THE VISUAL UNCERTAINTY PARADIGM FOR CONTROLLING SCREEN-SPACE INFORMATION IN VISUALIZATION

by

Aritra Dasgupta

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2012

Approved by:

____

Dr. Robert Kosara

____

Dr. William Ribarsky

____

Dr. Jing Yang

____

Mr. Shawn Bohn

____

Dr. Kexin Zhao

ABSTRACT

ARITRA DASGUPTA. The visual uncertainty paradigm for controlling screen-space information in visualization. (Under the direction of DR. ROBERT KOSARA)

The information visualization pipeline serves as a lossy communication channel for presentation of data on a screen-space of limited resolution. The lossy communication is not just a machine-only phenomenon due to information loss caused by translation of data, but also a reflection of the degree to which the human user can comprehend visual information. The common entity in both aspects is the uncertainty associated with the visual representation. However, in the current linear model of the visualization pipeline, visual representation is mostly considered as the ends rather than the means for facilitating the analysis process. While the perceptual side of visualization is also being studied, little attention is paid to the way the visualization appears on the display. Thus, we believe there is a need to study the appearance of the visualization on a limited-resolution screen in order to understand its own properties and how they influence the way they represent the data.

I argue that the visual uncertainty paradigm for controlling screen-space information will enable us in achieving user-centric optimization of a visualization in different application scenarios. Conceptualization of visual uncertainty enables us to integrate the encoding and decoding aspects of visual representation into a holistic framework facilitating the definition of metrics that serve as a bridge between the last stages of the visualization pipeline and the user's perceptual system. The goal of this dissertation is three-fold: i) conceptualize a visual uncertainty taxonomy in the context of pixel-based, multi-dimensional visualization techniques that helps systematic definition of screen-space metrics, ii) apply the taxonomy for identifying sources of useful visual uncertainty that helps in protecting privacy of sensitive data and also for identifying the types of uncertainty that can be reduced through interaction techniques, and iii) application of the metrics for designing information-assisted models that help in visualization of high-dimensional, temporal data.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Information visualization is that sub field of exploratory data analysis that communicates the hidden patterns in the data in a visually perceptible way. Often analysis questions are not known by the analysts apriori, but evolve in the course of their interactions with the data through its visual representation. As Donald Rumsfeld famously said:

> “There are known knowns; there are things we know that we know. There are known unknowns; that is to say there are things that we now know we don’t know. But there are also unknown unknowns there are things we do not know, we don’t know.”

Known unknowns and *unknown unknowns* also exist in information visualization and visual analytics. In fact, the visual analytics mantra of *detecting the expected, discovering the unexpected* [108] stems from the philosophy of there being both known and unknown factors that analysts encounter during their exploratory data analysis process. While the manifestation of unknowns is a direct corollary of the subjective nature of exploratory analysis, the visualization process itself accentuates the number of unknowns. This is primarily caused by two characteristics of the visualization process. Firstly, despite the increase in quality and resolution of computer displays, visualization still works in a space with a limited number of discrete pixels. Secondly, our lack of understanding of how the our perceptual system works, constrains us from designing effective ways of overcoming the limitations of human perception in recognizing visual structures on a limited resolution screen space. As a result, as the data gets progressively transformed, through the visual mapping on to the screen and through visual communication in the human mind, the number of unknowns increase. Examples of such unknowns in the screen-space include overlapping data points, hidden patterns due to choice of scale, complexity owing to loss of fidelity caused by lower dimensional projections of high-dimensional data, etc.

This necessitates the study of the appearance of a visualization on a limited resolution screen. Most of the current visualization models, however, treat visual representation as the end product in the visualization pipeline, even though it is clearly a central part of the analysis process, being the interface between the data and the human perceptual system [53]. To integrate the machine-side and human side of the visualization process, my collaborators and I (henceforth referred to as *we*) focus on the path of the data transformation, that is the visualization pipeline, for better understanding how we tackle the various unknowns [40]. The over-arching goal of my thesis is to develop a conceptual framework for studying the causes and effects of information loss that influence both the visual representation on screen and the visual communication process within the human mind. In the ensuing discussion, I have outlined the key elements of this framework and their functions.

1.1 Visualization Pipeline as a Communication Channel

Shannon had defined information as a measure of the decrease of uncertainty for the receiver of a message [100]. If visualization is viewed as a communication channel from the data space to the perceptual and cognitive mental space of the user [94], it is important to trace the uncertainty along different stages of the pipeline, so that the information com-

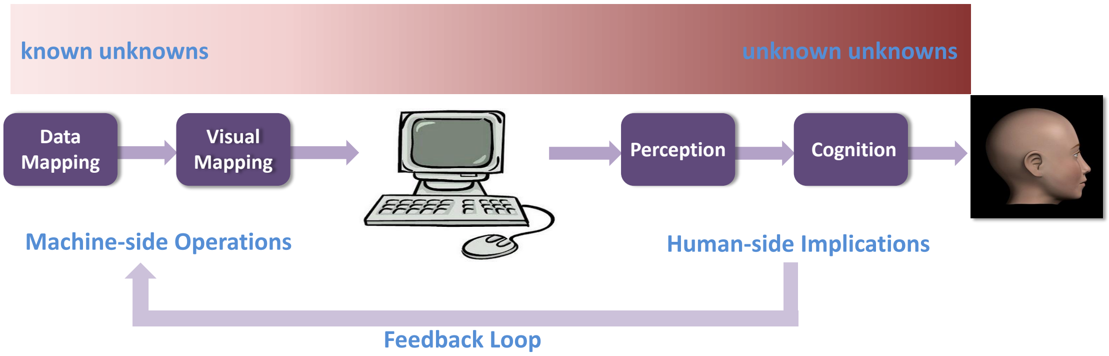


Figure 1: The conventional visualization pipeline augmented with a feedback loop from the human-side to the machine-side for making visual representations better informed about perceptual and cognitive implications and reduce the number of *unknown unknowns* from a visualization designer's point-of-view.

municated to the user can be optimized. Like a communication channel, visualization is also associated with the encoding and decoding of information. In Figure 1 we have augmented the conventional visualization pipeline [29] with two additional entities: the stages of perception and cognition to reflect the decoding steps and the feedback loop which is required to inform visual representations about the perceptual and cognitive implications. The feedback loop is informed about visual uncertainty [34] and can be used to measure visual representations according to he different levels of uncertainty in the screen-space.

Visual uncertainty takes into account the known and unknown unknowns that stem from the visualization process itself, rather than being present in the data. Conceptualization of the pipeline in terms of uncertainty analysis helps us in devising methods and metrics for achieving controlled information loss. Visualization of large, high-dimensional data almost certainty involves information loss due to the disparity between number of dimensions and data points and the limited number of screen pixels. In most applications, this loss is unintended [130], as there is degradation of data fidelity and visual quality. In some applications, however, where sensitive data is involved, some loss can be useful to hide certain attributes or values of the data. In both cases, designing effective visual representations, that capture the salient properties of the data and satisfy perceptual design principles at the same time, remains a significant challenge.

Measures about visual uncertainty can be used to optimize the visual representation. The data is followed through the pipeline, the different characteristics of end product is measured and fed back for modifying the earlier stages. This is similar to most engineering systems where the feedback is used to adjust the output. Our augmented visualization pipeline is similar in principle, to the one proposed by Van Wijk [115]. The important distinction is the characteristic of the feedback loop. In our case the feedback loop is used by the visualization designer to build better visual representations through controlled rendering and informed interaction design. In Van Wijk's model, the feedback loop is used by the analyst, based on his perception of the data, to build different representations of the data. Although both are manipulating visual representations, our model uses quantifiable means based on causes and effects of visual uncertainty, and therefore can be modelled and generalized as the basic principles of uncertainty are not affected by the subjectiveness of human judgement.

1.2 The Visual Uncertainty Paradigm

Visualization researchers have so far focussed on modelling uncertainty present within the data, while the same that stems visualization process itself, has not received much attention. We define *visual uncertainty* as the uncertainty that is associated with a visualization during encoding (in the screen-space) and decoding of information (in the mental space of the user). The concept of visual uncertainty thus subsumes known and unknown un-

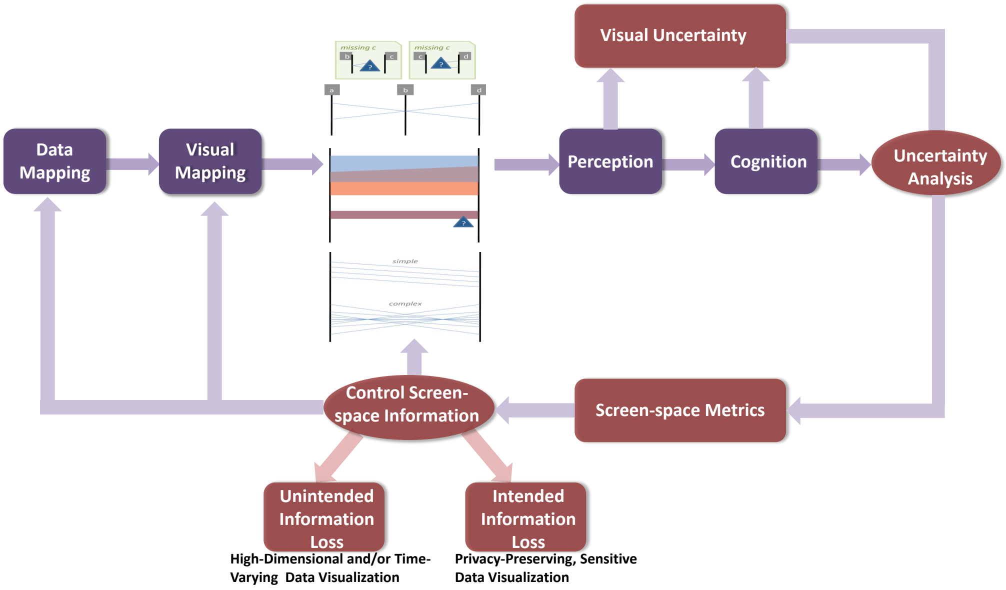

Visual Uncertainty
missing c
b
c
missing c
c
d
?
a
b
d
Data Mapping
Visual Mapping
Perception
Cognition
Uncertainty Analysis
simple
complex
Control Screen-space Information
Screen-space Metrics
Unintended Information Loss
Intended Information Loss
High-Dimensional and/or Time-Varying Data Visualization
Privacy-Preserving, Sensitive Data Visualization

Figure 2: The visual uncertainty paradigm for devising screen-space metrics that help control the visual representation, or the screen-space information. Potential applications include scenarios that involve intended information loss like in privacy-preserving data analysis and also unintended information loss like in high-dimensional, time-varying data analysis.

knowns and helps in addressing the information loss concerns associated with both the representation and communication of the data. We adopt the visual uncertainty paradigm for conceptualizing and quantifying the interplay between the different data-space and screen-space parameters in a visualization. This is done by systematically defining screen-space metrics (Figure 2) that quantify the different causes and of uncertainty at the different stages. The augmented pipeline illustrates that analysis of uncertainty in conjunction with the screen-space metric provide us with a closed loop. This in turn, allows us to control complex visual representations in cases where there is inherent information loss, like in high-dimensional data visualization and privacy-preserving data visualization.

My first contribution is a taxonomy of visual uncertainty for multidimensional visualization techniques. Uncertainty in the data space has been a widely researched area, but the uncertainty that is a product of the visualization process itself, has not been studied in detail so far. To fill this gap, we have categorized the sources, causes, and effects of encoding and decoding uncertainty, that are introduced at the different stages of the visualization pipeline, in the form of a taxonomy. We first propose a taxonomy of visual uncertainty for parallel coordinates (Chapter 3), which is an effective and widely used multidimensional visualization technique (Figure 3). The taxonomy is generalizable for those multidimensional representations that use position as one of the main visual variables [11] for encoding purposes.

By applying the taxonomy we systematically identify, analyze, and quantify those visual structures that reflect statistical properties of the data, and also those which inhibit the user perception like over-plotting, clutter, etc., with the help of screen-space metrics.

1.3 Controlling Screen-Space Information

Deconstructing a visualization in terms of its smallest indivisible components, that is the visual structures, enables us to control the screen-space information and aid the complex visual search process [27] by reducing the number of unknowns. As part of my thesis, I focus on two important use cases of this approach facilitated by the visual uncertainty paradigm, modelling intended and unintended information loss:

Unintended Information Loss: Unintended information loss is a common phenomenon in visualization. Especially for data involving the temporal dimension in addition to the large number of data dimensions, there are multiple challenges involved: selection and ordering of variables, quantifying and conveying the salient features features among different combinations of variables, and keeping track of changes over time. The *second contribution*

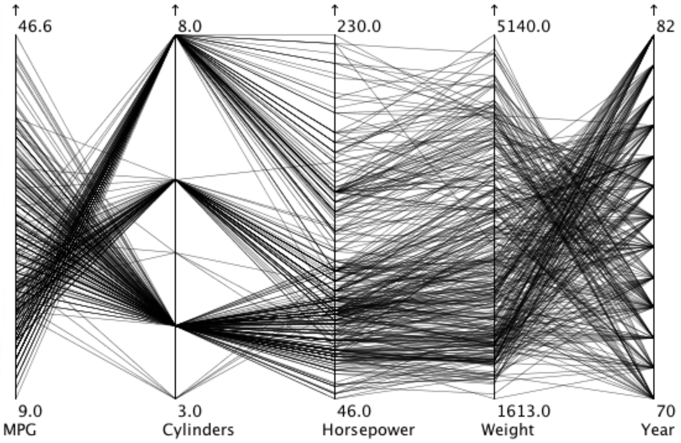


Figure 3: Multi-dimensional representation of the *Cars* dataset [52] using parallel coordinates, where each vertical line represents a data dimension and the poly-lines connecting them represent the records.

of my thesis is to devise screen-space metrics (Chapter 4) for quantifying various types of visual uncertainty, that can be subsequently used for designing an information-assisted model for visualizing high-dimensional, temporal data (Chapter 6). We argue that perceptually motivated quantitative metrics that define visual patterns and can be used as part of the feedback loop as shown in Figure 1 to optimize a visual representation and design direct manipulation based interaction techniques. Through our information-assisted model, we look at quantifying the important features in the data by using the metrics and communicating the trends through an interactive visualization system. I present a discussion of the applications of the system through a case study using simulation data from bioremediation experiments for carbon sequestration in soil sub-surfaces.

Intended Information Loss: Information loss is not always unwanted in a visualization. My third contribution is to demonstrate that controlled information loss helps in visualization of sensitive data. By exploiting the inherent information loss in visualizations we building models and techniques for privacy-preserving visualization (Chapter 5). Visualization techniques currently have an underlying assumption that there is unrestricted access to data. In reality, access to data in many cases is restricted to protect sensitive information from being leaked. Researchers in the field of data mining have proposed different techniques over the years for privacy-preserving data publishing and subsequent mining techniques on such sanitized data. A well-known drawback in these methods is that for even a small guarantee of privacy, the utility of the datasets is greatly reduced.

To fill this gap, we propose privacy-preserving visualization; instead of publishing the data or just the analysis results, a privacy-preserving visualization tool provides an interactive interface to both the data owner and outside users. The data owner can customize the tool to choose different reordered configurations of the data he wants to show to analysts without sacrificing privacy. Outside users cannot directly access the data, but only visualize the patterns in the data through the tool. Different constraints are imposed by the technique to prevent them from breaching privacy through interaction. The data is sanitized on the fly, based on the user's screen resolution and other viewing parameters. We propose a privacy-preserving visualization model and technique by adapting metrics for privacy-preserving data mining [3] and analyzing probable attack scenarios.

In this case, we intentionally hide some information and our knowledge about visual uncertainty helps in balancing 'how much to hide' and 'how much to show' to the user, so that the utility of the visualization is preserved as much as possible, while at the same time sensitive information is not leaked. In the process we also devise metrics for measuring privacy and utility (section 5.6 of the privacy-preserving parallel coordinates and scatter plots, by applying the visual uncertainty taxonomy. A visual approach to preserving privacy is a novel approach as also the conceptualization of a framework for privacy-preserving visualization.

CHAPTER 2: RELATED WORK

This thesis introduces two core concepts, visual uncertainty and privacy-preserving visualization, and establishes the relationship between the two. While data uncertainty has been widely researched in the field of visualization and outside, visual uncertainty is a novel idea to the best of my knowledge. The same applies for the concept of privacy-preserving visualization: a visual approach to privacy-preservation is introduced by this thesis and enables visualizations to be aware of sensitive data, a a hitherto unexplored area of visualization research. Moreover, the screen-space metrics that are the vehicles of this framework, are also a relatively new way to deal with important use-cases like high-dimensional and time-varying data visualization, as compared to data-space metrics. In this section I discuss the related work in each of these areas and set up the discussion of my research in the context of existing work.

2.1 Uncertainty in Visualization

Most existing work in visualization relates to data-space uncertainty (e.g., [91, 104]) and uncertainty involving geometrical primitives, like isosurface rendering. Our conceptualization of visual uncertainty applies in case of abstract data where a spatial context is not given [110].

There exists a plethora of discussions in the literature on classifying and categorizing uncertainty, for instance, in statistical forecasting, risk analysis, philosophy and psychology.

For a high-level classification of uncertainty in parallel coordinates, we take into account the different perspectives provided on uncertainty by Milliken [86], Norvig [98]; and Klir and Wierman [72]. We also refer to the typology proposed by Thomson et al. [109] for relating to data-space uncertainty.

Two existing schemes of uncertainty are most relevant for deciding our top-level classification. One such categorization is to consider uncertainty in physical systems (physical uncertainty) and that in the human mind (perceptual uncertainty) separately. For example, in behavioral sciences and neuroscience, there is an assumption that the nervous system performs its own probabilistic estimation about events in an environment, resulting in perceived certainty or uncertainty. Such results usually differ from those obtained from measurement of the events in the environment. This is partially true in case of mixed-initiative visualization systems as the data is first processed in physical systems, on the machine side, after which the human side takes over. On the human side, we have to take perception into account as well; uncertainty due to perception has been discussed by Russell and Norvig [98]. Holzhüter et al. [61] describe uncertainty in visualization and differentiate between input and output uncertainty. Relating the information visualization pipeline to the communication channel as discussed previously, we choose encoding and decoding uncertainty to be the topmost classifying schemes, which are further elucidated in Section 3.

2.2 The Case for Metrics

Several authors have argued for the need for metrics in information visualization. Tufte [111] was among the first to propose visual quality metrics for static two-dimensional charts. Taking a cue from Tufte's work, Brath [21] proposed a set of metrics to assess the efficacy of static 3-D presentations, based on data density, dimensionality and occlusion. Miller et al. [85] have argued for the important role of metrics in predicting the success of visualization tools and their validation. A more concrete work and similar to the approach we follow is seen in Bertini et al.'s work on scatter plots [14], which employs a non uniform sampling technique to reduce clutter in 2D scatter plots. They further argue for the need of metrics [15] where they define different visual quality metrics and provide a research direction for each. The definition of the metrics is applicable to our work, and we conceive screen-space metrics as a diagnostic model for quantifying visual quality and feature-preserving visual structures.

2.2.1 Information-theoretic Measures in Visualization

In recent times, the use of information-theoretic measures to quantify the information-content of a visualization have been proposed by several researchers. Purchase et al. argue about conceptualizing the visualization pipeline as a lossy information channel and mentions that information-theoretic measures can be used to measure the loss [94]. Rundensteiner et al. propose some measures of data quality and abstraction quality to make the connection between data and the screen spaces [97]. Yang-Peláez and Flowers have demonstrated how information content in visualization can be quantified without taking semantics into account [125]. More recently, Chen and Jänicke have shown how different information-theoretic concepts like entropy and mutual information can be used at different stages of the visualization pipeline [28]. We adapt and apply some of these metrics for a user-centric optimization of the display in cases of privacy-preserving visualization [35] and high-dimensional, time-varying data visualization.

2.2.2 Screen-Space Metrics for High-Dimensional Data Visualization

A key aspect of the metrics we propose is that they form the building blocks of an information-assisted visualization model [27, 36], but based on screen-space metrics rather than information abstracted from data. Our goal is to convey the different visual structures that encode information to the user. One of the early instances of such work is *Scagnostics* (*Sca*tterplot Dia*gnostics*), proposed by Tukey et al. [112]. This work was extended by Wilkinson et al. with more detailed graph-theoretic measures [120] for detecting a variety of structural anomalies in a geometric graph representation of the scatter plot data. The resulting rating can be used to pick views that show particular structures that are of interest to the user.

Similarly, in Pixnostics [99] the authors use image- and data-analysis techniques in conjunction to rank the different lower-dimensional views of the dataset and present only the best to the user. The method creates lower-dimensional projections that provide maximum insight into the data and optimizes the parameter space for pixel-oriented visualizations.

In Pargnostics, we focus on parallel coordinates and, similar to Scagnostics and Pixnostics, aim to reduce the analyst's burden of searching for the optimum views of the data. The over-arching goal of Pargnostics is to play the role of a multi-dimensional detective [64], where we diagnose structures of interest on behalf of the user. In addition, the user is aided with an interactive interface to enhance the effect of helpful structures (e.g., convergence/divergence, parallelism) and reduce the same for structures that hinder his analysis process (e.g., large number of crossings, over-plotting).

There are some instances of image-space based metrics in the context of parallel coordinates [66, 65]. Tatu et al. propose similarity-based functions based on Hough Space transforms to find clusters [107]. They propose dimension reordering based on analysis functions on the resulting image of parallel coordinates. Johansson et al. [67] propose a screen-space metric based on distance transforms to estimate the visual quality of the abstracted dataset. Our approach is different in the sense that we attach semantics to the structures that can be seen on the screen and not only define the metrics qualitatively but also quantitatively.

Optimal order of dimensions is another challenge in parallel coordinates and this has been addressed in earlier work: Ankerst et al. [9] compute the degree of similarity among dimensions based on data-space metrics and cluster similar dimensions together. A somewhat similar idea is pursued by Yang et al. [123]: similar dimensions are clustered and used to create a lower-dimensional projection of the data. In Pargnostics we focus on dimension reordering as a tool for optimization based on a new set of screen-space metrics Section 6.3. We also employ optimization where the users can select views that suit their analysis by browsing through a rank-ordered view by features.

2.3 Privacy-Preserving Visualization

While the issue of privacy has not been studied in detail in visualization, many techniques for publication and analysis of de-identified sensitive data have been developed in the field of privacy-preserving data mining (PPDM) [3].

2.3.1 Model and Metrics

We adapt the well-known *k*-anonymity model with the help of screen-space metrics. The *k*-anonymity model [106, 114] focuses on making *k* records indistinguishable with respect to quasi-identifiers so that identification through linking is prevented. *k*-anonymity problem has been shown to be NP-hard [84] and therefore many approximation algorithms have been proposed [24, 1]. We use the *k*-member clustering algorithm proposed by Byun et al.[24]. While we adopt the overall algorithm proposed by Byun et al., we use a different criterion for seeding and a different cost function. We also adapt the algorithm to work individually for each axis pair, rather than across all dimensions at once.

k-anonymity does not guard against homogenity attack due to the lack of diversity in sensitive attributes: even if records are indistinguishable with respect to quasi-identifiers, the malicious user can use his background knowledge and breach privacy. To address this, Machanavajjhala et al. proposed *l*-diversity [82] which ensures each group has at least *l* different values for the sensitive attribute. For example, in a disease dataset if a particular quasi-identifier group for people belonging to age above 50 only has the value *cancer*, then a user who knows the age of a person can correctly guess the value. We use this *l*-diversity property as a basis for constraining user interaction and showing at least *l* different values for the sensitive attribute. I discuss the visual model of privacy and applicability of the metrics in Section 5.2 and the technique [39, 38] in Section 5.3. To the best of my knowledge, no previous work exists that proposes a privacy-preserving visualization technique and the only instance we are aware of uses graph-based abstraction of web data for privacy-preserving manifold visualization [127].

