SOFTWARE. Control the Appearance of Marks in the View. `https://help.tableau.com/current/pro/desktop/en-gb/viewparts_marks_markproperties.htm#draw-paths-between-marks`, 2020. Accessed: October. C.1

[156] TAN, C. W., WEBB, G. I., AND PETITJEAN, F. Indexing and classifying gigabytes of time series under time warping. In *Proc. SIAM International Conference on Data Mining* (2017), pp. 282–290, doi: 10.1137/1.9781611974973.32. 3.15, C.5.2, C.9

[157] TELEA, A., AND WIJK, VAN, J. J. Visualization of Generalized Voronoi Diagrams. In *Proc. IEEE TCVG Symposium on Visualization* (2001), pp. 165–174, doi: 10.1007/978-3-7091-6215-6_18. 2.3, 2.15, B.3

[158] THEUS, M. *Graphics of Large Data Sets: Visualizing a Million - Scaling Up Graphics*. Springer, 2006, pp. 55–72. A.2

[159] THOMSON, G., AND MACPHERSON, F. Adelson's checker-shadow illusion. https://www.illusionsindex.org/ir/checkershadow, Accessed: December 2019. A.3.2

[160] TOMINSKI, C., GLADISCH, S., KISTER, U., DACHSELT, R., AND SCHUMANN, H. Interactive Lenses for Visualization: An Extended Survey. *Computer Graphics Forum 36*, 6 (2017), 173–200, doi: 10.1111/cgf.12871. A.2

[161] TORGO L. - DEPARTMENT OF COMPUTER SCIENCE OF THE FACULTY OF SCIENCES OF THE UNIVERSITY OF PORTO. California Housing Data (1990) - California Housing Price Prediction. https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html, 2021. Accessed: August. 3.1.2, 3.13, B.4.2

[162] TORY, M., SPRAGUE, D., WU, F., SO, W. Y., AND MUNZNER, T. Spatialization Design: Comparing Points and Landscapes. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1262–1269, doi: 10.1109/TVCG.2007.70596. A.3.2

[163] TRAUTNER, T., BOLTE, F., STOPPEL, S., AND BRUCKNER, S. Sunspot Plots: Model-based Structure Enhancement for Dense Scatter Plots. *Computer Graphics Forum 39*, 3 (2020), 551–563, doi: 10.1111/cgf.14001. B.3, B.3.3, C.2.1

[164] TRAUTNER, T., AND BRUCKNER, S. Line Weaver: Importance-Driven Order Enhanced Rendering of Dense Line Charts. *Computer Graphics Forum 40*, 3 (2021), 399–410, doi: 10.1111/cgf.14316. B.3

[165] TREAVETT, S., AND CHEN, M. Pen-and-Ink Rendering in Volume Visualisation. In *Proc. IEEE Visualization* (2000), pp. 203–210, doi: 10.1109/VISUAL.2000.885696. 2.2, 2.12

[166] TRUMBO, B. E. A Theory for Coloring Bivariate Statistical Maps. *The American Statistician 35*, 4 (1981), 220–226, doi: 10.2307/2683294. A.2

[167] TRUTSCHL, M., GRINSTEIN, G. G., AND CVEK, U. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In *Proc. ACM CHI* (1994), pp. 313–317, doi: 10.1145/191666.191775. A.2

[168] TRUTSCHL, M., GRINSTEIN, G. G., AND CVEK, U. Intelligently Resolving Point Occlusion. *IEEE Symposium on Information Visualization* (2003), 131–136, doi: 10.1109/INFVIS.2003.1249018. A.2

[169] VAN LIERE, R., AND DE LEEUW, W. GraphSplatting: Visualizing Graphs as Continuous Fields. *IEEE Transactions on Visualization and Computer Graphics 9*, 2 (2003), 206–212, doi: 10.1109/TVCG.2003.1196007. A.2

[170] VAN WIJK, J. J., AND TELEA, A. Enridged Contour Maps. In *Proc. IEEE Visualization* (2001), pp. 69–543, doi: 10.1109/VISUAL.2001.964495. 2.3, 2.17, A.2, B.3

[171] VAN WIJK, J. J., AND VAN DE WETERING, H. Cushion Treemaps: Visualization of Hierarchical Information. In *Proc. IEEE Symposium on Information Visualization* (1999), pp. 73–78, doi: 10.1109/INFVIS.1999.801860. 2.3, 2.13

[172] VAN WIJK, J. J., AND VAN DE WETERING, H. Cushion Treemaps: Visualization of Hierarchical Information. In *Proc. IEEE Symposium on Information Visualization* (1999), pp. 73–78, doi: 10.1109/INFVIS.1999.801860. B.3

[173] VERGNE, R., PACANOWSKI, R., BARLA, P., GRANIER, X., AND SCHLICK, C. Light Warping for Enhanced Surface Depiction. In *Proc. ACM SIGGRAPH* (2009), pp. 1–8, doi: 10.1145/1576246.1531331. 2.3, 2.24