2.3.2 Privacy Vs Utility

While the trade-off between privacy and utility is very much an open research area in PPDM, several metrics have been proposed to quantify privacy and utility individually. A comprehensive survey have been done by Bertino et al. [18] where the different metrics have been categorized and described. Entropy as a privacy metric was first proposed by Aggrawal et al. [2] and was developed further by others [17, 19]. Utility has been measured in terms of data quality and clustering quality and also with respect to preservation of patterns with respect to specific data mining techniques. We use the visual uncertainty taxonomy for systematically defining metrics [35] for measuring various aspects of privacy and utility Section 5.6.

2.4 Multivariate, temporal data visualization

We discuss the related work in the context of analytical abstractions for multivariate temporal data analysis and the different variants of parallel coordinates for this purpose.

2.4.1 Analytical Abstraction for Multivariate Temporal Data Visualization

Large multivariate temporal datasets require effective analytical abstractions that can capture the dynamic relationship among multiple variables. Integrating computational and visual aspects [59] by using clustering based visualization techniques for modelling similarity-based temporal behavior have been proposed, where parallel coordinates have been applied for visualizing the results of clustering. Pre-processing of data for extracting trend sequences and subsequent visualization of those temporal trends through parallel coordinates have also been used [74]. Functional temporal plots for visualizing changes in correlation in a matrix layout have been used in the context of gene-sequence modelling [83]. These analytical abstraction methods focus on extracting information from the data and then using visualization tools for communicating that information to the user. While the basic goal of our approach is the same, the philosophy behind our approach is different. In our approach, rather than considering parallel coordinates as an end-product for visualizing data-driven metrics, it itself is used to drive the analysis process.

Our focus is reducing the visual uncertainty by using metrics that describe the salient visual structures, so that there is a direct correspondence between the behavior of the metrics and structural change on screen. The system presented by Glatter et al. [56] follows a similar principle. In that case, a domain scientist specifies uncertain temporal patterns using a description language, and temporal evolutions and multivariate connections can be formulated as queries using this language. Extracting importance based relationship using information theoretic metrics to describe the visual structures [116] is another example of such an approach.

2.4.2 Temporal Parallel Coordinates

Several parallel coordinates variants have been proposed for scientific data applications. One of the examples is the application for hurricane data analysis [105] where statistical properties are mapped on to the parallel coordinates axes. In the interface proposed by Akiba and Ma[4], multivariate connections can be brushed using a parallel coordinates interface, which in turn is linked to time histograms and a direct volume rendering of selected attributes. But as observed in the case of tiled parallel coordinated display and min-max plots [25] applied to time-varying EEG data, the overall temporal distribution is not conveyed in this approach.

Johannson et al. use depth cues and variation of opacity to show temporal properties in parallel coordinates [68]. This approach suffers from clutter in case of large number of time steps and data-points. Our approach, is to look at the different recognizable visual features

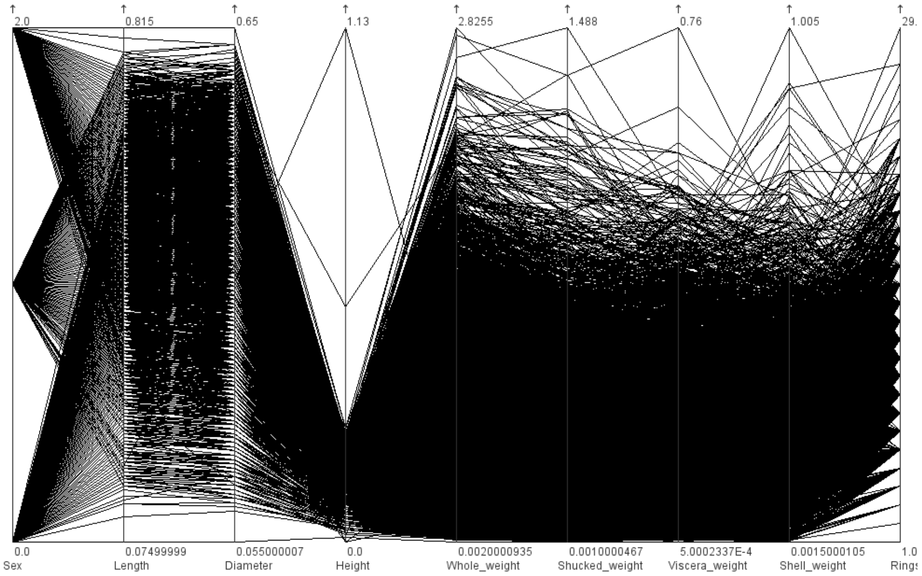


Figure 4: Parallel coordinates representation of the *Abalone* dataset [52] where a high-degree of over plotting and line crossings leading to clutter make it impossible to find useful patterns.

and use appropriate metrics to convey them. Blaas et al. [20] use data quantization and compression to handle large number of data points in the context of parallel coordinates. While this works well at the overview level, detailed exploration of features is difficult using this approach. Frequency-based representations like use of angular histograms [55] is similar to our approach.

In our work [41], we apply some of the visualization design principles for simulation data as outlined by Doleisch et al. [43], with focus on the exploration and analysis aspects. We have also incorporated some of the visualization strategies for dealing with time-series data [87], where we address the key issues of finding the temporal patterns and understanding the change in behavior over time through linking of different views of the data (Section 6.4).

CHAPTER 3: VISUAL UNCERTAINTY

Uncertainty is a twofold problem in visualization. On one hand, it is important for visualization to convey uncertainty in the data to the users, giving rise to the quest for effective means to measure and visually depict uncertainty [69]. On the other hand, the visualization process itself will introduce uncertainty. The former is primarily concerned with uncertainty in the data space, while the latter has to address sources of uncertainty in visual mapping, rendering, displaying, viewing, perception, understanding, and reasoning. While much of the existing work in the visualization literature focuses on data uncertainty [122, 91], discussions on uncertainty stemming from the visualization process itself are still limited. In scientific visualization, there is not always a clear boundary between data uncertainty and visual uncertainty, since the visualization process often involves the manipulation of geometric primitives (e.g., errors in isosurface extraction [95, 78] or in particle tracing [80]). Even when such geometric abstraction is considered as part of visual uncertainty, it represents only one specific type of uncertainty caused by the visualization itself. The aim of this work is to highlight the fact that there are many other types of uncertainty sources in the visualization process.

3.1 Taxonomy of Visual Uncertainty for Parallel Coordinates

We adopt a case-based research methodology by focusing on a specific class of non-spatial data visualization, namely *parallel coordinates visualization*, which is a powerful

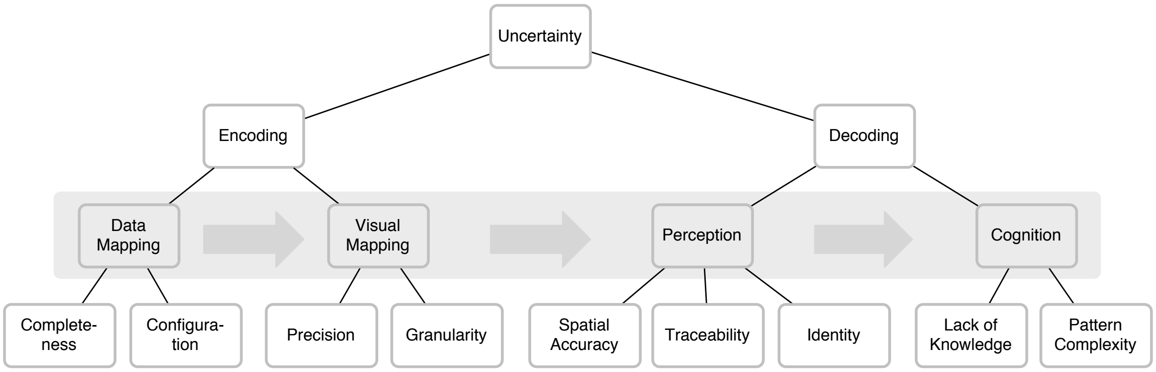


Figure 5: The proposed taxonomy of visual uncertainty in parallel coordinates. The shaded area indicates the level at which it maps to the stages in typical visualization pipelines.

tool for visualizing and analyzing multi-dimensional data. Almost everyone who has used parallel coordinates has seen the dreaded black screen which is composed of over-plotted lines and conveys a high level of uncertainty but a very limited amount of useful information (Figure 4). In practice, many visualizations contain more subtle forms of uncertainty. For example, axes with few values create focal points where many lines meet, but it is uncertain how lines continue to the next axis; over-plotting of lines that are very close together makes it impossible to tell exactly how many points are in a particular location; the inherent resolution of the pixel grid limits the perceivable resolution of the data; etc. By focusing on a specific class of visualization, we are able to conduct a detailed analysis of a manageable set of sources and effects of uncertainty and their relationships. We believe that this methodology and the major findings of this work can also be applied to other classes of visual representations.

Our taxonomy (Figure 5) separates the causes of uncertainty into two main groups: encoding (Section 3.2) and decoding (Section 3.3). The typical way of looking at uncertainty is from a decoding perspective, which includes our perceptual and cognitive processes when working with a visualization. Uncertainty is also introduced on the encoding side, however, through transformations of the data, mapping to the pixel grid, or selections of data and axes.

Uncertainty is usually the result of a number of causes, but we have attempted to narrow down the main reasons for uncertainty in specific cases. Most real-world scenarios will consist of combinations of these cases, and even within the taxonomy there is some overlap between some of the higher levels and the specific examples. As a working definition, we adopt the definition of Douglas Hubbard [63], which describes uncertainty as *the lack of certainty, a state of having limited knowledge where it is impossible to exactly describe existing state or future outcome, more than one possible outcome*.

The third level of the taxonomy coincides with the stages found in visualization pipeline models like Chi's [29]: data mapping and visual mapping. We add two stages on the human side of the pipeline, perception and cognition; while they are not very clearly delineated, we find them useful to structure the lowest level of the taxonomy.

3.2 Encoding Uncertainty

As data moves through the visualization pipeline, it gets transformed and mapped to visual coordinates and shapes. The encoding side of our taxonomy includes all the stages from data access to rendering the visualization on screen. Data acquisition and any uncertainty inherent in it is outside the scope of this work.

3.2.1 Data Mapping

In the first stage of the visualization pipeline, the user selects the data points and dimensions that are to be mapped onto the screen. In addition to the selection of the dimensions,

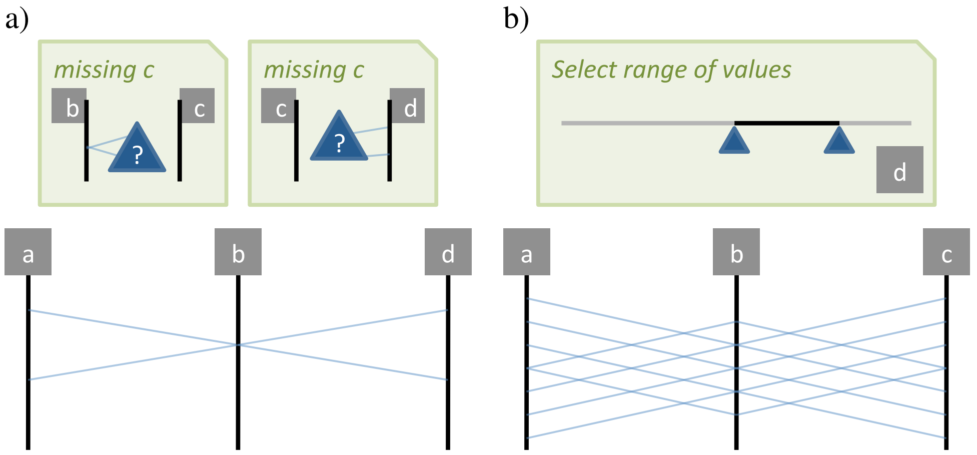


Figure 6: Completeness: Choosing not to include an entire axis (a) or single values (b) prevents the user from seeing some of the data, causing uncertainty about it.

their ordering is also determined, which is important for the patterns that will be visible once the visualization is drawn onto the screen. In contrast to data-space uncertainty, this process is entirely driven by the user, who picks which elements to show (usually in response to what is currently shown on the screen). This process seems benign and simple, but there are many possible configurations, many of which hide potentially interesting parts of the data.

Completeness: While parallel coordinates can show many dimensions at once, many high-dimensional datasets are still impractical to show all at once, or the user may choose to show a smaller number to gain more space per dimension. By leaving out dimensions, potentially interesting structures are not shown on screen, causing uncertainty about the complete set of patterns in the data (Figure 6a). It is also possible to filter the data on a dimension that is not part of the visualization (Figure 6b). The most common case for doing this is when there is a time dimension in the data, in which case the visualization shows the data for only one particular time step. When not all records are shown, patterns can be hidden that would be apparent if all the data was there, resulting in further uncertainty.

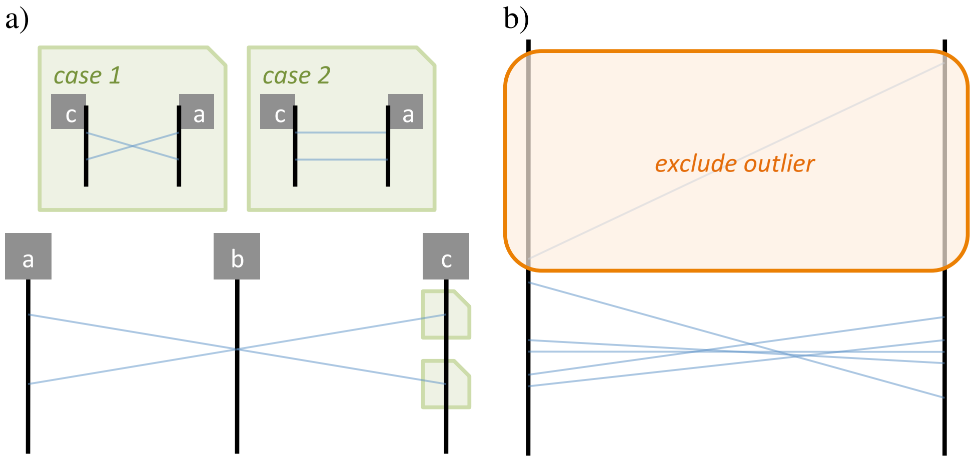


Figure 7: Configuration: a) Patterns can be missed when not all possible pairs of axes are represented; b) leaving out individual data values prevents the user from seeing parts of the data;.

Configuration: Even if all the axes are shown, their order is crucial to see patterns: most patterns are only visible between directly adjacent axes. Not only is it typically not feasible to try out all possible axis orderings, it is also uncommon to show the same axes several times in the same visualization [118]. The wrong choice of axis ordering can thus hide important patterns without the user being able to find out what he or she is missing, leading to uncertainty (Figure 7a). A common interaction in parallel coordinates is the exclusion of outliers on an axis, typically to give the remaining data more space (Figure 7b). In contrast to the completeness case above, the missing data is chosen by visual criteria, and is typically still shown on the other dimensions (and as a line that is leaving the screen). The exact values of those outliers are lost, however.

3.2.2 Visual Mapping

When the data is drawn onto the screen in the visual mapping and rendering stage (which we treat as one step), its uncertainty increases due to the limited resolution of the pixel grid. The application of information theoretic metrics in quantifying the screen-space artifacts has been discussed by Chen and Jänicke [28]. In parallel coordinates, a variety of artifacts are produced both on the axes and between them. While these also cause issues on the perception side of the taxonomy, there are really two separate phenomena at play here that need to be distinguished. The visual mapping side is also easier to assess due to its mechanical nature than the much more complex perception side.

Precision: The limited number of pixels on a display causes the locations of the data points to be quantized into a relatively small number of distinct values. In most real datasets, many data points end up getting mapped to the same pixel locations, and thus can no longer be differentiated. The information lost at this stage leads to uncertainty about the precise values of the data points (Figure 8a). When transparency is used, the colors of lines also mix, making it difficult to tell how many and which values are present. This is especially true when color is also used, such as for a gradient on one axis to more easily spot correlations. Even given perfect color perception, it is impossible to decode the resulting colors due to the limited resolution of the color values represented on the screen, and the resulting quantization of the colors (Figure 8b).

Granularity: Clustering naturally introduces uncertainty into the data, by reducing the number of values and representing them only as cluster boundaries or centroids and sizes. We are interested in the visual appearance of clusters between axes when they can be shown as polygons [89, 38]. Just as in data space, the visual clusters hide the individual lines, thus removing information about the distribution of lines within the cluster, and even the number of lines in each cluster (Figure 9a). A similar issue occurs on the axes, where the locations of the points are no longer known, even when the cluster boundaries are defined by the maximum and minimum axial values of points in cluster (Figure 9b). In that case,

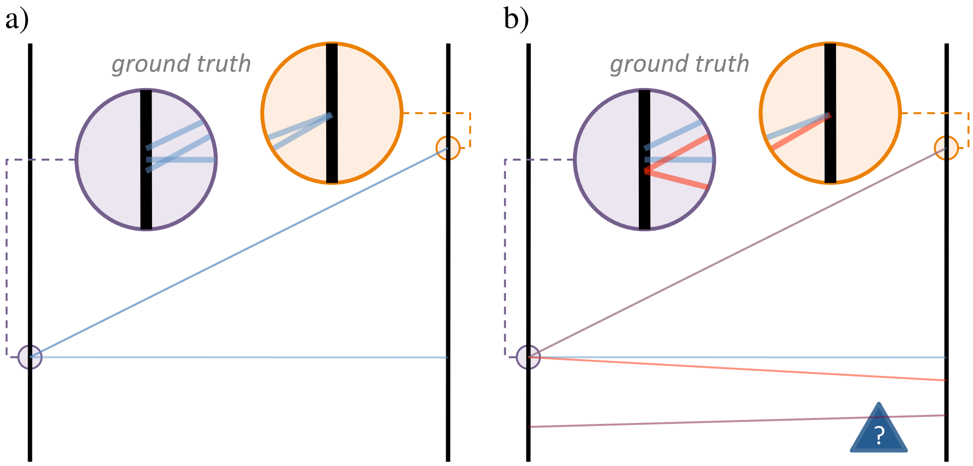


Figure 8: Precision: a) Pixel binning leads to a loss in precision, which makes it impossible to read values precisely; b) the colors of lines drawn over each other make it difficult to see brushing and the precise number of lines (when transparency is used).

it is not known whether the corners defining the cluster belong to the same data point or to different ones; several combinations are possible that are all equally likely (one line can run along the boundary or the boundary can consist of two distinct data points, for both boundaries independently).

3.3 Decoding Uncertainty

Once the information is encoded and the visualization rendered to the screen, the perceptual and cognitive processes of the user take over in interpreting that information. Decoding uncertainty occurs in the perception and cognition stages of our pipeline. We consider a source of uncertainty to be in the decoding branch only if the information concerned is fully encoded in the visualization. Without the information having been encoded first, it cannot be decoded, thus we give priority to the encoding stage. Analysis of decoding uncertainty enables us to evaluate a visualization technique by asking questions such as: is it perceptually confusing, does it incur a high level of cognitive load for reasoning, or is it only suitable for expert users who know how to interpret the visual representation?

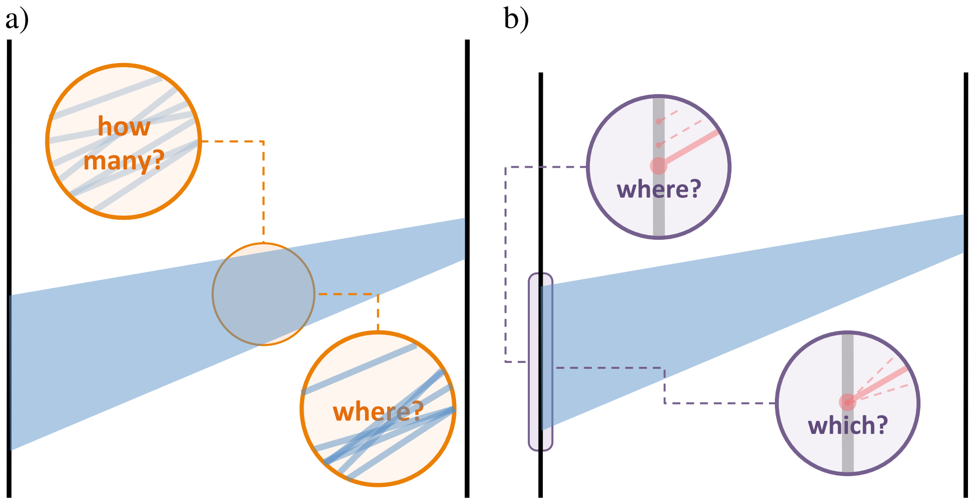


Figure 9: Granularity: a) Clustering of values hides information about the internal structure of the cluster and potentially the number of items in each cluster; b) the same is true for the internal structure of the cluster and the actual locations of the original data points.

3.3.1 Perception

In this section, we consider visual uncertainty resulting from the limits of the human vision system. Higher-level processes such as knowledge and the ability to perceive and recognize patterns are discussed in Section 3.3.2 on cognition.

Spatial Accuracy: The lack of knowledge about the exact spatial location of terminators (such as where records within a cluster are located or where the attributes are on an axis) or other geometric features (such as where two lines cross each other) causes uncertainty about the precise data represented. Perceptual accuracy concerns whether the user can differentiate visual objects from available information such as locations and colors. Sometimes, although the information is theoretically there on the screen, it can still be perceptually very difficult to perceive such information due to either the discriminative limit of the human vision system or perceptual illusion. This issue is different from (though related to) the missing information or lack of precision as discussed in Section 3.2.2. In the latter case

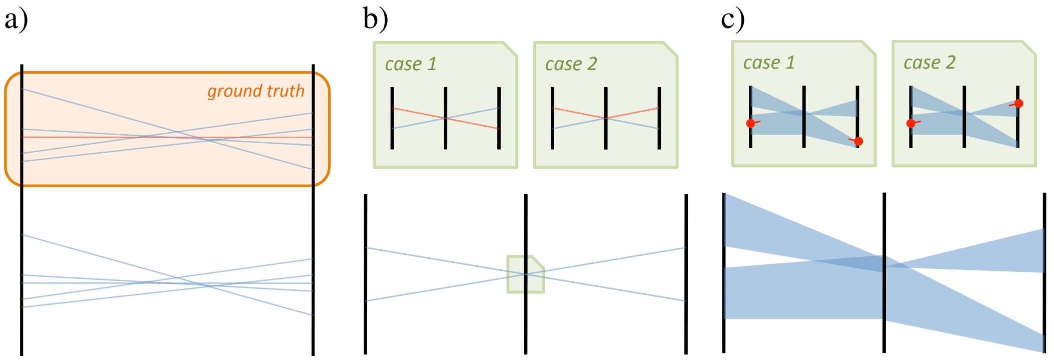


Figure 10: Traceability: a) Single lines are easily hidden among others, leading to uncertainty about the exact number of lines and fine details in the data; b) lines meeting in single points, or in very small neighborhoods, on axes cause ambiguity about the multi-dimensional nature of the data; c) clusters show similar issues and are difficult to trace across multiple dimensions.

uncertainty was theoretically there at the end of visual encoding, while former was caused by the human vision system.

Traceability: When there are many lines between adjacent axes, it becomes difficult to see individual ones in the resulting clutter. This is particularly problematic when most of the lines are almost parallel, but the ones that differ (which are often of particular interest) are hidden among or behind them (Figure 10a). Even if lines differ in the pixels at their end points, small angles between lines can cause confusion when looking at the space between axes. When lines converge onto the same pixel (or pixels that are very close together), it can become impossible to tell which line continues in which direction after that point (Figure 10b). While this can be a precision issue when the values are actually different, it becomes a pure traceability problem when the underlying data values are identical, and thus would never be mapped onto different pixels, no matter the resolution of the display. This is a common problem when categorical data is present in datasets visualized using parallel coordinates. A similar issue exists also for clusters, whose structure can be confusing due to splits and overlaps on and near axes (Figure 10c).

The common solution to the problem is interaction, which allows the user to highlight a particular record or cluster, but this is not always practical and certainly does not provide as much information as directly showing it. Users are also not always aware of traceability issues and simply fail to see subtle patterns or outliers.

Identity: Identity uncertainty is usually caused by many lines or clusters crossing, sometimes at low angles, making it difficult to uniquely identify a line or cluster [62]. A single line can easily be hidden behind many other lines that form a solid, or almost solid, structure (Figure 11a). This case is distinct from the traceability case because it may not be apparent on the axes that the line is even there; a line that is not known to be there cannot be traced by the user. Only hints between the axes can show that this data value even exists. Overlapping clusters create similar issues, with the additional problem that they can make the user assume the existence of clusters that are not actually there. The lines created by overlaps can be misinterpreted as distinct clusters, and even when not it is often impossible to tell how many clusters there are (Figure 11b). A related issue making it difficult to tell how many (and which) clusters exist is when colors of cluster mix. Does the mixed color present a distinct cluster of that color or the overlap between two clusters of other colors (Figure 11c)?

3.3.2 Cognition

Cognitive uncertainty is caused by difficulties in cognitive reasoning, such as confusion and misinterpretation. Milliken [86] classifies cognitive uncertainty into *state uncertainty*, *effect uncertainty* and *response uncertainty*. In data analysis, for example, state uncertainty

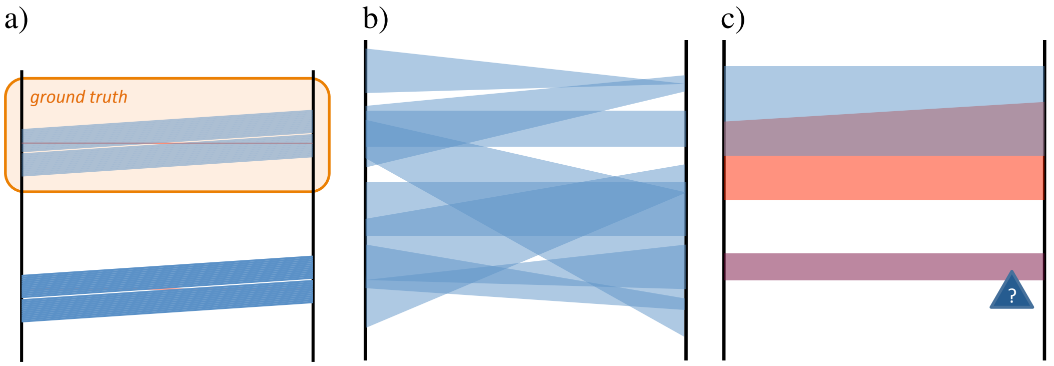


Figure 11: Identity: a) Color mixing leading to confusion among identity of lines; b) overlapping clusters leading to clutter; c) color mixing among clusters lead to confusion among clusters

may describe the lack of certainty about the data and information given. In the visualization context we term this category *lack of knowledge*. Effect uncertainty may describe the lack of certainty about what the information implies; response uncertainty may describe the lack of certainty about what action one should take (the latter two are outside the scope of this taxonomy).

Lack of Knowledge: Parallel coordinates require knowledge and experience to use for effective data analysis. Users who are unfamiliar with the way the technique depicts certain patterns may be unable to tell which pattern they are actually looking at (Figure 12a). Even when they are familiar with the technique, inconsistent axis scaling can mislead users. Parallel coordinates often scale every axis independently to make the most use of space, thus making direct comparison between them impossible, and shifting the locations of the zero on each axis. Patterns can be misinterpreted because of this (Figure 12b).

Pattern Complexity: Highly complex patterns in the visualization can lead to misinterpretations, even when they are correctly represented and readable on the perceptual level. While simple correlations, aggregation of values, etc., are easy to see, the superposition of

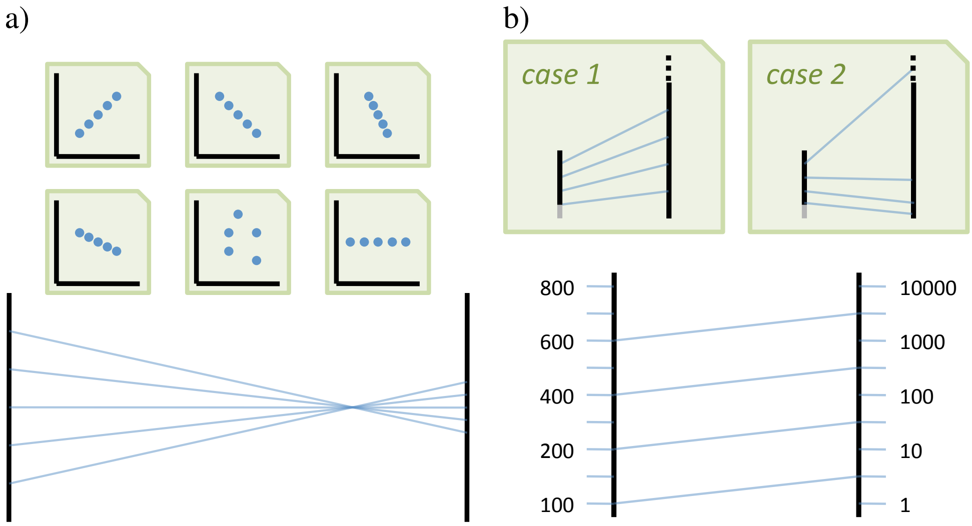


Figure 12: Lack of Knowledge: a) Not knowing how to read the sometimes complex patterns in parallel coordinates leads to uncertainty about the represented pattern; b) inconsistent axis scaling, in particular because of the different locations of the zero, can lead to issues in interpretation.

different patterns can lead the user to see one pattern but ignore the other (Figure 13).

3.4 Applying the Visual Uncertainty Taxonomy

In this section we analyze the existing research on parallel coordinates with respect to the taxonomy. Work on parallel coordinates has focused on two categories: qualitative tasks (clutter reduction, improving visual quality) and common analytical tasks and data operations (clustering, finding correlations, detecting outliers, privacy preservation). These tasks are based on the low-level analytical activities of a user [6] that are supported by parallel coordinates [7]. For the different uncertainty sources we analyze how these uncertainty sources reduce/enhance certain effects that are useful in some analysis scenarios. The discussions relating analytical tasks to visual uncertainty are summarized in Table 1 and described in detail in the following section.

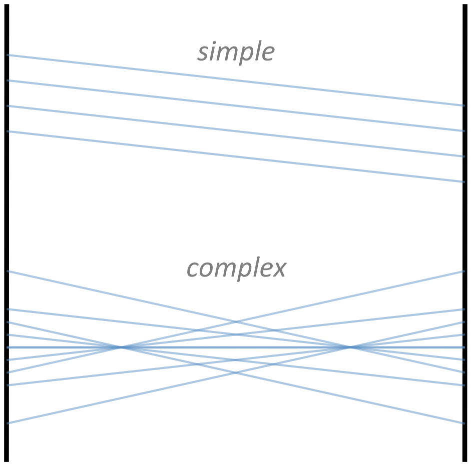


Figure 13: Pattern Complexity: More complex patterns in the visualization lead to more difficulty in reading and understanding the underlying data patterns.

3.4.1 Clutter

There are different definitions of clutter in the parallel coordinates literature. Peng et al. [92] define clutter as the relative number of outliers to the total number of data points and aim to have a configuration which optimized with respect to outliers. The authors use reordering technique to achieve that configuration. This technique addresses the uncertainty due to configuration of the visualization: while this type of uncertainty is reduced on one hand due to preserving outliers, particular selection and ordering of axis increases the same effect of uncertainty due to possible omission of salient patterns.

Identity: Some clutter reduction techniques aim to reduce the number of visual elements, at the visual mapping stage. In parallel coordinates, that means reducing or manipulating number of lines that connect the data points. Sampling Lens [48] is one such example where the data to be mapped on to the screen is abstracted based on density of the data points. Artero et al. [10] reduced non-important information in parallel coordinates based on the computed frequency and density plots from the original datasets. The screen space quality method [67] reduces clutter, while preserves the significant features in the original datasets at the same time, by filtering out data items based on distance transformation for data abstraction. These methods while reducing identity uncertainty, lead to lack of completeness in the visual representation.

Traceability: Another definition of clutter is according to Ellis and Dix [49], where clutter is attributed to large number of crossings and lines crossing at low angles In line-based parallel coordinates, clutter is caused by too many line crossings, several lines crossing at low angles and lots of lines converging or diverging from a small region on the axis. This relates to the uncertainty due to traceability between adjacent axes and across different axes, and also identity uncertainty on the axis. While in Pargnostics [37] the authors aim to retain data fidelity and reduce clutter through reordering-based optimization, in the previous case the authors reduce the number of visual elements, thereby leading to a completeness problem.

3.4.2 Clustering

Existing research on improving visual quality in parallel coordinates focuses on clustering [129, 54] in various forms.

Configuration: In both line-based and cluster-based parallel coordinates, binning helps in having pre-defined seeds for clustering. Binning can be either data-based or pixel-based. Pixel-based binning [89] helps in overcoming the problem due to high cardinality of a data-space, but leads to over-plotting. Cui et al. [33] have proposed metrics that measure the data quality. Pixel-binning therefore reduces configuration uncertainty. However, binning also leads to loss of precision and granularity uncertainty as many lines can end up on a single

Task	Data Cardinality	Data Dimensionality	Source	Intended Effect	Unintended Effect	Utility
Finding Correlations	Large		Large crossings	- Pattern Complexity	+ Identity	Inverse correlation
		Large	Axis selection	- Pattern Complexity	+ Configuration	Correlation between dimensions
Detecting Outlier	Large		Axis scaling	- Pattern Complexity	+ Loss of precision	Spotting anomalies in trend
Clustering	Large		Binning	- Configuration	+Precision	Seeds for clustering
	Small		Low crossing angles	- Pattern Complexity	+ Identity, Traceability	Clustering due to proximity, similarity
		Large	Axis selection	- Pattern Complexity	+ Configuration	Subspace clusters
Privacy-preservation	Any		Binning	+ Identity	N/A	Loss of precision and granularity
			Overlaps on the axis	+ Identity	+Pattern Complexity	Uncertainty in identifying individual values
			Cluster splits	+ Traceability	+Pattern Complexity	Uncertainty for traceability of sensitive clusters

Table 1: Connecting sources and effects of uncertainty to tasks and data properties (cardinality and dimensions). The positive sign indicates a particular effect of uncertainty is enhanced and negative sign implies the same is reduced. Usually, the intended effect is the reduction of a certain cause of uncertainty. In case of privacy, since increase of uncertainty is intentional, we consider the effect as being useful.

bin.

Low crossing angles help in the perception of proximity and similarity by inducing a Gestalt effect. For small number of data points, lines crossing at small angles generally mean lines are more or less parallel to each other, which indicates implicit clusters. In case of large number of data points, many lines crossing at low angles would tend to produce clutter. Clustering techniques in parallel coordinates aim to reduce the uncertainty related to data similarity and proximity and support analytical tasks of finding clusters within the data [10, 8]. Information loss is intended in these cases. However uncertainty can be introduced due to lack of granularity information. The techniques do not generally convey the number of records within a cluster.

Identity and Traceability: Zhou et al. [129] proposed geometry-based visual clustering to implicitly enhance the clustering in parallel coordinates by bundling the edges, and minimizing the edge curvatures and maximizing the parallelism of adjacent edges at the same time. Other than reducing clutter, they also achieve reduction of uncertainty through enhancing the perception of continuity by choosing curved edges instead of lines as the basic visual elements. This reduces traceability uncertainty by violating the gestalt law of continuity among visual structures. Further, clusters can be detected by superimposing semitransparent line segments on the screen to enhance important components [128] and thereby reducing identity uncertainty.

Wegman and Luo [119] also use transparency to identify regions of high over-plotting through their dense color. Holten et al. [60] have shown through user studies that improvements in visual enhancements do not always work well in practice. They have further argued for more formal evaluation measures for these techniques and we believe our definitions of visual uncertainty will help future approaches towards achieving a more quantitative basis for comparison.

3.4.3 Finding Correlations and Detecting Outliers

As pointed out in Table 1, line crossings, although lead to clutter in most cases, can be helpful in the case of a small number of data points, when large number of crossings at high angles is a useful representation for inverse correlations [37]. This enables the cognition of linear correlation, and thus reduces pattern complexity. For detecting outliers, normalization of the axes using non-linear scaling can be applied [7]. While this helps in reducing pattern complexity, there is significant loss in precision for the represented data.

3.4.4 Useful Uncertainty: Privacy

In case of privacy-preserving applications, contrary to the other categories mentioned above, certain effects of uncertainty are intentionally increased. For ensuring privacy data needs to be hidden, and in an interactive environment, there needs to be sufficient uncertainty to confuse the user so that he is not able to breach the intended privacy of the application. The uncertainty should, however be focused on the left part of the taxonomy tree, i.e, encoding uncertainty as increasing decoding uncertainty would degrade utility of the visualization to a much larger extent.

Loss of precision and granularity or lack of completeness are all related to information loss in visualization. While there has been sporadic mention of quantifying information loss [94, 130], we still lack a framework for describing it. In privacy-preserving visualization, a clustering technique based on screen-space metrics is used to set a lower bound on the number of records per cluster. By using pixel-based binning as a staring point of the clustering process, it exploits the inherent loss of precision to mask the real values of the records. Uncertainty is also increased by the unknown location of the records within the clusters,leading to granularity uncertainty, as we had shown earlier in Figure 9. Thus, encoding uncertainty here is caused by both loss of precision and granularity.

We discuss how useful uncertainty can be imposed and utilized for privacy-preservation purposes in Chapter 5. In the next chapter, we first discuss the metrics that can be used to measure the different causes and effects of encoding and decoding uncertainty, that are subsequently extended for privacy-preservation purposes and used for high-dimensional, temporal data visualization (Chapter 6).

CHAPTER 4: SCREEN-SPACE METRICS

The choice of metrics is based on our visual uncertainty taxonomy that enables us to define measures for both the encoding and decoding stages of the visualization pipeline. Several purposes or tasks motivate the design of visual representations and interaction designs. Our metrics are motivated by the tasks proposed by Amar and Stasko [6] and by the requirement of the visual representation being perceptually beneficial. Based on the taxonomy, we have translated the high-level goals of these tasks and requirements into the low-level components of visual uncertainty. This has enabled us to systematically define metrics that quantify the causes and effects of uncertainty, occupying the leaf nodes of the taxonomy tree. In this chapter, we take a bottom-up approach by describing the quantification model, the visual features and metrics for measuring encoding and decoding uncertainty; and then we discuss how they fit together within the taxonomy. Since privacy is a relatively new concept in the realm of visualization, metrics that are typically applicable for evaluation of privacy-preserving visualization, are discussed in Section 5.6 after introducing the techniques. Unless otherwise stated the proposed metrics are applicable to both parallel coordinates and scatter plot matrices.

4.1 Model

Pargnostics [37] are a set of screen-space metrics for parallel coordinates. As such, they inherently depend on the size of the display, measured as the number of pixels. Although

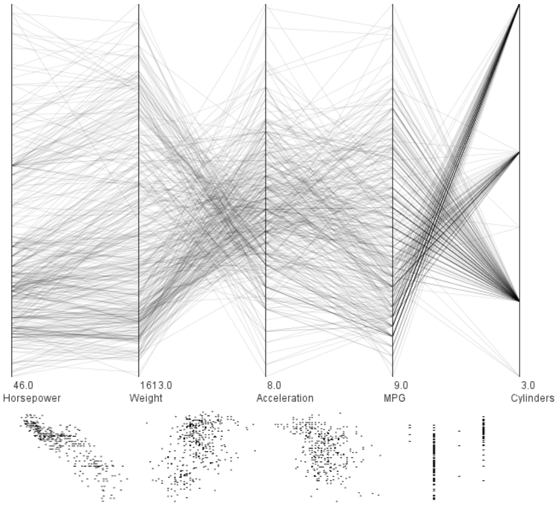
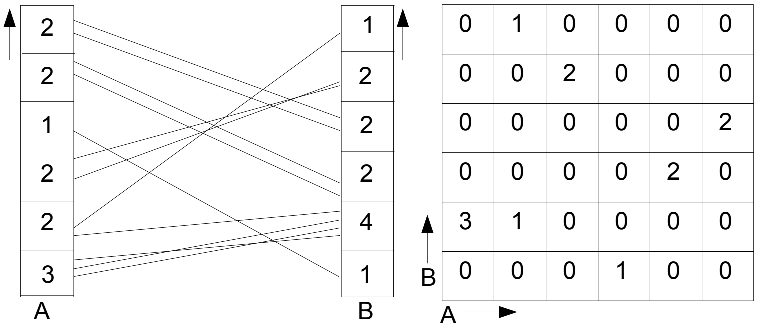


Figure 14: Scatter plot features mapped to parallel coordinates: The right axis of each axis-pair corresponds to the X-axis of the scatter plot and is labelled accordingly.

there has been substantial improvement in display technologies and gigapixel displays are not entirely unheard of now-a-days, the most common analytical tasks are done with visualizations on desktops, laptops and tablets, where the number of pixels is in the order of hundreds or thousands. On the other hand, the cardinality of data produced from scientific simulations or business transactions typically range from millions to billions. The quantification model and the subsequent metrics based on that addresses this disparity between number of pixels and number of data points, that manifests mainly during the visual mapping process.

The model is based on the common parallel coordinates layout, which draws vertical axes and lays the axes out horizontally from left to right.

We consider a parallel coordinates display to consist of a series of axis pairs, similar to scatter plots in a scatter plot matrix (Figure 14). The space between a pair of axes is where interesting patterns such as parallel or crossing lines, aggregations of lines, etc., can be observed. As pointed out by Li et al. [75] finding correlation in parallel coordinates



0	1	0	0	0	0
0	0	2	0	0	0
0	0	0	0	0	2
0	0	0	0	2	0
3	1	0	0	0	0
0	0	0	1	0	0

Figure 15: Illustrating binning with equal-width histograms: On the left is the binned space of the parallel coordinates and on the right is the degree of each bin in the 2D histogram.

ultimately boils down to finding those patterns between pairs of axes. This is also true for all relevant structures that the user is looking for in the data. Neighboring axis pairs of course depend on each other because the left or right axis has to correspond to the right or left axis of the neighboring pair, respectively.

4.1.1 Pixel-Space Histograms

The basis for most metrics is pixel-based binning (Figure 15). With each axis being h pixels high, we define three types of histograms: one-dimensional axis histograms, one-dimensional distance histograms, and two-dimensional axis pair histograms.

All binning is done in screen space, after the values have been transformed into pixel coordinates. The data values, v_i, are projected onto screen pixel coordinates, l_i (for the left axis in a pair) or r_i (for the right axis), by a mapping function f: $(l_i, r_i) = f(v_i)$. We disregard any padding around the axes, so $0 \leq l_i < h$.

One-Dimensional Axis Histogram. The basic one-dimensional axis histogram has h bins, and consists of bins b_i that count the number of lines starting or ending in them. Metrics like axis entropy, density median and over plotting are based on this histogram. In the following formula, i is in the range $[0; h-1]$.

$$b_i = |\{k \mid \lfloor l_k \rfloor = i\}|$$

One-Dimensional Distance Histogram The one-dimensional distance histogram records the slope of the lines between the axes, measured as the vertical distance in pixels. This measure can be used to calculate the actual angle when the spacing between the axes is given. The steepest line going up or down covers the entire height of the display, and can thus have a vertical distance in the range $(-h; h)$. This leads to a total of $2h-1$ bins, as the up and down ranges overlap at the value 0. This histogram is used as a basis for the parallelism metric.

$$d_i = |\{k \mid \lfloor r_k - l_k \rfloor = i\}|$$

Two-Dimensional Axis Pair Histogram. Looking further at the space between the axes, we define a histogram of all the lines, covering both axes. This is a two-dimensional histogram, consisting of bins b_{ij}, with both i and j in the range $[0; h-1]$.

$$b_{ij} = |\{k \mid \lfloor l_k \rfloor = i \wedge \lfloor r_k \rfloor = j\}|$$

This is the basis for the calculation of the number of line crossings, crossing angles, convergence/divergence and mutual information.

4.2 Measuring Encoding Uncertainty

On the encoding side, we propose metrics for quantifying information loss such as entropy and over plotting and for quantifying relative information content of dimensions using mutual information.

4.2.1 Axis Entropy

To quantify the characteristic of the data distribution, we have developed two metrics: axis entropy and density median. Axis entropy signifies the variance in the distribution, if there is a lot of uncertainty involved with the distribution or it is pretty even, leading to high entropy. Entropy measures the uncertainty or disorder within the data values and we use Shannon entropy [32] to capture this characteristic. We consider each axis in the screen-space as a random variable. We use the probability of intersection of a data record with a pixel-bin, computed from the frequency of each pixel-bin on an axis as the basis for computing Shannon entropy. Entropy for an axis (A) in terms of its pixel bins $(a1, a2, \ldots a_h)$ is given by:

$$H(A) = -\sum_{k=1}^{h} p(a_k) log(p(a_k)) \tag{1}$$

where $p(a_k) = \frac{1}{\beta_k}$. When the entropy (H(A)) is plotted over time, the trajectory of the time-series indicates the overall stability or instability of the variable.

The density median quantifies if high or low concentrations of a variable are manifested at a time step. Density median $(\tilde{D})$ is computed from the median of the frequencies of the pixel-bins in a one-dimensional axis-histogram. The location, that is the pixel coordinate of the median $(\tilde{\beta})$, is then plotted over time. A high value of the median at a particular time step means dominant values at that time step are the high ones and a low value means dominant values are the low ones.

Both the density median and axis entropy metric are based on univariate properties, meaning, they are independent of axis adjacency in parallel coordinates. The two components of our data density metric are : the density median determining the locus of skewness, and the nature of density in terms of data disorder or randomness.

4.2.2 Mutual Information

In exploratory data analysis, the information the user is looking for is subjective: task oriented and difficult to model. Mutual information [32] provides a general measure of dependency between variables; its value is zero when they are conditionally independent. Pearson correlation coefficient is extensively used for this purpose, but a lack of such correlation does not imply that two variables are independent. In Pargnostics, we treat the data dimensions as random variables, and our computation is based on the binned space. The probability that a dimension assumes a particular data value is therefore equivalent to the binned value in the screen space.

Let X and and Y be random variables. Mutual information $I(X;Y)$ is defined in terms of probability distributions as:

$$I(X;Y) = \sum_{i=1}^{h} \sum_{j=1}^{h} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i) p(y_j)}$$

In this case $p_i = \frac{b_i}{h}$, using the one-dimensional axis histogram, where $p(x_i, y_j)$ denotes the joint probability of random variables X and Y and in this case $p(x_i, y_j) = \frac{b_{ij}}{h}$ using the two-dimensional axis histogram (Section 4.1.1).

Some of the dimensions which exhibit high mutual information (Figure 56) in the cars data are *MPG* and *weight*, *weight* and *acceleration*, *horsepower* and *weight*, etc. This is expected as we know lighter cars have good fuel economy and better acceleration. Thus mutual information provides an indication to the user, which axis pairs are likely to convey interesting information. We also use the mutual information metric for cluster based, privacy-preserving visualizations, that are discussed in Section 5.6.