[174] VIOLA, I., KANITSAR, A., AND GROLLER, M. Importance-Driven Volume Rendering. In *Proc. IEEE Visualization* (2004), pp. 139–145, doi: 10.1109/VISUAL.2004.48. 2.2

[175] VOISARD, A. Mapgets: A Tool for Visualizing and Querying Geographic Information. *Journal of Visual Languages & Computing 6*, 4 (1995), 367–384, doi: 10.1006/jvlc.1995.1021. 2.2, 2.8

[176] WALDECK, C., AND BALFANZ, D. Mobile Liquid 2D Scatter Space (ML2DSS). In *Proc. International Conference on Information Visualisation* (2004), pp. 494–498, doi: 10.1109/IV.2004.1320190. A.2

[177] WALLNER, G., AND KRIGLSTEIN, S. Multivariate Visualization of Game Metrics: An Evaluation of Hexbin Maps. In *Proc. Annual Symposium on Computer-Human Interaction in Play* (2020), pp. 572–584, doi: 10.1145/3410404.3414233. B.2

[178] WAMBECKE, J., VERGNE, R., BONNEAU, G.-P., AND THOLLOT, J. Automatic lighting design from photographic rules. In *Proc. Eurographics Workshop on Intelligent Cinematography and Editing* (2016), pp. 1–8, doi: 10.2312/wiced.20161094. A.4

[179] WARE, C. Quantitative Texton Sequences for Legible Bivariate Maps. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1523–1530, doi: 10.1109/TVCG.2009.175. 2.2, 2.10

[180] WEAVER, C. Building Highly-Coordinated Visualizations in Improvise. In *Proc. IEEE Symposium on Information Visualization* (2004), pp. 159–166, doi: 10.1109/INFVIS.2004.12. 2.2, 2.4

[181] WEISKOPF, D., AND HEINRICH, J. Continuous Parallel Coordinates. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1531–1538, doi: 10.1109/TVCG.2009.131. C.2.1

[182] WILKINSON, L. *The Grammar of Graphics*. Springer, 2005. A.2

[183] WILLIAMS, L. Casting Curved Shadows on Curved Surfaces. *ACM SIGGRAPH Computer Graphic 12*, 3 (1978), 270–274, doi: 10.1145/965139.807402. B.3.1

[184] WOODRUFF, A., LANDAY, J., AND STONEBRAKER, M. Constant Density Visualizations of Non-uniform Distributions of Data. In *Proc. ACM UIST* (1998), pp. 19–28, doi: 10.1145/288392.288397. A.2

[185] YANG, J. C., HENSLEY, J., GRÜN, H., AND THIBIEROZ, N. Real-Time Concurrent Linked List Construction on the GPU. *Computer Graphics Forum 29*, 4 (2010), 1297–1304, doi: 10.1111/j.1467-8659.2010.01725.x. C.2.2, C.3.1

[186] ZACKS, J., LEVY, E., TVERSKY, B., AND SCHIANO, D. Reading Bar Graphs: Effects of Extraneous Depth Cues and Graphical Context. *Journal of Experimental Psychology: Applied 4*, 2 (1998), 119–138, doi: 10.1037/1076-898X.4.2.119. 2.4, 2.30

[187] ZENG, Q., ZHAO, Y., WANG, Y., ZHANG, J., CAO, Y., TU, C., VIOLA, I., AND WANG, Y. Data-driven Colormap Adjustment for Exploring Spatial Variations in Scalar Fields. *IEEE Transactions on Visualization and Computer Graphics Pre-Print* (2021), doi: 10.1109/TVCG.2021.3109014. B.3.1

[188] ZHAO, J., LIU, X., GUO, C., QIAN, Z. C., AND CHEN, Y. V. Phoenixmap: An Abstract Approach to Visualize 2D Spatial Distributions. *IEEE Transactions on Visualization and Computer Graphics 27*, 3 (2021), 2000–2014, doi: 10.1109/TVCG.2019.2945960. B.3.2

[189] ZHAO, S., MCGUFFIN, M., AND CHIGNELL, M. Elastic Hierarchies: Combining Treemaps and Node-Link Diagrams. In *Proc. IEEE Symposium on Information Visualization* (2005), pp. 57–64, doi: 10.1109/INFVIS.2005.1532129. 2.2, 2.7

[190] ZHUKOV, S., IONES, A., AND KRONIN, G. An Ambient Light Illumination Model. In *Proc. Rendering Techniques* (1998), pp. 45–55, doi: 10.1007/978-3-7091-6453-2_5. A.3.2, B.3.1

[191] ZINSMAIER, M., BRANDES, U., DEUSSEN, O., AND STROBELT, H. Interactive Level-of-Detail Rendering of Large Graphs. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), 2486–2495, doi: 10.1109/TVCG.2012.238. A.2