4.2.3 Over-plotting

Over-plotting is a measure of the quality of a visualization. This is especially relevant in the case of parallel coordinates. Here, due to the large dataset sizes and a limited number of screen pixels, several data points can be mapped to the same pixel value on two adjacent dimensions. The degree of over-plotting helps to estimate the information density between adjacent dimensions. High over-plotting leads to dense cloud of lines and the user might often be interested in a view that minimizes it. Quantitatively, over-plotting is a side effect of binning and is therefore an indication of the information loss that occurs during processing. Having different bin size would directly affect over plotting and help us achieve controlled information loss. This information loss can be intended or unintended based on the context of application. For example, in a privacy-preserving application we intentionally want to hide the precise values, and the over plotting metric helps us control the information loss. This is also the basis for selecting seeds for privacy-preserving clustering and that is described in Section 5.3.

The degree of over-plotting *O* is computed from the count of each bin of a two-dimensional histogram. Each count in a bin greater than 1 contributes to over-plotting.

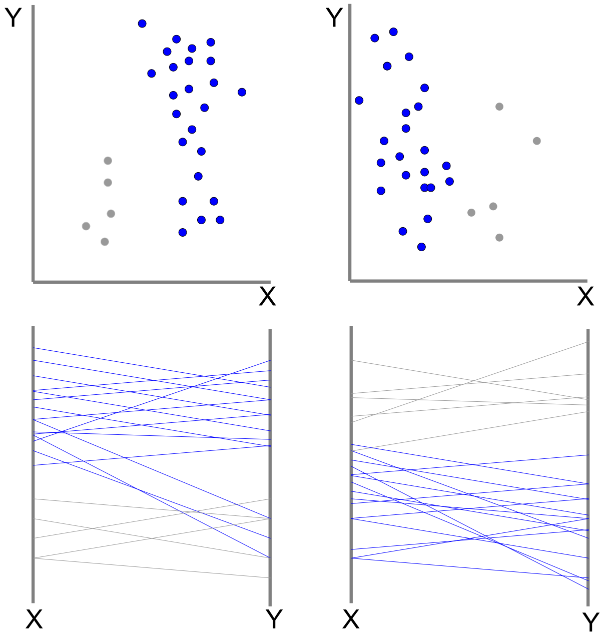


Figure 16: Density median computes if the high values (as in the left image) or low values (as in the right image) are dominant.

$$O = \sum_{i=1}^{h} \sum_{j=1}^{h} \begin{cases} b_{ij} & \text{if } b_{ij} > 1 \\ 0 & \text{otherwise} \end{cases}$$

The total degree is then normalized by the number of data points n to give the normalized degree of over-plotting:

$$O_{norm} = \frac{2O}{n(n-1)}$$

4.3 Measuring Decoding Uncertainty

On the decoding side, we propose metrics for measuring visual quality and salient visual patterns. Visual quality is measured in terms of human ability to discriminate among a number of crowded or collocated data points. The visual patterns that we quantify are those for correlations and clustering.

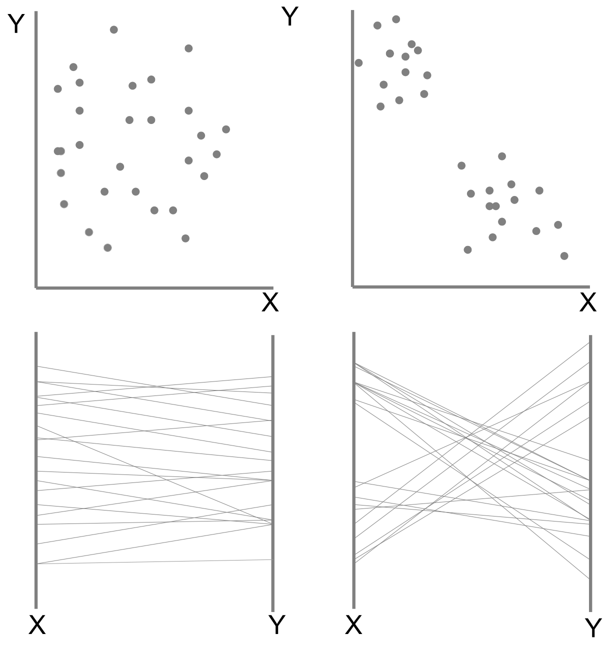


Figure 17: Axis Entropy determines disorder or randomness within the data. Low entropy generally signifies visually detectable patterns as in the right image.

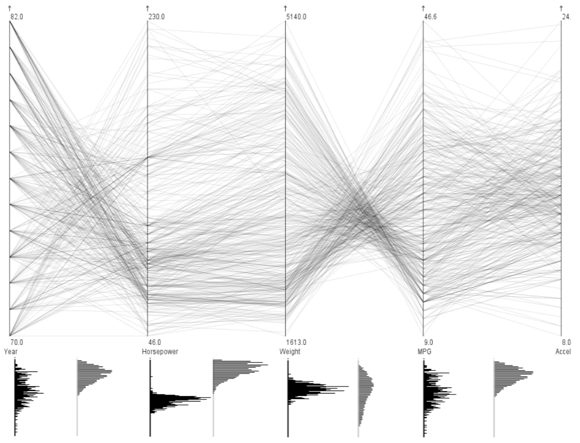


Figure 18: Distance histograms (left half of each cell below the parallel coordinates) and angles of crossings (right half) histograms for different dimensions of the cars data.

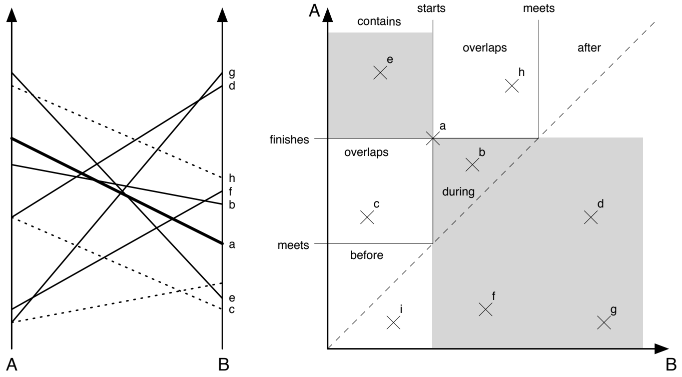


Figure 19: Using the interval classification, we can simplify the structure by using Rit's sets of possible occurrences [96]. The different lines in parallel coordinates are shown in the scatter plot/SOPO diagram on the right. The thick reference line *a* is used as the reference on the right. All intervals that fall into the shaded areas represent lines that cross the reference line, those lines are drawn as solid lines; dotted lines do not cross the reference line.

4.3.1 Number of Line Crossings

A prominent feature of parallel coordinates are crossing lines between pairs of axes. Many crossing lines typically mean some kind of inverse relationship, especially if the crossings occur mostly around the center. Line crossings also contribute to clutter and can make it difficult to identify which lines are going where. The importance of line properties in implying correlations have been studied [77, 118]; Ellis and Dix [49] also dealt with line intersections as a cause for display clutter. Also, user studies [75] show that judging correlations can be a complicated task because of several artifacts in parallel coordinates. Since line crossings directly affect correlations, especially inverse correlations, it is important the user is able to maximize or minimize it for optimization.

To efficiently calculate the number of crossings, we interpret each line between a pair of axes as a directed interval. Allen's interval algebra for temporal reasoning [5] enumerates all possible relationships between pairs of time intervals: A *before* B, A *meets* B, A *over-*

Table 2: Conditions for intersection based on interval relation and direction, grouped and ignoring symmetrical cases (see Figures 20 and 19).

Relation	Direction	Intersection
before/after	same	no
	opposite	no
meets	same	no
	opposite	no
overlaps	same	no
	opposite	**yes**
during	same	**yes**
	opposite	**yes**
starts/finishes	same	no
	opposite	**yes**
equals	same	no
	opposite	**yes**

laps B, A *starts* B, A *finishes* B, and A *equals* B. All relationships except *equals* can also be applied as B R A, leading to a total of 13 different ones.

For the purpose of determining line crossings, we add the direction of the line or interval as a second attribute (Figure 20). We are not interested in the absolute direction, but only whether the two lines are pointing in the same or opposite directions. Disregarding the *before/after* and *meets* relations, which trivially cannot lead to intersections, we can classify which relationships and directions lead to intersecting lines and which do not (Table 2).

This classification can be simplified by making use of work by Rit, who devised a way to graphically propagate temporal constraints [96]. Similar to the point-line duality between scatter plots and parallel coordinates, sets of possible occurrences (SOPOs) depict intervals as points on a two-dimensional diagram with two time axes: one for the start and one for the end of the interval. Combining all the areas that correspond to the line crossing conditions, we find a simple rule for determining whether two lines cross (Figure 19).

Given the two-dimensional histogram, we can calculate the number of line crossings, L:

$$L = \sum_{i=0}^{h-1} \sum_{j=0}^{h-1} \sum_{k=i+1}^{h-1} \sum_{l=j+1}^{h-1} b_{ij} b_{kl}$$

This leads to a complexity of $O(h^4)$, though in practice it is much lower: if b_{ij} is zero, the program does not need to enter the two innermost loops. In real-world data sets, the histogram is very sparse, so this condition has a large effect on algorithm runtime.

4.3.2 Parallelism

In addition to line crossings, there are also structures where lines are parallel or close to parallel to each other. Such lines can mean correlations between two dimensions [7].

Li et al. described the difficulties user have judging correlations in a parallel coordinates environment [75]. Parallelism can also imply closely aggregated lines or clusters within a subset of the data. When given the choice between line crossings and parallel lines, the latter are often preferable to crossings because they lead to less clutter. This is of importance for the display optimization described below.

To describe parallelism, we compute a vertical distance histogram between any two connecting points on adjacent axes (Figure 18). Positive values in this histogram mean lines going up, negative values indicate lines pointing down. Axis pairs with a high degree of parallelism tend to have narrow distributions of directions, while ones with no or very little apparent parallelism cover the entire distance spectrum with no apparent clustering of values.

Parallelism is defined both in terms of direction and extent. The median distance value indicates the direction and the extent of parallelism is given by the interquartile range: a narrow interquartile range implies high parallelism. We normalize the distances between 0 and 1, by dividing by the highest possible distance. We then compute parallelism P_{norm} as follows based on the interquartile range between the 25% and the 75% quartiles, q_{25} and q_{75} .

$$P_{\text{norm}} = 1 - |q_{75} - q_{25}|$$

The subtraction is done to get a higher parallelism value for a higher degree of parallelism (and thus a smaller interquartile range). The direction is given by the median M_P, which is not normalized (the direction only makes sense in pixel coordinates):

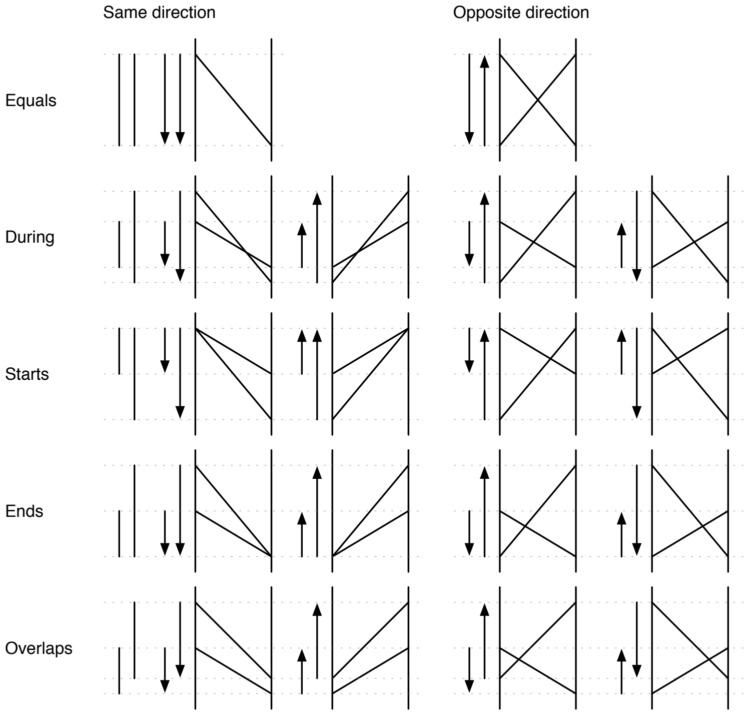


Figure 20: The possible configurations of two lines in parallel coordinates, classified using Allen's interval algebra. *Before*/*after* and *meets* relations are not shown, because they trivially mean no intersection. See Table 2 for a classifications of crossings using these criteria.

$$M_P = q_{50}$$

4.3.3 Angles of Crossing

While the number of line crossings can be an issue, an equally important one involves the angles at which lines cross. Lines crossing at flat angles are harder to follow than ones crossing at close to right angles [62, 117]. Lines that are part of clusters also tend to cross at low angles [118].

We calculate the crossing angles between pairs of lines. Except for parallel lines, any two lines will have two crossing angles that add up to 180°. Since we care about how close to a right angle the lines cross, we choose the smaller one of the two. The angle calculation is only performed for lines that are known to cross based on the crossing lines conditions above. They are then rounded and binned into whole degree bins, and we calculate the median crossing angle. The resulting histogram (Figure 18) is used for display purposes, while the median is used as the criterion for optimization.

The calculation is based on the two-dimensional axis histogram. For each pair of bins that are found to contain crossing lines, we calculate the crossing angle. This means that for multiple lines crossing at the same point, we only have to perform the calculation once, and can add $b_{ij}b_{kl}$ to the histogram (similar to the way it is done for the number of crossings above).

4.3.4 Convergence, Divergence

A common pattern in parallel coordinates is comprised of few values on one axis that branch out to many values on the other. Similar to *clumpiness* in Scagnostics there can be many lines converging to a point or diverging from a point between adjacent axes. They represent sine functions with multiple periods [51]. These structures are a useful characterization of the data points as they reveal associative relationships: many-to-one or one-to-many relationships between points on adjacent axes.

Convergence. Convergent structures between a pair of axes comprise of those lines on the left axis which converge to a single bin on the right axis. Mathematically, the total convergence C between two axes can be calculated as:

$$C = \sum_{i=1}^{h} \sum_{j=1}^{h} \begin{cases} 1 & \text{if } b_{ji} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Divergence is the mirror image of convergence and is given by

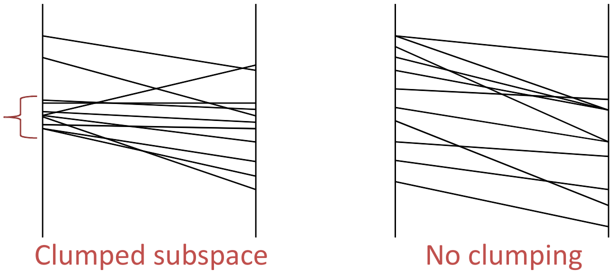


Figure 21: Illustrating clumping metric in for measuring subspace densities in parallel coordinates.

$$D = \sum_{i=1}^{h} \sum_{j=1}^{h} \begin{cases} 1 & \text{if } b_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$$

The ratios $C_{avg} = \frac{C}{n}$ and $D_{avg} = \frac{D}{n}$ where n is the total number of lines between two axes, indicate the average degree of convergence or divergence between each axis. To give an overview of the degree of convergence/divergence present between each pair of dimensions, we show histogram of convergence or divergence, whichever is greater. The values are normalized as follows:

$$C_{norm} = \frac{C}{\max(b_{ij})}$$

$$D_{norm} = \frac{D}{\max(b_{ij})}$$

Converging/diverging lines have an implicit aggregation among them. And they are more prominent between adjacent categorical dimensions. Thus convergence/divergence, like over plotting, is used for seed selection in privacy-preserving clustering in the screen-space.

4.3.5 Subspace Clumping

Earlier we showed that converging and diverging structures are interesting structures in parallel coordinates. However, in case of large datasets, lines do not converge to or diverge from precisely a pixel, but from a local neighborhood of pixels (Figure 21). Highlighting these neighborhoods is a way of implicit clustering and also an indicator of multiple modes in the data. Computation of clumped subspaces requires two parameters: the number of contiguous neighbors and strength of a neighborhood that is considered as clumpiness. Our algorithm has two subroutines: first, the average sparseness in an axis pair is computed and second, the sparseness value is used to decide when to cut off a clumping cluster. The average strength of the clumped clusters gives us the clumping factor (C_f).

For determining the number of contiguous neighbors, we first compute sparse regions based on the strength of the pixel bins. For this we use the two-dimensional histogram defined based on all the lines, covering the adjacent axes. This histogram, consists of bins $\beta_{k,l}$, with both k and l in the range $[0; h-1]$. A clumping cluster is cut off once a sparse region is found. Let us denote a sparse bin by be indicated by $\bar{\beta}_{k,l}$.

The convergence-divergence metric is used as a selection criteria by the algorithm. The axis with higher average convergence/divergence is selected and iterations through the pixel-bins on that axis are performed, following the two subroutines that are described as follows:

Subroutine for computing sparse regions:

1. Set threshold for clumpiness to the average frequency of a bin, i.e., the number of records divided by number of bins, i.e. $t = \frac{n_r}{h}$.

2. If the bin frequency is less than the quartile of the threshold, i.e., $\bar{\beta}_{k,l} < 0.25t$, consider the bin as a sparse bin.

3. Compute the number of contiguous sparse bins (η) for each sparse region found. Let a sparse region be denoted by e.

4. Compute average sparseness as the number of contiguous sparse bins divided by the number of sparse regions. So we get

$$avg(\eta) = \frac{1}{e}\sum_{c=1}^{e}\eta_c$$

Subroutine for computing clumping factor:

1. If a bin contains lines greater than t, add it to the cluster.

2. Continue adding bins until a sparse bin ($\bar{\beta}_{k,l}$) is found.

3. If the number of contiguous sparse bins is less than $avg(\eta)$, add those bins to the cluster, else break the cluster.

4. Repeat steps 1 to 3 for all the remaining bins.

5. Add the number of bins in each clumped cluster. Let the number of contiguous clumped bins in each cluster be denoted by ζ. Divide by the number of such clusters (v). That is the clumping factor (C_f).

$$C_f = \frac{1}{v}\sum_{c=1}^{v}\zeta_c$$

Source of Uncertainty	Cause/Effect	Metric	Visualization Technique	Application
Encoding:Data Mapping	Configuration, Pattern Complexity	Axis Entropy	PC, SP	HDDV
		Mutual Information	PC,SP	HDDV, PPDV
Encoding:Visual Mapping	Precision	Cluster Summary Error	Clustered PC, Clustered SP	PPDV
		Over-plotting	PC,SP	HDDV
	Granularity	Cluster Range	Clustered PC,Clustered SP	PPDV
	Granularity, Spatial Accuracy	Cluster Configuration	Clustered PC,Clustered SP	PPDV
Decoding:Perception	Traceability,Identity, Pattern Complexity	Line Crossings	PC	HDDV
		Crossing Angles	PC	HDDV
		Overlap Entropy	Clustered PC,Clustered SP	PPDV, HDDV
		Overlap Clutter	Clustered PC,Clustered SP	PPDV,HDDV
Decoding:Cognition	Pattern Complexity	Parallelism	PC, Clustered PC	HDDV
		Convergence/Divergence	PC, SP	HDDV
		Clumping Factor	PC, SP	HDDV
	Lack of Knowledge	Disclosure Risk	Clustered PC, SP	PPDV

Table 3: Screen-space metrics that have been systematically defined based on the sources, causes and effects of visual uncertainty they help quantifying in parallel coordinates (PC) and scatter plots (SP) for high-dimensional data visualization (HDDV) and privacy-preserving data visualization (PPDV). A single metric can measure multiple uncertainty causes and a single uncertainty cause can be measured by multiple metrics based on our conceptualization.

4.4 Discussion

In Table 3 we classify the different metrics according to the different causes and effects of uncertainty for the purposes of high-dimensional data visualization and privacy-preserving data visualization.

4.4.1 Encoding Uncertainty Metrics

An instance of analytical task to visual uncertainty mapping is the reduction of configuration uncertainty through information-theoretic metrics like entropy and mutual information that guide the adjacency of dimensions in parallel coordinates and scatter plot matrices. Since these metrics essentially quantify the information content on screen, on the decoding side they measure pattern complexity. Information loss due to loss of precision is measured by over plotting and axis entropy. In case of privacy-preserving clustering, loss of granularity is measured by cluster range (Section 5). For privacy, the location and orientation of records is also important in separating susceptible clusters from the secure ones. Cluster configurations help measure that.

4.4.2 Decoding Uncertainty Metrics

An example of the task to visual uncertainty mapping in case of high dimensional analysis is finding correlations through reduction of pattern complexity, which is quantified by parallelism. Line crossings and angles of crossings in parallel coordinates help quantify discriminability in terms of identity and traceability uncertainties, both within axis pairs and across adjacent axes. Line crossings also quantify pattern complexity as crossings in the middle denote inverse correlations. The same applies for angles of crossing, as low crossing angles denote aggregation properties. Convergence/divergence and clumping factor denote clustering properties and thus quantify pattern complexity. These metrics are mainly applicable for high-dimensional data visualization. For privacy-preserving visualization, the derivatives of axis entropy and line crossings, which are overlap entropy and overlap clutter (Section 5) respectively help control visual quality, while at the same time, address privacy concerns. Moreover, the degree of user knowledge about the data directly affects the disclosure risk at different levels of granularity, that is discussed in Section 5.

CHAPTER 5: PRIVACY IN VISUALIZATION

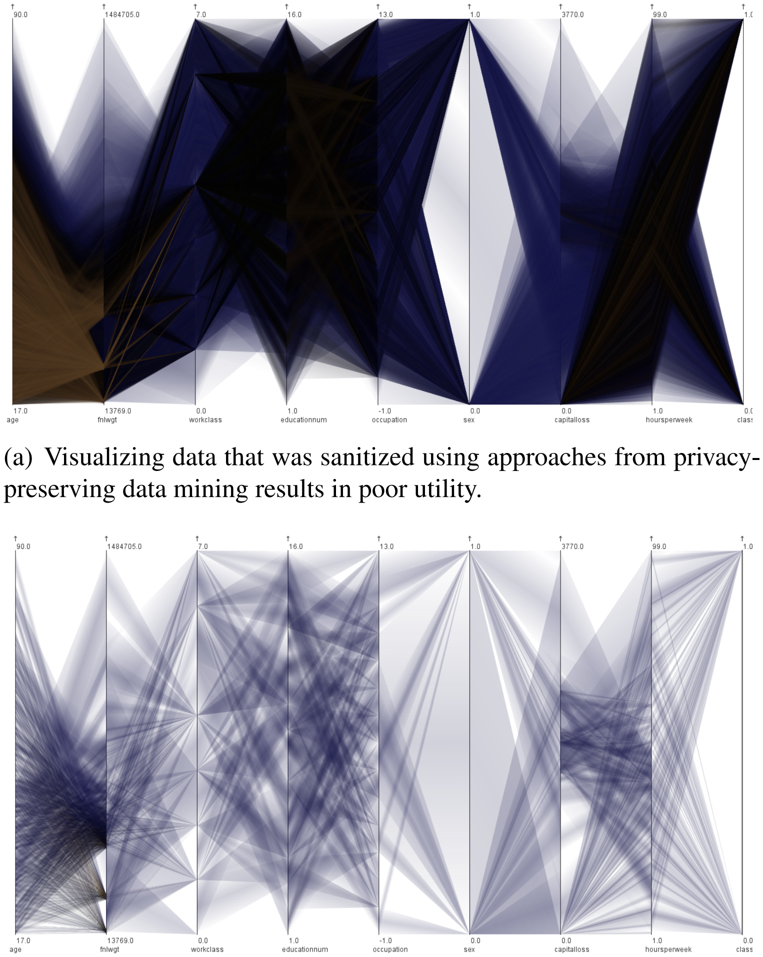
The increase in size and complexity of real-world data demands more effective information visualization and visual analytic techniques. But accessibility is a key issue to much of this data due to privacy concerns. Medical records, financial information, sensitive corporate data, etc. can only be shared with outside users for analysis purposes after explicit identifiers have been removed (i.e., the data has been *sanitized*). There are also regulations that make such disclosures punishable by law, like the *Health Insurance Portability and Accountability Act* (HIPAA) in the United States. Even after removal of explicit identifiers, data from public databases might still make it possible to identify individuals in such sanitized data.

Privacy-preserving data mining (PPDM) distinguishes between *quasi-identifiers* and *sensitive attributes*. Sensitive attributes are those whose exact values need to be protected, so that they cannot be linked to an individual. Examples include disease names in medical databases and company-specific information in corporate databases. Quasi-identifiers are those attributes that, taken together, can identify an individual even if that person's name or complete address is not included. This is possible because the same information can be found in public databases, and enough quasi-identifiers can narrow the choices down to a single person (e.g, given a date of birth, gender, and zip code, it is often possible to pinpoint a single person).

The problem with PPDM techniques is that even for a minimum privacy guarantee, there is a significant loss of utility [22]. Moreover, the published output might be susceptible to mining by malicious users, who might use the analysis results to breach the privacy safeguards. The preservation of the privacy of published results has been termed *output privacy* [23]. While a lot of work has been done on protecting the privacy of the input data to the mining algorithm, not much work has focused on output privacy protection. Our approach includes a consideration of output privacy, but is situated between input and output privacy.

In our work,we treat the problem of information privacy from a visualization perspective. Instead of publishing the data or just the analysis results, a privacy-preserving information visualization tool provides an interactive interface to both the data owner and outside users. The data owner can customize the tool to choose different views of the data he wants to show to analysts without sacrificing privacy. The outside user cannot directly access the data, but only visualize the patterns in the data through the tool. Different constraints that are imposed within the model to prevent him from breaching privacy through interaction. The data is sanitized on the fly, based on the user's screen resolution and other viewing parameters.

Similar to PPDM, we assume that the data holder is aware of the the sensitivity of the data attributes and in what context the privacy can be breached. The main concern with sensitive data is their misuse [31, 18]. Privacy can be personal (e.g., medical records) or corporate (e.g., company records) [31]. In both cases the goal of a privacy-preserving technique is to minimize disclosure of sensitive information even after data analysis techniques have been used. At the same time, the access of non-sensitive information and their fidelity should



(a) Visualizing data that was sanitized using approaches from privacy-preserving data mining results in poor utility.

(b) Clustering by axis pair and using a different metric in the clustering, more of the visual structure can be retained.

Figure 22: Comparing data clustering to our visual clustering approach. Both algorithms make it impossible to tell fewer than three records apart, but our approach provides higher utility.

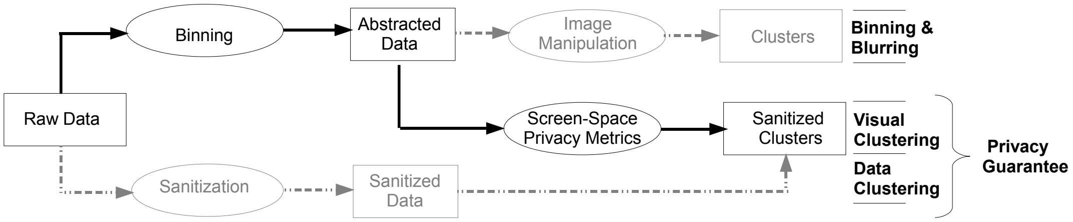


Figure 23: Possible approaches to privacy-preservation in information visualization (Section 5.1): binning and blurring, data clustering, and visual clustering. Only the bottom two approaches guarantee a given level of privacy, in the case of data clustering, it comes at the price of considerable utility loss.

not be affected by the sanitization process.

In this chapter, we first discuss the motivation of using screen-space metrics for preserving privacy and the basic elements of a privacy-preserving visualization model (Sections 5.1 and 5.2). Next we demonstrate the applicability of the model by describing the technique and its various features. We apply clustering in parallel coordinates based on the well-established k -anonymity and l-diversity metrics. We discuss those in details in Section 5.3.

5.1 Potential Approaches

Why develop a new approach to hiding information in visualization? There are more obvious approaches, like blurring the image or simply visualizing the output of a PPDM sanitization technique. This section discusses the shortcomings of these naïve approaches (see also Figure 23).

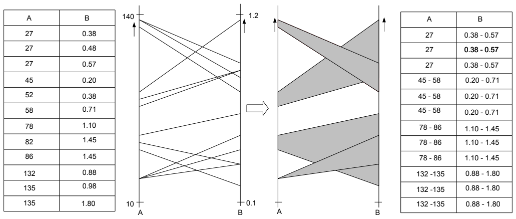
Binning and Blurring: When a dataset is visualized, there is a natural loss of precision due to the limited resolution of the screen. To add to the information loss, the image could be blurred to hide individual records. The drawback of this approach, however, is the limited control over information loss, in particular when it comes to privacy. Single data points might exist far enough away from others so as not to be blurred together with them. There is no guarantee that each visible point contains at least a given minimum number of records.

Data-Space Sanitization: Another approach is applying the sanitization algorithms proposed in the data-mining literature and then visualizing the resulting data. While that guarantees a level of privacy, it also comes at the price of greatly reduced utility, a problem that is know well in PPDM [22]. The resulting visualization is very close to being useless (Figure 22(a)), as the clusters cover much more of the axes than using our approach.

Screen-Space Sanitization: Effective visual representation is one of the key factors that lead to high utility in visualization [81]. This requires modeling the appearance of a visualization on screen, and controlling the attributes of the visualization to control the amount of information that is shown to the user. Using visual metrics for the sanitization process, it is possible develop a clustering that is much better suited for the purposes of visualization. This is the approach we describe in this work.

5.2 Basic Elements of a Privacy-preserving Visualization Model

In visualization the limited number of screen pixels is a key constraint that needs to be taken into account while mapping values from data to the screen. This results in information loss and we generally strive to minimize it. Ensuring information privacy in the screen space means the information loss entailed is both necessary and sufficient to protect the sensitivity of the represented data. In this section we discuss the different sanitization approaches from the perspective of information loss and demonstrate how we can achieve privacy-preservation in visualization in general.



A	B
27	0.38
27	0.48
27	0.57
45	0.20
52	0.38
58	0.71
78	1.10
82	1.45
86	1.45
132	0.88
135	0.98
135	1.80

A	B
27	0.38 - 0.57
27	**0.38 - 0.57**
27	0.38 - 0.57
45 - 58	0.20 - 0.71
45 - 58	0.20 - 0.71
45 - 58	0.20 - 0.71
78 - 86	1.10 - 1.45
78 - 86	1.10 - 1.45
78 - 86	1.10 - 1.45
132 -135	0.88 - 1.80
132 -135	0.88 - 1.80
132 -135	0.88 - 1.80

Figure 24: Concept of *k*-anonymity applied in parallel coordinates. Individual lines are clustered in the form of trapezoids, after binning. In this case, $k = 3$.

5.2.1 Modelling Information Loss

There are two types of information loss during the visual mapping process: *intended* and *unintended* [130]. Binning, described earlier in Section 4.1 is an example of intentionally losing information to fit the data on screen. Over-ploting, clutter are examples of unintended information loss and are a function of the different artifacts produced on screen. For achieving privacy we want to have a controlled information loss by manipulating both these types of losses. We distinguish between two types of information loss in the context of privacy, both of which result in data hiding: a) loss of precision and b) loss of granularity.

5.2.1.1 Loss of Precision

Loss of precision occurs due to the quantization that results as a mapping from data space to screen space. A common example of data abstraction to fit the data on screen is binning. This is an example of surjective mapping where more than one data elements are represented by a pixel. Loss of precision might appear to be a good enough condition to protect the sensitivity of private data. We will show here that this not sufficient, though. The precision loss incurred is equal to Δb which represents the bin size. Information content of the binned data space can be represented by I_s where $I_s = \log_2 \frac{range_{bin}}{\Delta b}$ [125]. If $\Delta b > 1$ we indeed lose some information.

However, this is not sufficient because it does not constrain the number of records in a bin; a bin might only contain one record, making it possible to drill down to the individual value. Since we require the fine-grained details to be hidden from a malicious user, this is therefore not a sufficient condition for achieving privacy.

5.2.1.2 Loss of Granularity

The desirable condition for privacy-preservation is loss of granularity, i.e, the user sees just aggregated data and is unable to breach the privacy by accessing the fine-grained details of the records. Loss of granularity subsumes loss of precision. To achieve this, we apply some constraints on the visibility of the data by not showing individual records, but clusters of records. This can be achieved in two ways: *Data-space sanitization* and *Screen-space sanitization*.

In the context of parallel coordinates we hide fine-grained information by showing clusters instead of lines. Privacy protection is done at two levels: *record-level* and *cluster-level*. In record-level privacy we ensure that each record belongs to a cluster of membership k such that it is indistinguishable from those $k-1$ other records (k-anonymity [106, 114]). An adversary can then link a sensitive attribute to at least k records with a probability P_k = $\frac{1}{k}$. In cluster-level privacy the goal is that there is certain amount of diversity in clusters such thateach cluster belonging to a quasi-attribute can be linked l different clusters belonging to the sensitive attribute [82], therefore leading to diversity in the possible values. Therefore probability of linking is reduced as $P_l = \frac{1}{kl}$

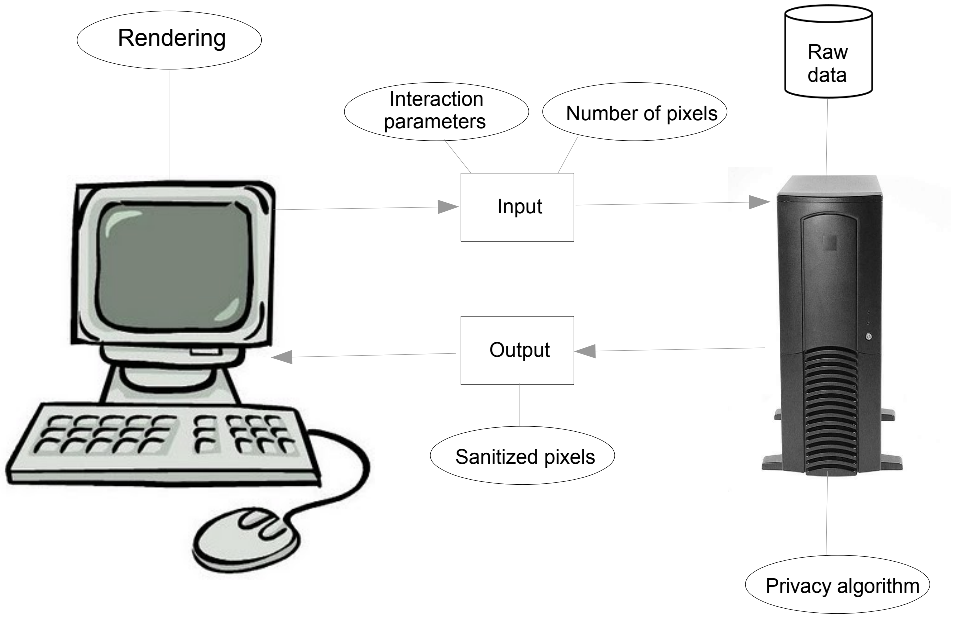


Figure 25: Client-server architecture for handling privacy in visualization.

5.2.2 Remote Visualization

Our technique is based on a client-server architecture (Figure 25), where the data resides on the server and the client only fetches the clustered data and displays it on screen. This model is similar to the idea of interactive privacy described by Dwork [46], where an interface is provided by data owners to interactively filter responses to user queries and add noise where needed to preserve privacy. Similarly, in our system, the user cannot access the raw data, but can only set the interaction parameters necessary (in this case order of dimensions and number of pixels) for the server to apply the privacy-preservation algorithm (in this case, clustering) and return the sanitized clusters. Axis order is important: the server has knowledge of which dimensions are sensitive and which ones are quasi-identifiers so the output is tailored towards that configuration. The screen-space metrics, used as a starting point for clustering in our technique, are dependent on pixel-based binning. The number of pixels thus determines the appearance of the clusters.

5.3 Implementation of *k*-anonymity

The primary goal of privacy-preserving visualization is that the user should not be able to access the quasi-identifiers and/or the sensitive attributes in the raw data. Therefore, we intentionally hide information from the user by imposing de-identification constraints in the screen-space. To achieve this, we exploit two types of information loss: the inherent information loss in parallel coordinates for ensuring loss of precision and additional loss from grouping together records as a necessary step for sufficient loss of granularity to achieve a desired level of privacy.

In the privacy-preserving variant of parallel coordinates, based on the idea of *k*-anonymity, our program combines *k* records into one cluster, and displays it as a trapezoid instead of as individual lines (Figure 24). We have adapted an existing clustering mechanism in order to maximize the utility of the resulting clusters for visualization [24]. We use pixel space coordinates for our model. That means that all coordinates are first transformed into screen space and then rounded to integers. This places them in pixel-sized bins, which reflect the lower boundary of precision of the display. The clustering is done after transforming the records to their respective pixel-coordinates and is based on the properties of the pairs of adjacent axes and the steps are as follows:

5.3.1 Seeding

The quality of the clusters depend strongly on the initial seeds we choose. Axis-pairwise clustering enables us to look at the type of information loss between adjacent dimensions and based on that select our seeds at different stages of the iteration. We have initially used the degree of over-plotting as the seeding criteria. However, there are other properties of the bins convergence-divergence [37] that needs to be taken into account: Categorical dimensions or numerical dimensions with very few distinct values are likely to have more converging/diverging structures. Numerical dimensions with a more even distribution are likely to have more over-plotting. The seeding algorithm we use here, takes into account these properties of the axis-pairs and also chooses the seeding dimension and seeding bin, i.e, the bin from which a seed record is chosen, accordingly. The steps are as follows:

1. Determine if axes are numerical or categorical.

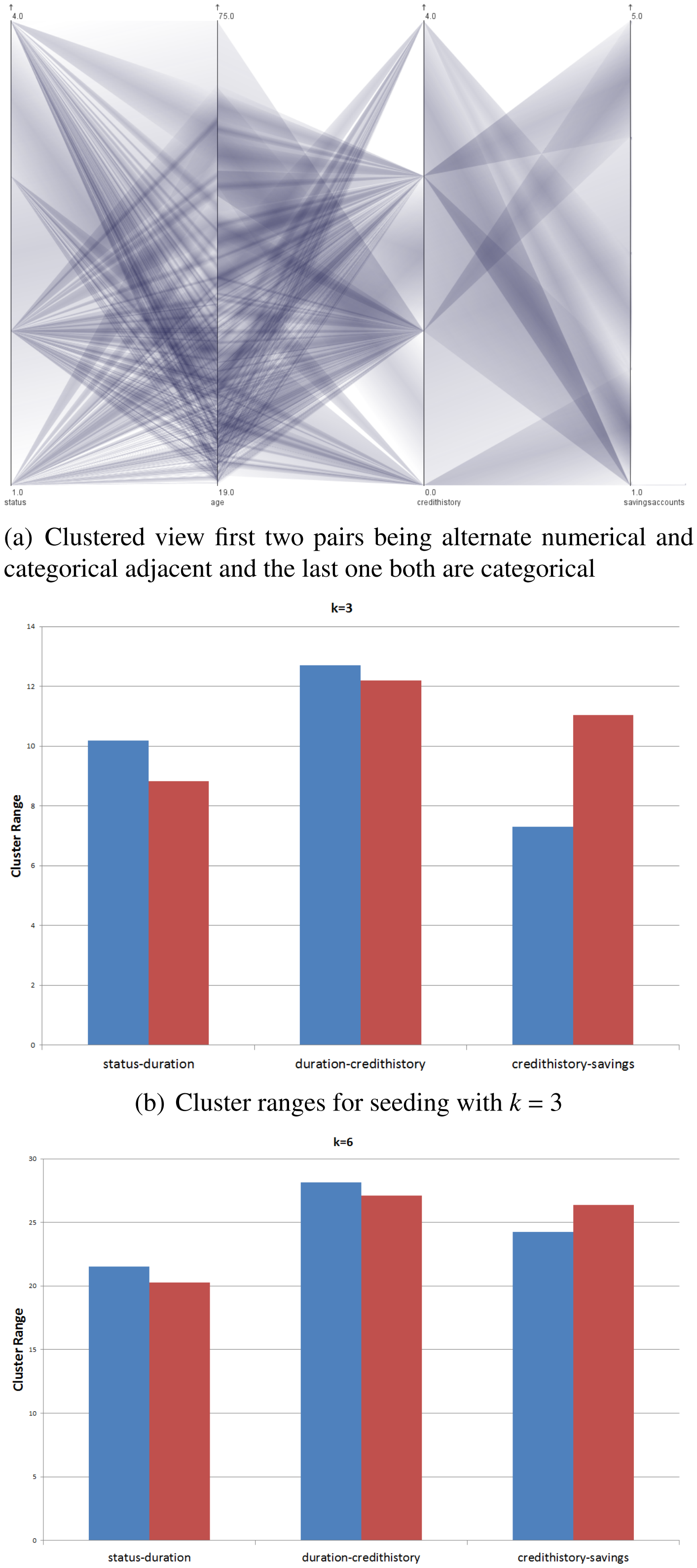
2. In case of both categorical axes use over-plotting degree as the criterion, for a numerical and categorical adjacency use the degree of convergence/divergence as the criterion.

3. Compute degree of convergence and divergence for both axes.

4. Convergence and divergence are mirrors of each other. If convergence is greater than divergence, use convergence as the basis for selecting seed, else use divergence. In case of the former, the right dimension is the seeding dimension and in case of the latter, the left dimension is the seed dimension.

5. If both are numerical dimensions check which one of convergence/divergence and over-plotting is greater. Choose that metric to pick the highest frequency bin from which the record is chosen.

6. When the values of either over-plotting degree or degree of convergence/divergence



(a) Clustered view first two pairs being alternate numerical and categorical adjacent and the last one both are categorical

(b) Cluster ranges for seeding with $k = 3$

(c) Cluster ranges for seeding with $k = 6$

Figure 26: Different seed-selection criteria like over-plotting (blue bars) and convergence/divergence (red bars) have a significant effect on the cluster ranges. Lower cluster ranges obscure less of the data and thus help perceive the different trends and patterns that exist between axes.

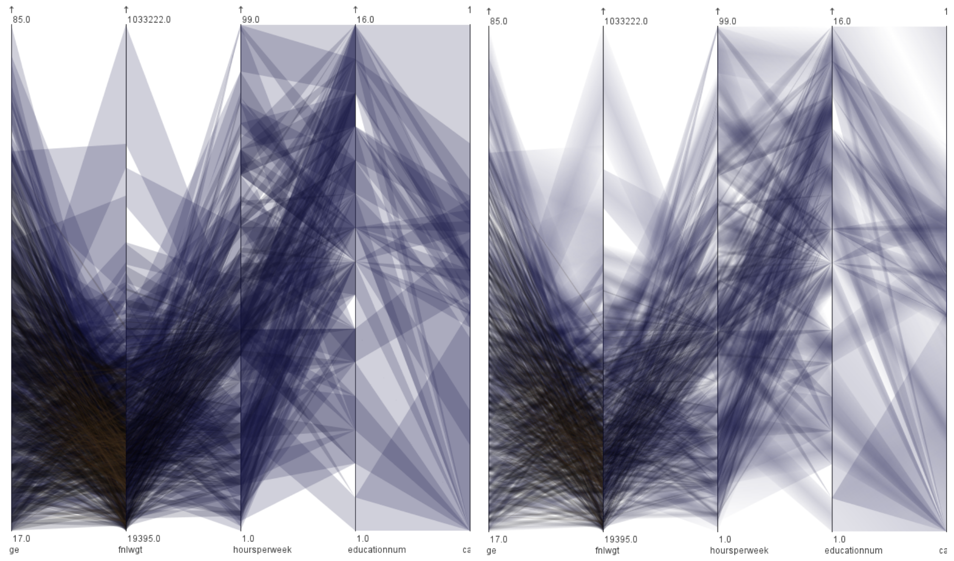
is equal to 1 for all the bins, we build a histogram hierarchically by combining bins that are in the nearest neighborhood: if b adjacent bins have a value of 1, then we put b values in a bin. We select a record from the highest frequency bin in the new histogram. Following this method, we get a coarser histogram and this enables us to avoid selecting a record from a bin which has a sparse neighborhood.

5.3.2 Clustering

After choosing the initial seed, the clustering algorithm searches for the best record and adds it to the current cluster until the threshold value k is reached. Our method departs from the original algorithm in two ways: a) choice of a distance metric and b) locality-preserving clustering.

Choice of a distance metric: The original algorithm uses an information loss metric based on generalization hierarchy of the attributes as the distance function [24]. Earlier we have argued that a purely data-based clustering approach like this does not work well for visualization (Figure 22). Instead we use the Manhattan distance as the cost function as the goal here is to find visually similar records. The Manhattan distance metric allows us to minimize the vertical distance between the lines on the axes. Manhattan distance translates directly to what we observe as cluster size on the axes.

Locality-preserving clustering: Instead of multi-dimensional clustering, we employ axis-pairwise clustering that takes just local properties into account. This helps to retain the local features between adjacent axes leading to smaller and more discernible clusters and thus less occlusion. In the initial iteration of the clustering, records are grouped into $\lfloor \frac{n}{k} \rfloor$ clusters based on the seeds we chose earlier. After that, there are still $n \bmod k$ records left.



(a) Rendering without gradient

(b) Rendering with gradient

Figure 27: Rendering without gradient(top) tends to produce more clutter than rendering with gradient(bottom)

Those are added to existing clusters following the same initial steps, this time minimizing the cost function for a particular cluster and adding it to the cluster which incurs minimum cost. The process is repeated for each axis pair.

5.3.3 How Seeding Criteria Affect Clustering

We choose cluster range as a measure for visual quality of the clusters. Cluster range is measured as the sum of the number of pixels spanned by the records in a cluster on each axis. Figure 26 shows a clustered parallel coordinates configuration with three categorical axes and one numerical axis. The bar graphs show that, in case of both categorical dimensions, cluster ranges for seeding with over-plotting degree are lower than that with degree of convergence/divergence. In case of a numerical-categorical adjacency, the degree of convergence/divergence gives lower cluster ranges. We have observed that the choice of axis is critical in producing smaller clusters, and a wrong choice leads to larger cluster ranges and therefore more occlusion.

5.3.4 Rendering

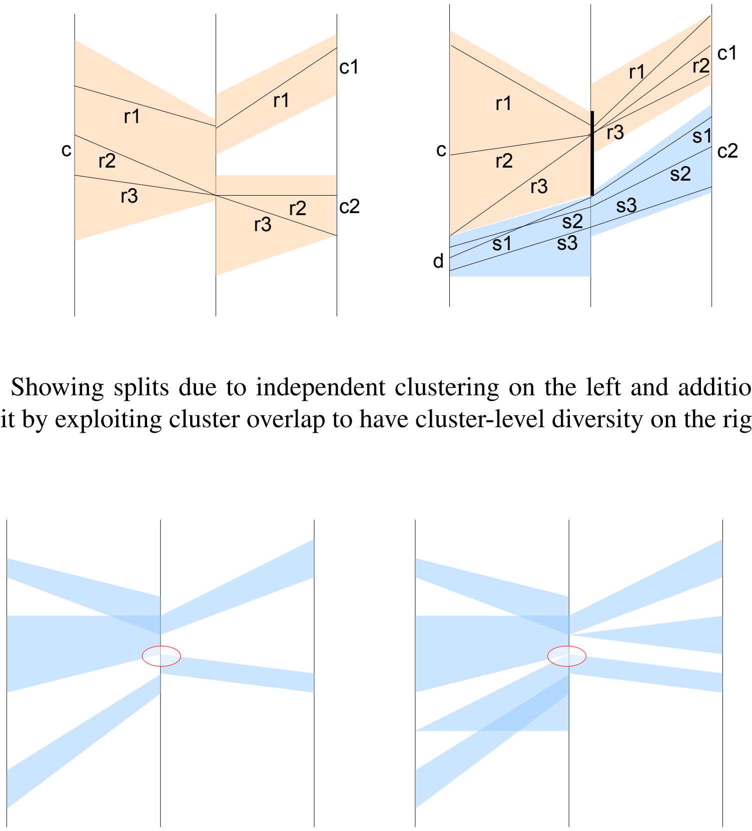
An issue arising in the rendering of the clusters is that they create a large number of visual artifacts. In addition to the sheer clutter from many overlapping clusters, it is difficult to tell exactly how many clusters are overlapping at each point. Sharp edges of the clusters create a large amount of visual noise that also makes the display harder to read.

We originally used a depth-from-color effect [93], where we ordered the clusters by size and drew the largest ones first. That helps somewhat, and especially makes smaller clusters stand out (which are more relevant, because they provide more specific information). But the clutter and noise issues remain.

Based on previous ideas of how to draw clusters in parallel coordinates, in particular Fua et al.'s hierarchical parallel coordinates approach [54], we developed a different way of rendering the clusters that does not use sharp edges (Figure 27). Unlike Fua et al., we do not indicate the cluster centroid with a line. Rather, the color's alpha channel varies perpendicular to the cluster's main direction (the direction of the centroid), creating a fuzzy boundary. Scaling the same range of alpha values over larger clusters produces more fuzziness, while smaller clusters appear sharper. The overall effect of many overlapping clusters is similar to splatting [128].

5.3.5 Cluster Diversity

k-anonymity ensures record-level privacy which is a necessary but not sufficient condition for privacy protection. The *k*-anonymity method is susceptible to the homogeneity problem, where a cluster based on quasi-identifiers can have the same values for the sen-



(a) Showing splits due to independent clustering on the left and additional split by exploiting cluster overlap to have cluster-level diversity on the right.

(b) Cluster splits for different configurations of data. On the left: Splits at the edge compromising privacy with $k = 2$: In the circled area, we know there is a distinct value at that point leading to attribute disclosure. With $k > 2$ it gets difficult to guess the value because that data point can belong to multiple clusters which overlap at that point.

Figure 28: Demonstrating cluster splits and overlap.

sitive attribute and thus the value of the sensitive attributes can be guessed. Therefore, we apply the concept of l-diversity [82] as a constraint for filtering the clusters that are highlighted on interaction. A privacy-preserving visualization technique differs from its data-mining counterpart because of the added challenge of efficiently handling different interaction conditions. We address this in parallel coordinates by adapting the l-diversity condition to the dynamic user interaction.

5.3.5.1 Cluster Splits

An artifact of independent clustering between adjacent axes is that clusters are discontinuous and they appear to get split when highlighted as shown in Figure 28(a). On the left we see that the records $r1$ and $r2$ in cluster c are contained in cluster $c1$ on the adjacent axis while the record $r3$ is contained in the cluster $c2$. When c is selected by the user, both $c1$ and $c2$ get highlighted. These splits add to the uncertainty in guessing the exact value of a record.

5.3.5.2 Added Perturbation

A cluster can also be continuous as shown on the right in Figure 28(a), where all $r1$, $r2$ and $r3$ are contained in the same cluster $c1$. In that case, there is no split. But to apply the l-diversity constraint between a quasi-identifier and sensitive dimension, we need a cluster to split into at least l different clusters on the sensitive dimension. For this we use the overlapped pixels on a particular axes (Figure 28(a)) by clusters from adjacent dimension and highlight l different clusters. In this case, $l = 2$, and the cluster $c2$, that is also highlighted on selection of c, is actually a continuity of cluster d. If there are no overlaps, we do not show the clusters on the sensitive dimension. Effectively, we add some random noise to the clusters and alter the actual data values that are perceived. Although this perturbation lowers utility, this is a necessary step to protect the sensitivity of the data values.

5.3.5.3 Adaptive *l*-diversity

A dimension, on the whole, might not be sensitive, but some of the values can be. In the German Credit dataset [52], we have four different values for the sensitive dimension, of which only the value 4 (bad credit) is deemed sensitive by the data owner. The *l*-diversity constraint is only applied in case of a cluster that has a record with that data value (Figure 49). Another example of this scenario would be disease datasets, where cold and flu might not be considered sensitive by a data owner or an individual, but diseases like cancer are sensitive and need privacy-preservation.

A couple of special cases arise when i) a sensitive dimension has only two values, for example a class variable with a binary yes or no; so in that case a cluster can only be 2-diverse and ii) there are *n* different values on the sensitive dimension and a cluster has to be $n-diverse$. In these cases we do not show any of the highlighted clusters between the quasi-identifier and sensitive dimension. This reduces the utility of the resulting visualization, but imposing that restriction is critical from a privacy-preserving perspective.

5.4 Measuring Privacy and Utility Based on Visual Uncertainty

In this section we conceptualize the relationship between privacy and visual uncertainty in the context of position-based representations like scatter plots, parallel coordinates, etc. We refer to malicious users with intention of privacy breach as *attackers*, and outline our assumptions regarding their background knowledge about the data in course of our description of the metrics.

5.4.1 Applying the Visual Uncertainty Taxonomy

Visual uncertainty can be decomposed into a set of encoding and decoding uncertainties, according to the visual uncertainty taxonomy [34]. This taxonomic approach offers an opportunity to identify the causes, effects and sources of uncertainty that can be related to privacy and utility of a visualization. In Table 1 we tie these different elements together. In course of our ensuing discussion we refer to the various levels of the taxonomy tree proposed in that work.

Binning and clustering are the basic elements of our privacy model. For encoding uncertainty, one option can be hiding sensitive data values which would lead to completeness uncertainty at the data mapping stage. However, since the goal is to minimize information loss, we do not consider that option. Instead, we focus on quantifying uncertainty introduced at the visual mapping stage, since privacy-preservation is based on screen-space properties. Encoding uncertainty in the form of precision and granularity are introduced due to binning and clustering. These are causes of intended uncertainty, affect the static visual representation of the clusters and are not influenced by interaction.

The components of decoding uncertainty affect how an attacker is able to gain information by using interaction. Cluster overlaps cause identity uncertainty and and splits cause traceability uncertainty (in parallel coordinates). These are related to both privacy and utility. For example, if an attacker knows the existence of a data value on one of the dimensions, and tries to guess the values for the other dimensions, then cluster overlaps help in creating identity uncertainty and making privacy breach difficult by hiding the cluster membership. However, too many overlaps create clutter and therefore make effective perception of patterns more difficult. Thus decoding uncertainty includes both intended and unintended forms of uncertainty. Metric-based analysis of visual uncertainty helps quantify these different forms and design a privacy-preserving visualization that balances these trade-offs.

5.4.2 Disclosure Risks and Attack Scenarios

Two types of re-identification types have been identified in the literature [45, 73]: a) identity disclosure: this occurs if the intruder is able to assign a particular identity to any record in the data and b) attribute disclosure: this occurs when an intruder learns something new about the sensitive attribute values that can be linked to an individual, without knowing which specific record belongs to that individual. For example, if all 30-year-old male patients in the disclosed database who live in a particular area had a prescription for diabetes, then if the intruder knows that John is 30 years old and lives in that particular area, he or she will learn that John is a diabetic, even if the particular record belonging to John is unknown. Identity disclosure typically needs external information like quasi-identifiers, so medical databases try to guard against this type of disclosure. Attribute disclosures are likely to occur in corporate data, which are not generally associated with external information.

Two ways to break the privacy of a sanitized dataset suggested in the literature [70] also apply in case of visualization. The first one is called *prosecutor re-identification scenario*, where an intruder (e.g., a prosecutor) knows that a particular individual (e.g., a defendant) exists in an anonymized database and wishes to find out which record belongs to that in-

Type of Visual Uncertainty	Disclosure Risk	Visual Artifacts
Granularity	Number of records per cluster Connections among data points within the cluster	Cluster range Cluster configurations (Section 5.7.2)
Spatial Accuracy	Exact coordinates of data points	Cluster range (Section 1)
Identity	Confirmed existence of a data-point Cluster membership of a record	Cluster overlaps (Section 5.6.3)
Traceability	Trace records within clusters across multiple axes.	Cluster splits (Section 5.7.1)

Table 4: Connecting causes and effects of uncertainty to disclosure risks and visual artifacts.

dividual. In the second one, known as the *journalist re-identification scenario*, an attacker tries to re-identify an arbitrary individual. The intruder does not care which individual is being re-identified, but is only interested in being able to claim that privacy breach is possible.

5.5 Mapping Disclosure Risks to Visual Uncertainty

Design and analysis of visualization techniques have commonly been studied based on two factors: analytical tasks that a user performs [102] and visual quality that needs to enhanced to convey information effectively [16]. In addition to these factors, another focal point of a privacy-preserving visualization is preventing disclosure of sensitive information through the visual structures. In this section we argue how disclosure risks related to privacy can be mapped to the causes and sources of visual uncertainty in the screen-space.

In visualization, at least some information about the data is typically available, like labels and value range on axes, and the minimum and maximum boundaries of each cluster. The notion of totally 'blind' attack, without any knowledge about the data, may not be applicable to privacy-preserving visualization. Furthermore, an attack usually consists of a series of progressive actions, building on incrementally acquired knowledge. An attacker may start with little knowledge, and by making observations from the information conveyed in visualization, the attacker may identify a possible end point of an arbitrary line in a cluster. From that, the attacker gradually identifies more information about the line by moving from one axis to another, or works out information about other lines in the same cluster. Regardless of how complex an attack is, it can be decomposed into a set of basic attacking actions, such as: determining the number of data points in a cluster, finding connections among them, determining if a specific record is in a particular cluster, finding coordinates of a two-dimensional record, and so on. It is not difficult to combine such basic attacking actions with acquired knowledge to enable complex attack actions, for instance, given a disclosed value of a record (i.e., an end point on an axis), find all other values of this record (i.e., all end points of this line; or given a disclosed line in a cluster, find all other lines in the cluster. We assume that an attacker is knowledgeable about parallel coordinates and scatter plots, and may have some interactive analytical and visualization tools to aid his/her activities.

5.5.1 Visual Uncertainty in the Context of Privacy-Preserving Visualization

Privacy-preserving clustering exhibits inherent visual uncertainty that can be utilized for defending against potential attack scenarios. In Table 4 we summarize the causes of visual uncertainty that correspond to the privacy issues and connect them to the visual artifacts that are the sources of uncertainty and act as an additional layer of protection besides privacy-preserving clustering against possibilities of disclosure. Granularity uncertainty is related to number of elements per cluster and the connections among them. We assume that an attacker has knowledge of k, that is the number of data points per cluster. The sources of granularity uncertainty are i) cluster ranges on an axis-pair that hide the precise location of the data points and ii) cluster configuration that hides the connection among the points. Because of the different possibilities of cluster configurations depending on different values of *k*, an attacker has to overcome this form of uncertainty to gain knowledge about the data. In many cases, using interaction an attacker can know about a cluster configuration (Section 5.7.2). Knowing a cluster configuration in many cases would not reveal the precise location of the end points. Accurately guessing the non-edge coordinates of records inside the cluster would mean that the attacker has to work out a number of combinatorial cases to overcome the lack of spatial accuracy due to the pixel resolution of the cluster range (Section 1). Even if cluster edges comprise of real data points, many clusters overlapping at the same point can cause identity uncertainty. Moreover, even if the attacker knows that a certain data point or a record exists in the database, overlaps can make it difficult for him/her to identify which cluster that entity belongs to and also trace the path of the record across different axis-pairs in parallel coordinates (Section 5.6.3).

5.5.2 Connecting Privacy, Utility, and Visual Uncertainty

Informations is a measure of the decrease of uncertainty for the receiver of a message [100]. If visualization is viewed as a communication channel from the data space to the perceptual and cognitive mental space of the user [94], it is important to trace the uncertainty along different stages of the pipeline, so that the information communicated to the user can be optimized. In case of privacy-preserving visualizations, some forms of uncertainty would be intended, to prevent disclosure of sensitive information. In general, increasing the amount of *visual uncertainty* in a visualization will increase *privacy* of the visualization while decreasing its *utility*. In other words, privacy and utility are functions of visual uncertainty.

Let $u_1, u_2, \ldots, u_p \in [0,1]$ be the quantities corresponding to a set of measurable uncertainties in a visualization, with 0 being most certain and 1 being most uncertain. Let us define two approximated measurements for privacy (m_p) and utility (m_t) of the visualization as:

$$m_p = \sum_{1}^{p} u_i, \quad m_t = \sum_{1}^{p} (1 - u_i) \tag{2}$$

In general, m_p and u_i are positively correlated, while m_t and u_i are negatively correlated. In this work we propose different measures related to u_i that address the different causes and effects of uncertainty as categorized by the taxonomy. Here we use the term “measure” in a broad sense, including both computational measurement and human-centered quantitative evaluation. In practice, privacy and utility may also be affected by factors other than visual uncertainty, such as the environment where the visualization is used. The above measurement, m_p and m_t should be used only for comparing visualization with different forms of anonymization while those other factors remain unchanged.

5.5.3 Role of Metrics

For some forms of visual uncertainty, like that due to pattern complexity, screen-space metrics already exist, like Scagnostics for scatter plots [120], Pargnostics for parallel coordinates [37], etc. However, metrics for other types of uncertainty are missing in the current literature. Some of the metrics that we propose [35] are applicable beyond the context of privacy, where the issue of disparity between the large number of data points and limited number of pixels arise.

Encoding uncertainty serves as the initial defensive mechanism against attackers with no background knowledge about the data. Loss of precision due to binning and high-level of granularity due to clustering make it difficult for an attacker to guess the exact value and number of data points within a cluster. To quantify these, we have developed the *cluster range metric* and the *cluster summary error metric*. The cluster range metric also captures the decoding uncertainty involving spatial accuracy for guessing the location of the data points within a cluster. Encoding uncertainty cannot be reduced by using interaction.

When an attacker has some background knowledge about the data, the different components of decoding uncertainty help in confusing the attacker. When an attacker knows about the existence about a particular data-point in the database, the process of privacy breach starts by associating a data point with a cluster. Identity uncertainty due to cluster overlaps make that association difficult. Overlaps also lead to clutter, where identifying paths of clusters itself is difficult. We quantify the privacy aspect of identity uncertainty through the *overlap entropy* metric and the utility aspect dealing with clutter, through the *overlap clutter* metric. For line-based parallel coordinates, traceability of lines across multiple dimensions is an advantage over scatter plots. However, in privacy-preserving parallel coordinates, due to axis pair-wise clustering, clusters appear to split across axis. This leads to uncertainty due to lack of traceability, that we capture through our *average split count* metric. For controlling the uncertainty due to pattern complexity we use the Pargnostics metrics and also compute the mutual information between adjacent dimensions. These are essential for choosing the best adjacency configurations for parallel coordinates and also point out the pairs of axes with high utility in scatter plots.

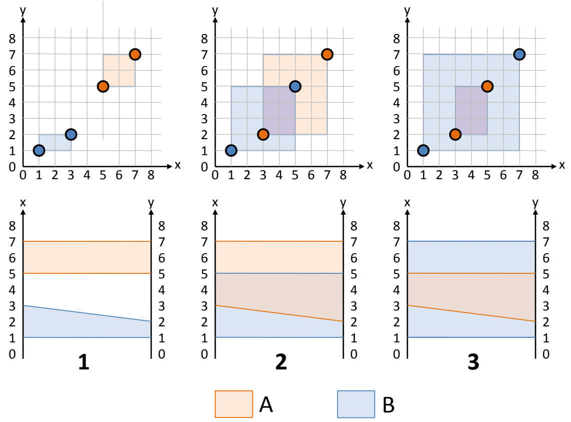


Figure 29: Pixel-based binning and clustering in parallel coordinates and scatter plots. Illustrating different ways for 2-anonymizing the four data points in a pixel grid. For a much larger number of points, estimation of privacy requires metrics.

5.6 Metrics for Uncertainty Measurement

In the following section we introduce the metrics for measuring the different types of uncertainty. We describe the type of uncertainty measured by each metric and quantitatively describe each with illustrations and examples from the *Diabetes* dataset that are described in detail in Section 6.8.

5.6.1 Cluster Summary Error

The further an actual record is perceived to be located from its actual position, the more difficult it would be to precisely guess the value of a record. In case of pixel-based representation, binning already introduces loss in precision due to quantization error. A cluster-based representation accentuates the error: the further away a record is from the cluster centroid, the more difficult it will be for knowing the exact value. We measure the summary error for privacy-preserving clustering as the Manhattan distance between the its actual pixel coordinate and the pixel coordinate of the cluster centroid. This is similar to the

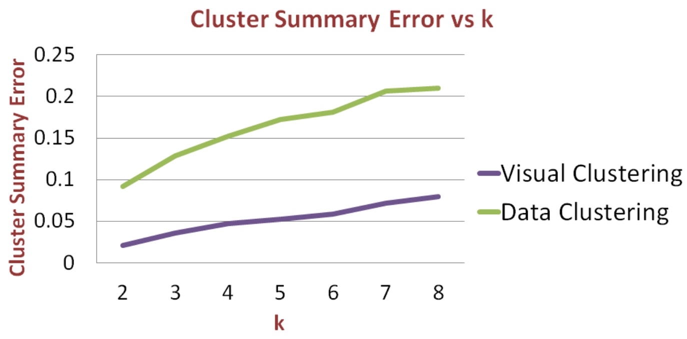


Figure 30: Average cluster summary error for all dimensions of the Diabetes dataset that are shown to increase monotonically with increasing k.

class consistency measure [103] for selecting optimal views of the data using scatter plots. Although in case of cluster-based representation this is not directly reflected in what the user sees on screen, the metric gives a quantitative measure of precision uncertainty.

Consider a cluster C_t consists of n_l records. It intersects with an axis, spanning over several pixel bins, $a_t, a_t + 1, \ldots, b_t - 1, b_t$ where $0 \leq a_t \leq b_t \leq h - 1$, where h is the total number of pixels of this axis. The centroid of the intersected section is thus:

$$\eta_t = \frac{a_t + b_t}{2}$$

The error of this intersection ε_t can be defined as follows:

$$\varepsilon_t = \frac{1}{n_l h} \sum_{i=1}^{n_l} |s_i - \eta_t| \tag{3}$$

The average error over all clusters if given by:

$$\varepsilon = \frac{1}{n_c} \sum_{i=1}^{n_c} \varepsilon_t \tag{4}$$

where s_i is the actual mapped pixel coordinate of record R_i on that axis.

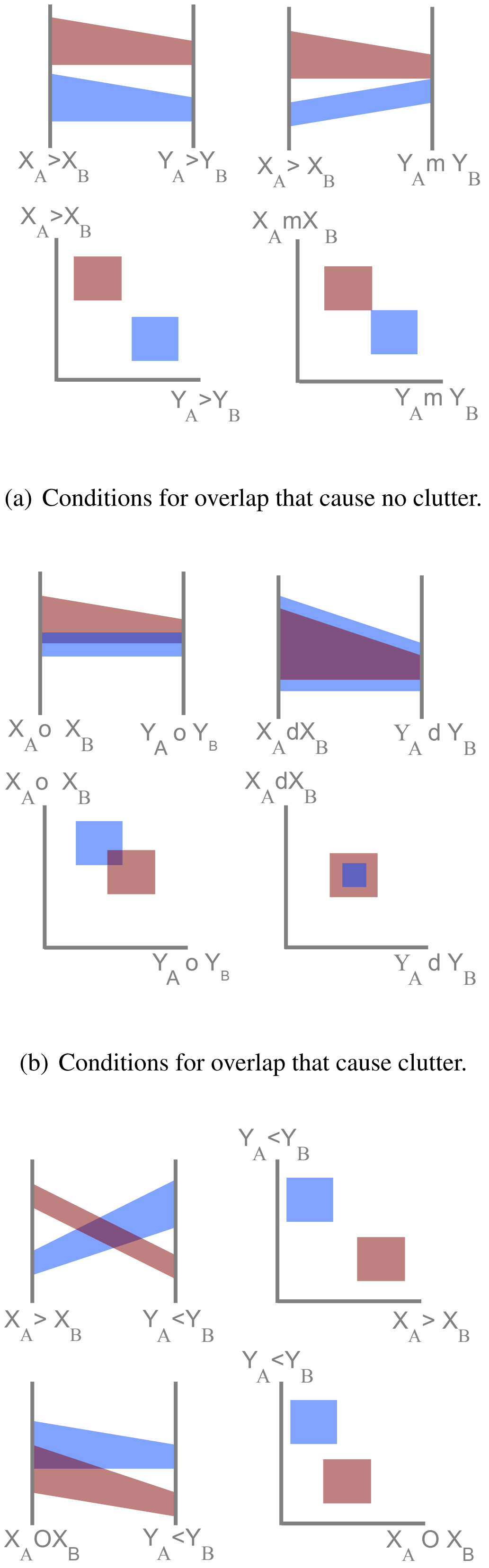
We compare the cases in Figure 29 for the cluster summary error metric. For x-axis, we have: $\epsilon_{A1} = 0.056$, $\epsilon_{B1} = 0.056$; $\epsilon_{A2} = 0.222$, $\epsilon_{B2} = 0.222$; and $\epsilon_{A3} = 0.111$, $\epsilon_{B3} = 0.333$. For y-axis, we have: $\epsilon_{A1} = 0.056$, $\epsilon_{B1} = 0.056$; $\epsilon_{A2} = 0.278$, $\epsilon_{B2} = 0.222$; and $\epsilon_{A3} = 0.167$, $\epsilon_{B3} = 0.333$.

Therefore configuration 1 is less private than configurations 2 and 3 on both axes.

As shown in Figure 30 the average cluster summary error is much higher in case of data-based clustering. This coincides with higher cluster ranges which leads to clutter and reduces the visual quality. Information loss in terms of precision of data values is also much higher in this case.

5.6.2 Cluster Range

Cluster ranges on the axes mask the precise location of the data points. A cluster in the data-space is perceived in terms of the number of record it contains. In the screen space, it is perceived in terms of the number of pixel bins covered by the cluster on the axes, which we define as cluster range. When the analysts has no background knowledge about the data and tries to randomly guess if data points exist or not, cluster ranges lead to granularity uncertainty and that due to spatial accuracy. Larger a range is, the less accurately one can estimate the value of any record within this range, and at the same time, more likely it will cause overlapping among clusters. Though a cluster range can be perceived as both a privacy and utility metric, since its primary role is masking data values and the uncertainty is intended, we consider cluster range as a privacy metric. Unlike cluster summary error, cluster range is independent of the number of records in the cluster or their individual values.



(a) Conditions for overlap that cause no clutter.

(b) Conditions for overlap that cause clutter.

(c) Conditions for overlap that cause clutter in parallel coordinates but not in scatter plots.

Figure 31: Illustrating difference in effects of cluster overlaps for scatter plots and parallel coordinates. The red cluster is represented by *A* and the blue cluster by *B*.

	$Y_A > Y_B$	$Y_A < Y_B$	$Y_A m Y_B$ $Y_B m Y_A$	$Y_A s Y_B$ $Y_B s Y_A$	$Y_A f Y_B$ $Y_B f Y_A$	$Y_A = Y_B$	$Y_A o Y_B$ $Y_B o Y_A$	$Y_A d Y_B$ $Y_B d Y_A$
$X_A > X_B$	N	B	E	OB	OB	OB	OB	OB
$X_A < X_B$	B	N	E	OB	OB	OB	OB	OB
$X_A = X_B$	OB	OB	OB	OB	OB	OB	OB	OB
$X_A m X_B$ $X_B m X_A$	E	EB	EB	OB	OB	OB	OB	OB
$X_A s X_B$ $X_B s X_A$	OB	OB	OB	OB	OB	OB	OB	OB
$X_A f X_B/$ $X_B f X_A$	OB	OB	OB	OB	OB	OB	OB	OB
$X_A o X_B$ $X_B o X_A$	OB	OB	OB	OB	OB	OB	OB	OB
$X_A d X_B$ $X_B d X_A$	OB	OB	OB	OB	OB	OB	OB	OB

Table 5: Cluster overlaps depending on the relationship of the clusters (A and B) on the axes (X and Y)in parallel coordinates. **N**: No overlap either on the axes or between the axes; **B**: Overlap only between axes; **OB**: Overlap between as well as on the axes; **E**: Meeting at the edge,**EB**: Meeting on the axes and overlap between axes.

Consider a cluster C_t. Its intersection with the axis spans between pixel coordinates a_t and b_t, where The normalized range of this cluster is thus $(b_t - a_t)/(h-1)$. We can define an axis-based metric as the average range of all clusters intersecting with the axis as:

$$\gamma = \frac{1}{n_c(h-1)} \sum_{t=1}^{n_c} (b_t - a_t) \tag{5}$$

where n_c is the total number of clusters.

We compare the cases in Figure 29 for the cluster range metric. For x-axis, we have: $\gamma_{A1} = 0.125$, $\gamma_{B1} = 0.125$; $\gamma_{A2} = 0.5$, $\gamma_{B2} = 0.5$; and $\gamma_{A3} = 0.25$, $\gamma_{B3} = 0.75$. For y-axis, we have: $\gamma_{A1} = 0.125$, $\gamma_{B1} = 0.125$; $\gamma_{A2} = 0.625$, $\gamma_{B2} = 0.500$; and $\gamma_{A3} = 0.375$, $\gamma_{B3} = 0.750$. Configurations 2 and 3 are therefore more private than 1.

	$Y_A > Y_B$	$Y_A < Y_B$	Y_AmY_B Y_BmY_A	Y_AsY_B Y_BsY_A	Y_AfY_B Y_BfY_A	$Y_A = Y_B$	Y_AoY_B Y_BoY_A	Y_AdY_B Y_BdY_A
$X_A > X_B$	N	N	N	N	N	N	N	N
$X_A < X_B$	N	N	N	N	N	N	N	N
$X_A = X_B$	N	N	O	O	O	O	O	O
X_AmX_B X_BmX_A	N	N	E	O	O	O	O	O
X_AsX_B X_BsX_A	N	N	O	O	O	O	O	O
X_AfX_B X_BfX_A	N	N	O	O	O	O	O	O
X_AoX_B X_BoX_A	N	N	O	O	O	O	O	O
X_AdX_B X_BdX_A	N	N	O	O	O	O	O	O

Table 6: Cluster overlaps depending on the relationship of the clusters (A and B) on the axes (X and Y)in scatter plots. The notations are the same as in parallel coordinates except for the case *B* as there is no distinction between overlap between and on the axes in scatter plots.

5.6.3 Overlap Clutter

Cluster overlaps lead to identity and traceability uncertainty in perceiving the path of the clusters, therefore leading to clutter. In line-based parallel coordinates, the vertical distance between the start and end points of a line on adjacent axes can be treated as intervals [5] to determine when lines cross [37]. In the case of cluster-based parallel coordinates and scatter plots, we treat the cluster ranges on each axis as intervals for detecting cluster overlaps. Allen's interval algebra defines 13 possible cases between two intervals, X and Y: X before Y, X starts Y, X ends Y, X meets Y, X during Y, X overlaps Y, and X equals Y. All but the last condition also have a symmetrical case. Given the 13 cases between the two clusters on each axis, we have to investigate $13 \times 13 = 169$ possible cases.

For parallel coordinates there are four possible pairwise relationship between cluster ranges: no overlap (N), meeting at the edges (E); overlap on and between axes (*OB*) and meeting at the edges and overlap between axes (EB). In fact, all cases where overlap happens on the axes also means that there is overlap between the axes. For scatter plots on the other hand, there is no distinction between an overlap *on* the axis and that *between* the axes, because the coordinate positions can be anywhere between the axes. The possible relationships between cluster ranges in scatter plots, are therefore, N, E, and O. A simple enumeration of the $13 * 13$ conditions enables us to draw a distinction among these overlap cases. The different possibilities for parallel coordinates are shown in Table 5 and those for scatter plots are shown in Table 6. The symmetrical conditions are shown together except for the greater than (>) and less than (<) condition as there is a distinction between parallel coordinates and scatter plots in this case.

Based on these conditions we can derive three cases that are relevant for clutter: A) overlap conditions that do not lead to clutter in either parallel coordinates or scatter plots (Figure 31(a)), B) overlap conditions that lead to clutter in both parallel coordinates or scatter plots (Figure 31(b)), and C) overlap conditions that lead to clutter in parallel coordinates but not in scatter plots (Figure 31(c)). For A, the conditions are either 'before/after' or 'meets' on both axes. For B, the conditions are 'overlaps', 'starts/finishes' and 'during on both axes.

One distinction between parallel coordinates and scatter plots is when there is a 'before/after' condition on one axis and a different condition on the other. This is reflected in a much higher number of N in Table 6 for scatter plots than parallel coordinates. In these cases clusters overlap between axes in parallel coordinates but there is no perceptual overlap in scatter plots. These conditions are the basis for our overlap clutter metric.

For the clusters, the upper bound for the number of overlaps is $\frac{n_c(n_c-1)}{2}$. Therefore we

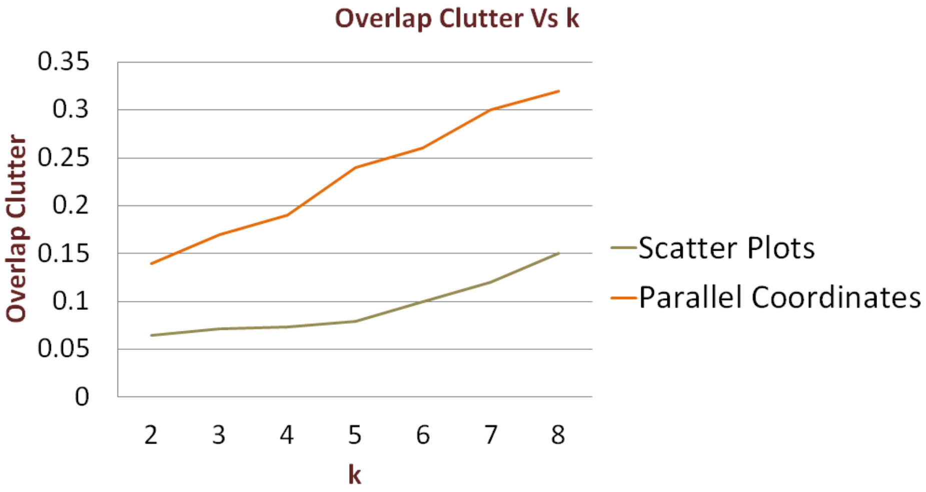


Figure 32: Clutter in privacy-preserving scatter plots is lower than parallel coordinates due to the lesser number of overlaps.

compute clutter in parallel coordinates (C_P) as:

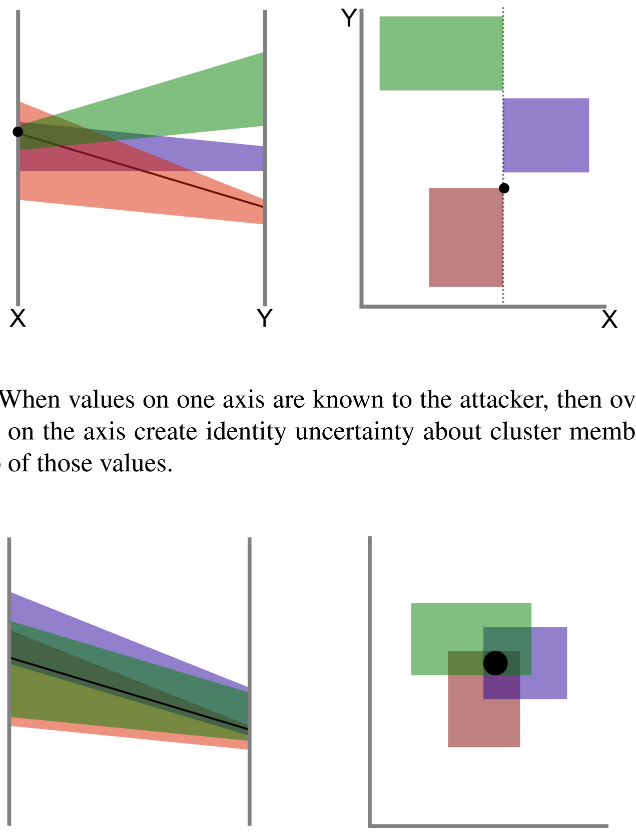
$$C_P = \frac{2n_o}{n_c(n_c - 1)} \tag{6}$$

where n_o is the total number of overlaps in parallel coordinates or scatter plots. For scatter plots we denote clutter by C_S. The difference between C_P and C_S for different cluster sizes, i.e. k is shown in Figure 32. Evidently, $C_P > C_S$, i.e., identity and traceability uncertainty are lower in scatter plots than parallel coordinates because of the lesser number of overlaps for different values of k.

5.6.4 Overlap Entropy

The previous metrics do not take any possible background knowledge of an attacker into account. If an attacker knows which cluster a data point belongs to, then the privacy breach becomes easier, than the case when there is identity uncertainty regarding associating a data point with a cluster.

Overlapping cluster ranges on the axes lead to uncertainty because of difficulty in know-



(a) When values on one axis are known to the attacker, then overlaps on the axis create identity uncertainty about cluster membership of those values.

(b) When values on both axes are known to the attacker, then overlap on both axes create identity uncertainty. Large number of such overlaps also reduces the mutual information between adjacent axes.

Figure 33: Illustrating how overlaps on one axis and that on both axes lead to uncertainty.

ing which cluster a pixel bin belongs to and consequently, tracking the clusters across different axis-pairs. When certain data values are known to attackers, overlaps help in creating uncertainty about cluster membership of a data point as illustrated in Figure 33(a). We use an information theoretic measure in the form of Shannon's entropy to quantify the uncertainty in tracking precise membership of a bin in cluster. This is similar to the privacy metric based on entropy suggested by Agrawal et al. [2] and Bertino et al. [18].

Consider n_c clusters on an axis in a privacy preserving parallel coordinates visualization. The axis has h pixel bins. Each bin may intersect with zero, one or several clusters, while each cluster may span over one or more bins. As identifying an empty bin is trivial, the uncertainty is thus associated with those bins that intersect with one or more clusters.

Assume that the attacker has no a priori knowledge about any cluster, so the probability of making a correct guess of the association between a bin and a cluster is independent and identically-distributed.

Let α_i be the number of clusters intersect with bin β_i, where $0 \leq i \leq h-1$.

Given a cluster, C_t, the probability mass function for identifying this cluster at bin β_i is thus

$$P(t@i) = \begin{cases} 0 & \text{if } C_t \text{ does not intersect with } \beta_i \\ 1/\alpha_i & \text{if } C_t \text{ intersects with } \beta_i \end{cases}$$

The entropy in relation to cluster C_t is thus the following sum computed over all non-zero $Pt@i$.

$$H_t = -\sum_{i=0}^{h-1} P(t@i) \ln P(t@i)$$

We can compute an information-theoretic measure of uncertainty of the axis as

$$\Phi = \frac{\sum_{t=1}^{n_c} H_t}{n_c H_{max}} \tag{7}$$

H_{max} is the maximum entropy value for H_t, which is associated with a situation where every cluster spans over every pixel bin, that is, $P(t@i) = 1/n_c$ for every cluster and every bin. The lower H_t is, the lower information it contains about C_t, and thus higher privacy. From empirical results, we have observed that the absolute value of entropy increases with increasing k. Since with increasing k there are more overlaps, we have more uncertainty in the screen-space. When we compare with data-based clustering, the entropy value is lower as compared to visual clustering, because of less number of overlaps among clusters (Figure 35b).

5.6.5 Mutual Information

When an attacker knows both coordinates of a two-dimensional data point, then uncertainty due to overlaps is only caused when overlaps are on both axes. As shown in Figure 33(b), these types of overlaps reduce the mutual information between the two adjacent axes. We consider the mutual information as an utility metric and this metric is important for handling interaction scenarios like reordering axes. The mutual information, a measure of the reduction in uncertainty of one variable due to the knowledge of the other, needs to be maximized for utility purposes.

The general formula for the mutual information between two random variables X and Y is

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \ln \frac{P(x,y)}{P(x)P(y)} \tag{8}$$

Where $P(x,y)$ is the joint probability of x and y, and $P(x)$ and $P(y)$ are the marginal prob-

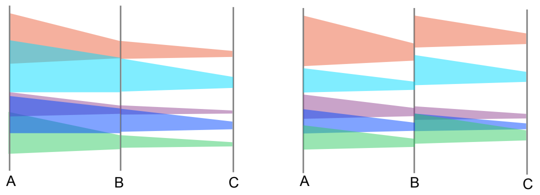


Figure 34: In case of multi-dimensional clustering,on the left, there is no traceability uncertainty on brushing, as clusters are continuous. In case of axis pairwise clustering, on the right, traceability uncertainty of an axis pair depends on the average number of split cluster on the adjacent axis.

abilities. For a record with values $x_{i,j}$ and $x_{i,j+1}$ on adjacent axes on a parallel coordinates plot, the joint probability is equal to the uncertainty of that record's exact location, which is determined by the number of clusters that contain it. Consider two axes, x and y. Given a specific cluster C_t, we can compute the probability mass functions $P(t@x_i)$ and $P(t@y_j)$ as in the previous section. The joint probability mass function can be defined as:

$$P(t@x_i, t@y_j) = \begin{cases} 0 & \text{if condition (i)} \\ \\ \frac{1}{\alpha_{x,i}\alpha_{y,j}} & \text{if condition (ii)} \end{cases}$$

where condition (i) is when C_t does not intersect with the i^{th} bin on x-axis or the j^{th} bin on y-axis; and condition (ii) is when C_t intersects with both the i^{th} bin on x-axis and the j^{th} bin on y-axis. We can thus compute the mutual information for cluster C_t between the two axes using $I(X;Y)$. The maximum mutual information is when all clusters intersect with the two axes at the exactly same bins. As expected, mutual information decreases for increasing k, but not monotonically for every k, as shown in Figure 35a).

5.6.6 Average Split Count

The average number of split cluster per axis pair is an indication of the traceability uncertainty. When an attacker selects a cluster of interest, the larger the number of cluster splits on the adjacent axes, the more difficult will it be to trace the cluster that contains the same record as the selected cluster. This form of uncertainty helps in meeting the *l*-diversity criteria [82], which ensures sufficient diversity between a quasi-identifier axis and a sensitive attribute axis, so that a cluster cannot be associated with exactly one sensitive value.

For computing the average split count, we have to consider two axis pairs together as shown in Figure 34. For each cluster, we compute the number of splits on adjacent axes. The average number of spits per cluster in axis pair, indicates the level of traceability uncertainty where T is given by the following equation:

$$T = \frac{1}{n_c} \sum_{i=1}^{n_c} \frac{1}{Split(C_i)} \tag{9}$$

where $Split(C_i)$ counts the number of split clusters for the i_{th} cluster on adjacent axis. The lowest traceability uncertainty is when $T = 1$, that is, there is a one-one association between clusters on adjacent axes. In case of multidimensional clustering T is always equal to 1. We consider T as a utility metric and $1 - T$ as a privacy metric. In this respect multidimensional clustering has higher utility than visual clustering. However, the privacy is lower because each cluster can be associated with exactly one cluster on the adjacent axis. In case of *l*-diversity this becomes a problem and can lead to disclosure of sensitive attributes.

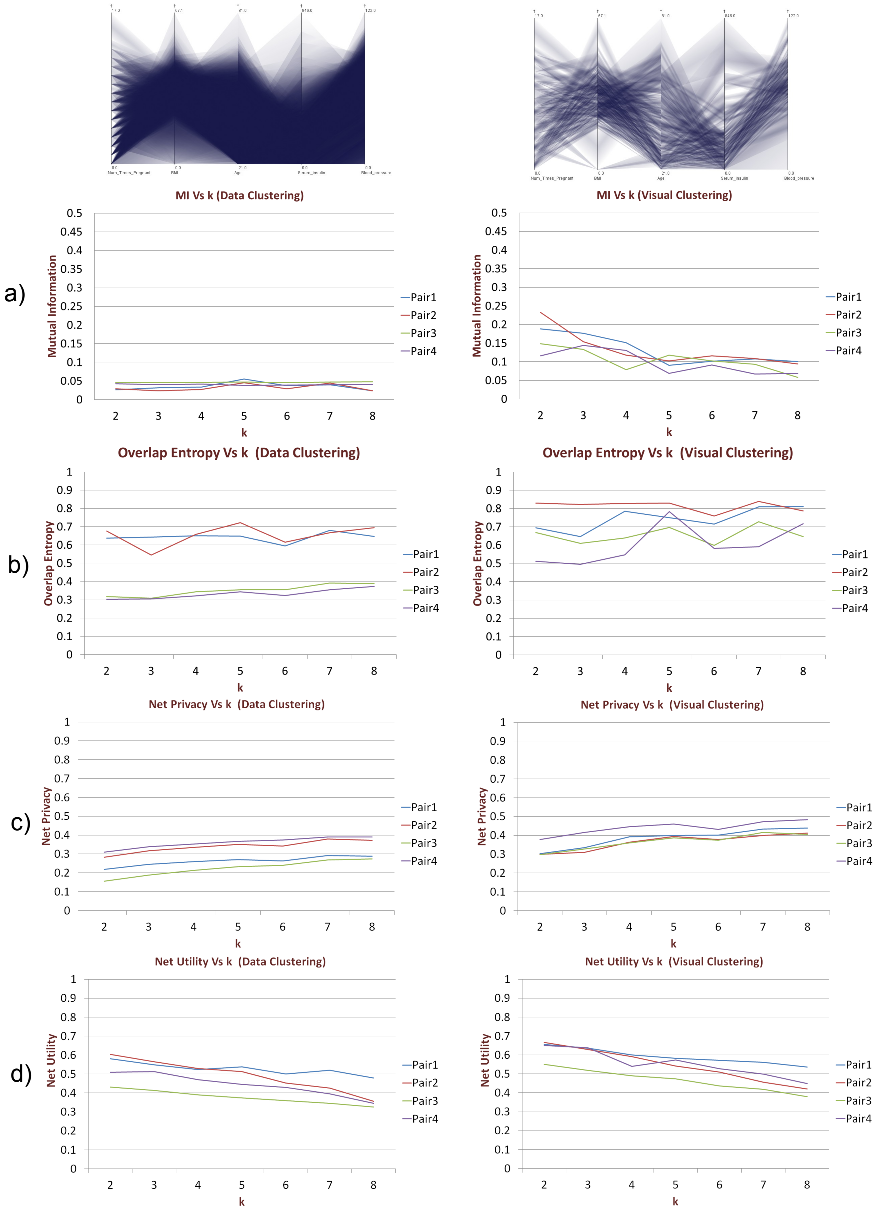


Figure 35: Comparison of privacy and utility metrics for four different axis pairs in case of data clustering and visual clustering.

5.6.7 Measures based on Pargnostics

Visual structures in a visualization represent semantic patterns within the data. Transformation of the representation distorts those structures as in privacy-preserving clustering. The different visual artifacts, like parallel lines, converging/diverging lines and line-crossings between adjacent axes; represent the trends and relationship between adjacent data dimensions. In case of cluster-based visualization, it is useful to see how the structures get preserved or distorted with comparison to line-based parallel coordinates.

Since most cluster boundaries represent connection between actual data points, we treat the cluster boundaries as lines and apply the Pargnostics metrics on this lines. Even in some cases where the cluster boundaries do not represent actual data points, their orientation between an axis-pair leads to the overall perception of the dominant visual structure there.

Cluster Parallelism. To describe parallelism, we compute a vertical distance histogram between any two cluster boundaries on adjacent axes. Then we look at the distribution of the distance values and estimate the interquartile range. Narrower range implies higher parallelism. We normalize the distances between 0 and 1, by dividing by the highest possible distance. With large cluster ranges $Par_{cluster}$ gets distorted.

Cluster Convergence/Divergence. In the original parallel coordinates, lines converging to or diverging from a few points on the adjacent axis form a frequently occurring pattern. We exploit these properties for seeding our clusters. The points with most convergence/divergence (which of them is the dominant structure) are our starting points for clustering. Similar to Pargnostics, we use the two-dimensional axis histogram to calculate the amount of convergence/divergence between adjacent axes and normalize the values with the maximum value of convergence/divergence. If V is the average pattern preservation for an axis pair, then we have:

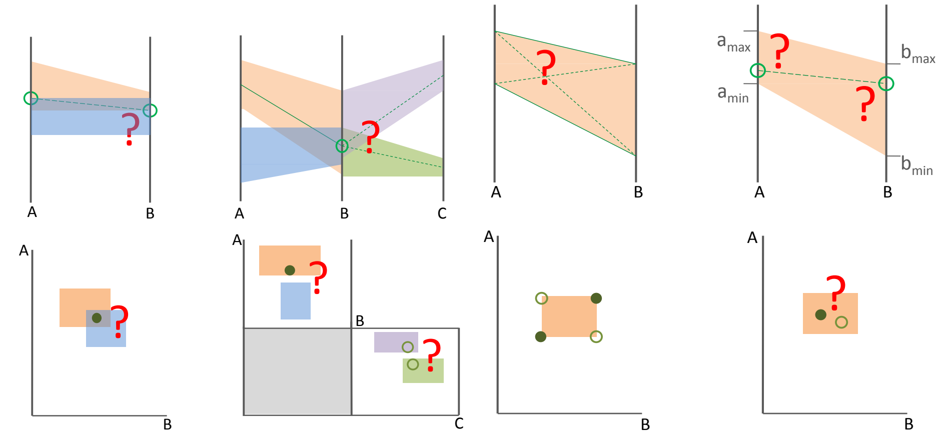
$$V = \frac{1}{2}\left(\frac{Par_{cluster}}{Par_{lines}} + \frac{CD_{cluster}}{CD_{lines}}\right) \tag{10}$$

5.6.8 Additivity of Uncertainty Metrics

The metrics proposed in the previous discussion are added based on whether a metric relates to privacy or utility. While some of the privacy metrics can also be applied for measuring utility or vice versa, we categorize the metrics based on their primary characteristic (either privacy or utility).

The metrics for encoding uncertainty, like cluster summary error and cluster range given by Equations 4, 5, and those for decoding uncertainty given by overlap entropy and average split count given by Equations 7, 9 give us a measure of privacy.

The metrics for decoding uncertainty, like overlap clutter, mutual information, Pargnostics metrics and average split count, given by Equations 6, 8, 9, 10 give us a measure of utility. We normalize the measures by giving equal weights to all the different uncertainty measures. Depending on the data owner or the visualization designer, these weights can be made non-uniform. Referring back to Equation 2, these different metrics represent the different amounts of uncertainty (u_i) in the visualization. The measures for utility have been computed as $(1-u_i)$ so that lower uncertainty corresponds to higher utility. We calculate privacy (m_p) and utility(m_t) as described below. The graphs for m_p and m_t are shown in Figures 35c and 35d.



(a) Cluster overlaps between axes (left) and across adjacent axis pairs (right) create uncertainty in guessing the cluster membership of known data points.

(b) Different cluster configurations (left) and cluster ranges (right) create uncertainty in knowing the coordinates and orientation of individual records inside a cluster.

Figure 36: Illustrating how visual uncertainty helps in preserving privacy for different re-identification scenarios. Inter-cluster uncertainty helps in privacy-preservation when an attacker attempts to determine the cluster membership of a known data point in case of the prosecutor re-identification scenario. Intra-cluster uncertainty helps in privacy-preservation when an attacker attempts to gain knowledge about the data at a lower level of granularity than what is shown, in case of journalist re-identification scenario.

$$m_p = \frac{\eta + \gamma + \Phi + (1 - T)}{4} \tag{11}$$

$$m_t = \frac{V + I + T + C}{4} \tag{12}$$

5.7 Handling Attack Scenarios Through Metrics

Privacy-preserving clustering exhibits inherent visual uncertainty that can be utilized for defending against potential attack scenarios. In Table 4 we summarize the causes of visual uncertainty that correspond to the privacy issues and connect them to the visual artifacts that are the sources of uncertainty and act as an additional layer of protection besides privacy-preserving clustering against possibilities of disclosure.

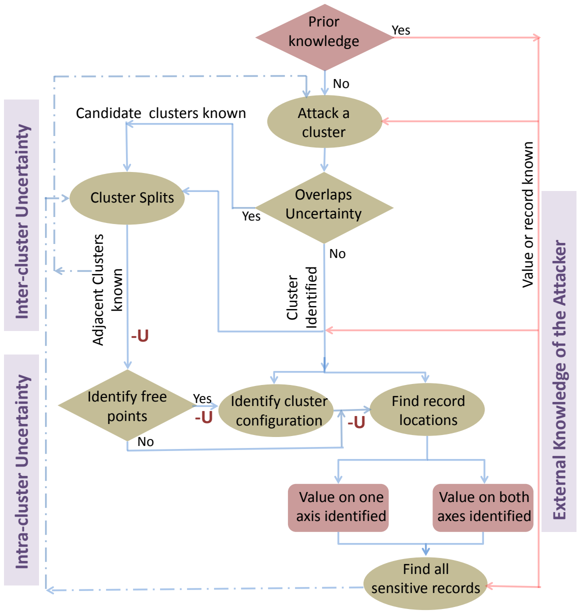


Figure 37: Summarizing the connection among the different attack scenarios, inter and intra-cluster uncertainty and privacy risks. Red arrows and boxes represent either use of prior knowledge or gaining knowledge about sensitive data from the visualization. $-U$ indicates reduction of uncertainty through interaction and use of combinatorics.

Inter-Cluster Uncertainty: Even if cluster edges comprise of real data points, many clusters overlapping at the same point can cause identity uncertainty. Moreover, even if the attacker knows that a certain data point or a record exists in the database, overlaps can make it difficult for him/her to identify which cluster that entity belongs to and also trace the path of the record across different axis-pairs in parallel coordinates (Section 5.6.3). Examples of inter-cluster uncertainty are shown in Figure 36(a).

Intra-Cluster Uncertainty: Granularity uncertainty is related to number of elements per cluster and the connections among them. We assume that an attacker has knowledge of k, that is the number of data points per cluster. The sources of granularity uncertainty are i) cluster ranges on an axis-pair that hide the precise location of the data points and ii) cluster configuration that hides the connection among the points. Because of the different possibilities of cluster configurations depending on different values of k, an attacker has to overcome this form of uncertainty to gain knowledge about the data. In many cases, using interaction an attacker can know about a cluster configuration (Section 5.7.2). Knowing a cluster configuration in many cases would not reveal the precise location of the end points. Accurately guessing the non-edge coordinates of records inside the cluster would mean that the attacker has to work out a number of combinatorial cases to overcome the lack of spatial accuracy due to the pixel resolution of the cluster range (Section 1). Examples of inter-cluster uncertainty are shown in Figure 36.

Uncertainty Reduction and Attack Scenarios: In Figure 37, the flowchart describes the different levels of knowledge that an attacker can have and how the different sources and types of uncertainty need to be overcome for possible privacy-breach scenarios. In the flowchart, we can observe that without any background knowledge, an attacker has to first overcome inter-cluster uncertainty and only then can narrow down to a single or candidate clusters for analyzing their internal structures. As we will discuss in the following sections, -*U* denotes the cases where reduction of one form of uncertainty leads to the reduction of the other.

5.7.1 Inter-Cluster Uncertainty: Overlaps Among Clusters Across Adjacent Axis Pairs

Overlaps among clusters across adjacent axis pairs is only applicable to parallel coordinates. In the screen-space, clustering is done independently for each axis pair. Therefore the clusters across axes appear to be discontinuous at the connecting points. When the records in a particular cluster continue on to two different clusters, then there appears to be a *split* (Figure 38(a)). While the cluster splits generally are helpful for preserving privacy as they hide the precise path a poly-line traverses across the different axes, leading to traceability uncertainty, some of these splits can divulge information about the cluster configurations. Similar to the derivation of the overlapping conditions in the preceding discussion, the split conditions can be derived from the overlap conditions between clusters on adjacent axes from Allen's interval algebra.

Relation between adjacent clusters: Overlap relationship between split clusters on adjacent axis-pairs, that can be derived from Allen's interval algebra, can reveal the location of the free points on the originating cluster. Refer to Figure 38(a), left image. Here since $C : A = E$ and $C : B = E$, we immediately know that there are no free points. In Figure 38(a), right image, $C : C_s = O$, but since the condition is *during*, there is no knowledge of split and hence the number of free points is not known. In Figure 38(b), for the left image, $A : B = N$. In that case not only the free points are revealed, but we also know area on the cluster which do not contain data points. In case of the right image, where $A : B = O$, we can know the free points if there is enough transparency. In that case, however we are not able to to know the area that does not contain any data points.

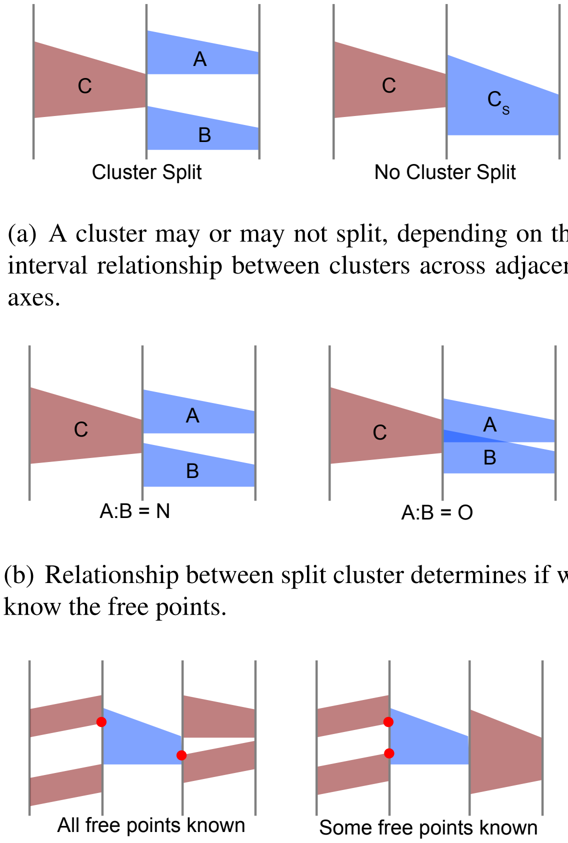
Number of Splits: The number of splits is upper bound by the number of records in each cluster, i.e., if a cluster contains k records, then there can be a maximum of k splits. If there are less than k splits, then there is a higher probability of uncertainty in tracing which records belong to which split clusters. On the other hand, if there are exactly k split clusters, then each split cluster contains a record each from the originating cluster and there is no uncertainty in guessing the distribution of records.

Knowledge about configuration: With brushing, clusters that contain the same record for all axes are visible and therefore the split configurations on both sides of a two-dimensional cluster can be known. Two such configurations are shown in Figure 38(c), left and Figure 38(c), right. For the left image, free points on both side are known. Since the number of free points on both side is one, it follows from our discussion in Section 5.7.2, that the cluster cannot have a pivot edge configuration and thus, has at least one real edge line. In the second case, any free points that might exist are not revealed due to the during relationship. So, the configuration of a cluster is not known exactly and one has to work out the combinatorial cases mentioned in Section 1 for guessing the exact location of the end-points of the records.

5.7.1.1 Uncertainty Due to Overlaps

In this section, we formally model the uncertainty caused by overlaps based on two types of uncertainty: identity uncertainty and traceability uncertainty; and also discuss any uncertainty reduced due to the overlaps.

Identity Uncertainty:Between adjacent axes, cluster overlaps can lead to identity uncertainty about the existence of a data point or about the cluster membership of a data point, based on the attacker's background knowledge. The probability computations in Section 1 would be affected if multiple clusters overlap or meet at the point under consideration. In that case, if we assume an attacker knows the existence of a record, the probability of inference would be reduced, proportional to the number of overlapped clusters. If the attacker does not know if the data point exists or not, the uncertainty would further depend on the overlap type. If the point under consideration lies at the meeting point of lines, then the



(a) A cluster may or may not split, depending on the interval relationship between clusters across adjacent axes.

(b) Relationship between split cluster determines if we know the free points.

(c) a) Free points known and not a pivot configuration, b) Free points not known and might be an edge or a pivot configuration.

Figure 38: Illustrating how the different types of splits reveal information about the originating cluster.

probability is not affected as the precise location is revealed. However, if the point lies in an overlapped region, then the probability of inference is further reduced, proportional to the area of overlap. We would that the number of edge-meeting conditions for clusters to decrease with increasing k. This is because with increasing k there is a greater loss in precision of the values on the axis as the cluster ranges get higher [38]. This is beneficial from a privacy point-of-view because for smaller k, cluster boundaries meeting precisely on a pixel can potentially cause disclosure of the the data-points.

Traceability: In this section we address the traceability uncertainty, that is, knowing the configurations within a cluster, how does one know the distribution of the records in the split/continuing clusters? This depends on the number of splits. If the number of splits is equal to the number of records within the cluster, then the split clusters share one record each with the originating cluster. This is a less uncertain case than when the number of splits is less than the number of records in each cluster. Then the distribution of records in the continuing clusters is not immediately known.

Let t be the number of splits where $t < k$. The number of records in each split cluster can range from 1 to $k - t + 1$. The problem of finding the number of possible distribution of records in the split clusters then reduces to finding the number of t-subsets of k elements. Let us assume the worst-case scenario of the attacker knowing the records within a cluster and then trying to find out their distribution in the split clusters. This is analogous to the problem of distributing k distinguishable objects (the records) in t non-empty indistinguishable boxes. This is given by Stirling number of the second kind [101], $S(k,t)$ which is given by the following standard formula:

$$S(k,t) = \frac{1}{t!}\sum_{i=1}^{t} -1^{i} C(t,i)(t-i)^{k} \qquad (13)$$

$C(t,i)$ denotes the combination of t things, taken i at a time. A two-dimensional table of values for the above formula relating k to t is readily available in the discrete mathematics literature [79]. The number of possibilities when the number of splits approaches k increases drastically and thus, if the attacker does not have any further background knowledge about the data, it would be very difficult to know the distribution of records in the split clusters.

5.7.1.2 Uncertainty Reduction

Uncertainty about a configuration can be reduced by knowing the free points as we had discussed in Section 5.7.2.5. In case of an overlap (O) between the originating cluster and split cluster, free points are revealed, while in case of an edge-meeting (E), they are not. Let f_j denote the number of free points on the j^{th} axis. The ability of an attacker to know the exact number of free points depends on the overlap relation between the originating cluster and the split clusters. If t_j is the number of split clusters on the j^{th}, C^j is the originating cluster, and C^i_j is a corresponding split cluster, then the number of free points known is given by:

$$f_j = \sum_{i=1}^{t} \begin{cases} 1 & \text{if } C_j : C^i_j = O \\ 0 & \text{if } C_j : C^i_j = E \end{cases}$$

For a given dimension, we calculate the net risk of known free points on the j^{th} axis by the following formula, where f^i_j denotes the number of split clusters on the j^{th} axis for the

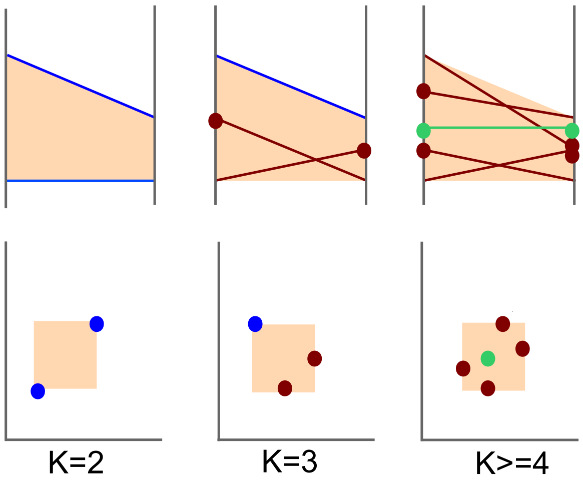


Figure 39: From left to right, a cluster can be defined by: all edge elements, a combination of edge and pivot elements, and also no edge element when $k >= 4$. Edge elements are shown in blue, pivot elements in red, and free elements in green.

i^{th} originating clusters, n being the number of originating clusters:

$$R(f_j) = \frac{1}{n}\sum_{i=1}^{n}\frac{f_j^i}{t_j^i} \quad 0 < R(f_j) < 1 \tag{14}$$

In case of all free points known for every cluster, $R(f_j) = 1$. The lesser the value of $R(f_j)$, the lower is the risk for the j^{th} dimension.

5.7.2 Intra-Cluster Uncertainty: Cluster Configurations

A two-dimensional cluster configuration in scatter plots and parallel coordinates is defined by the pixel coordinates of the data points within the cluster. A pixel coordinate in scatter plots is represented by a point, while in parallel coordinates, it is a line, by the point-line duality principle [65]. For the sake of clarity, we refer to the points/lines as elements within the cluster. The shape of a cluster in parallel coordinates can be a quadrilateral or a triangle and the same in scatter plots is either a rectangle or a line. In case of triangular clusters, the uncertainty is very low, since coordinates of the two borders or end-points is always known. Thus $k = 2$ has no anonymization effect in case of triangular clusters. In case of numerical dimensions, the probability of occurrence of quadrilateral clusters is much higher than triangular or linear ones. In this section, we study the orientation of the elements within clusters for different values of k, how they cause different types of uncertainty and their relation to privacy for different values of k.

5.7.2.1 Different elements in a cluster

Location of the elements within the cluster determine the uncertainty associated with them. Since cluster borders are formed by data-points, elements located at the edge or corner of a cluster have high vulnerability. On the other hand, the cluster coordinates that are not on the corner have a higher level of privacy. We term these as *free points*. We define the different elements within the cluster as follows:

Corner elements: In parallel coordinates these are the lines connecting two pairs of corner points: they can be either the borders or the diagonals. In scatter plots these are the corner points. These are marked in blue in Figure 39. These are most vulnerable to disclosure as one of the coordinates of the corner points is always known to the attacker.

Pivot elements: In parallel coordinates these are the lines that connect corner points to free points. In scatter plots these are free points located on the edges. These are marked in red in Figure 39. These are less vulnerable than the edge elements, as one only of the coordinates is revealed by the visualization.

Free elements: In parallel coordinates, these are the lines that connect a a pair of free points. In scatter plots, these are the non-corner coordinates. These are marked in green in

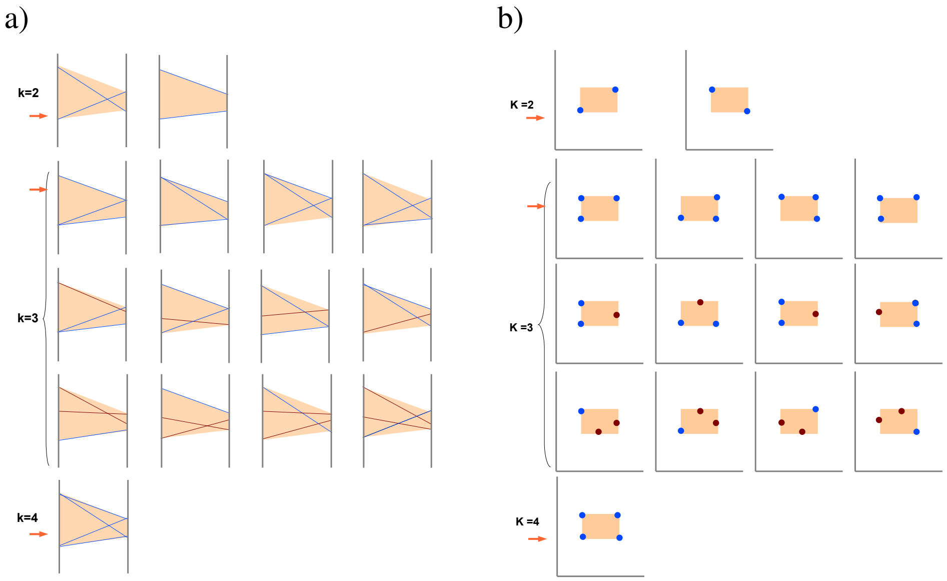


Figure 40: Base configurations in a) parallel coordinates and b) scatter plots on which the configurations for higher k are based upon. The arrowheads denote edge-only configurations and the others are mixed edge configurations.

Figure 39. These have the highest degree of privacy as they can be located anywhere within the cluster and both coordinates are difficult to guess. In triangular or linear clusters, free elements can be located only on one of the axes, therefore the uncertainty is much lower as one of the coordinates is always known.

Depending on the value of k, a configuration can be defined by only edge elements, a combination of edge and pivot elements, or only pivot elements. These are also shown in Figure 39. In the following sections we define and quantify how different configurations can be formed by the cluster elements.

5.7.2.2 Base Configurations

To define a cluster configuration minimally, the edges or the corners have to be defined first. We term those cluster configurations as base configurations, which are concerned with the orientation of edge elements, and from which others can be derived. Base config-

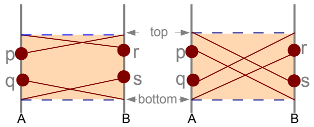


Figure 41: Illustrating how free points define false edges. The dotted lines represent false edges. For each pair of free points, pq or rs, there are two options: either they cross or do not cross. There are thus two ways for each free point to be associated with a corner point: top or bottom. This denotes an ordered selection of points for defining a false edge.

urations (n_{b_k}) for $k = 2$, $k = 3$ and $k = 4$ are shown in Figure 40. When the number of edge elements is equal to k, we call the configuration a edge-only configuration. Otherwise, it is either a mixed edge configuration, made of edge elements and pivot elements or for $k >= 4$ there can be pivot edge configurations made of edge and pivot elements. For triangular clusters, there can be only one edge-only configuration. The following analysis therefore, applies to quadrilateral clusters.

Edge-only configuration: Configurations that are formed by only edge elements. Let the number of possible edge-only configurations for a certain k be n_{el_k}. For $k = 2$, $n_{el_k} = 2$. For $k = 3$, there can be four additional configurations as shown in Figure 40 and For $k = 4$ there is one added configuration as all the edge elements can form edges. For $k > 4$, there cannot be any new edge added, so $n_{el_k} = 7$.

Mixed edge configuration: In a mixed edge configuration, two pivot elements can define an edge/corner while the other edge/corner consists of an edge element. Since a minimum of three elements are needed, for $k = 2$ there is no mixed edge configuration. For $k = 3$, there can be four different possibilities for a one-edge configuration and four more, for a two-edge configuration, giving a total of 8 mixed-edge configurations (n_{em_k}). These are shown in Figure 40.

Pivot edge configurations: Those configurations where no edge/corner is formed by an edge element but only by the pivot elements are referred to as pivot configurations (n_{ep_k}). These configurations have a higher level of privacy as in absence of real edges, there can be many different possibilities for connection among corner and free points. Pivot configurations are only possible for $k >= 4$ and their structure depend on the number and distribution of free points.

Let the number of free points be f. Number of data points within a cluster = k. Number of corner points on each side= 2. Therefore, maximum number of free points possible for any k is given by $f_{max} = 2(k-2) = 2k-4$. Figure 41 illustrates how pivot configurations are built from free points in parallel coordinates. Let p and q be the the free points. If they have to define the corner points, then they can connect with either the top corner point or the bottom one on the other axis: lines from p and q either intersect or do not. The same argument applies to r and s and this is how the false edges denoted by the dotted lines are formed. This denotes an ordered selection of free points on each axis, the order (pq in the left image and qp in the right image) being the direction. A minimum of 4 free points are needed to define a pivot configuration and from the formula of f_{max}, we can deduce that pivot configurations are not possible for $k < 4$. The number of possible pivot configurations when $k = 4$ is thus given by the number of possible selection of two points for each axis, that is $2! * 2! = 4$. For higher k, this is similar to a combinatorial problem when we have to select n different things, taken r at a time and the order matters. Here n refers to the free points, that is f and r refers to the available locations, i,e., 2. To define a pivot configuration, the minimum number of free points on one axis is 2 and so the maximum number of free points on the other axis is $f_{max} - 2$. The total number of pivot configurations is therefore given by:

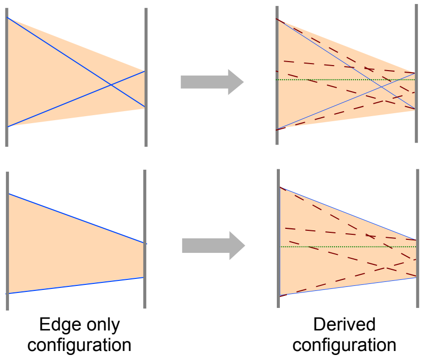


Figure 42: Illustrating derived configurations based on edge-only configurations for $k = 2$. The dotted red lines represent the possible pivot edges that can be added and the dotted green line denotes a free edge that can be added, giving a total of five degrees of freedom for a derived configuration.

$$n_{ep_k} = \sum_{i=2}^{f_{max}-2} P(i,2) * P(f_{max} - 2 - i, 2) \quad k > 4 \tag{15}$$

The total number of base configurations is therefore given by the sum of the edge-only, mixed and pivot edge configurations.

$$n_{b_k} = n_{el_k} + n_{em_k} + n_{ep_k}$$

5.7.2.3 Derived Configurations

Derived configurations are those that can be constructed from the base configurations for $k = 2,3,4$ as the added elements become pivot elements or free elements. The pivot elements have four degrees of freedom: they are attached to any one of the four corners. For the free element there is an added degree of freedom, giving a total of five degrees of freedom for a derived configuration. For example, if $k = 5$, a configuration can be built with an edge-only configuration with two edge elements (these are the two possible edge-only configurations for $k = 2$). Each of the additional three elements have 5 degrees of freedom each, and therefore total number of possible configurations with two edge elements is $2*3*5 = 30$. The formation of a derived configuration from a edge-only configuration for $k = 2$ is shown in Figure 42.

Now let us generalize the formula for any k. If the number of edge elements is i, the number of non-edge elements is $k-i$. From the discussion above, the number of derived configurations (n_{d_k}) is given by:

$$n_{d_k} = \sum_{i=2}^{k_{max}} n_{b_i} * (k-i) * 5 \quad n_{b_i} = n_{el_i} + n_{em_i} + n_{ep_i} \tag{16}$$

where $k_{max} = k$ if $k < 4$ and $k_{max} = 4$ if $k => 4$. The factor of $(k-i)*5)$ is a multiplicative factor that gives the configurations for added pivot elements and free elements, without adding edge-elements.

The total number of possible cluster configurations is given by:

$$n_{c_k} = n_{b_k} + n_{d_k}$$

5.7.2.4 Uncertainty due to Cluster Configuration

In this section we quantify granularity uncertainty caused by cluster configurations and also look at the potential reduction in uncertainty about a configuration from visual artifacts.

Granularity Uncertainty

The most fundamental metric for analyzing the disclosure risk within a cluster is the number of data entities in the cluster. $k > 1$. One could assume that all records have equal

k	**Edge-only** (n_{el_k})	**Mixed edge** (n_{em_k})	**Pivot edge** (n_{ep_k})
2	2	0	0
3	4	18	0
4	7	61	4
5	7	115	21
6	7	170	126
7	7	225	462
8	7	280	1287

Table 7: Number of possibilities for the base configurations for different values of k.

probability to be identified, the risk for a specific record L_i to be identified is thus:

$$P(L_i) = \frac{1}{k}, \quad i = 1, 2, \ldots, k$$

In practice, this almost equates to a scenario where k records are put into bag, and an attacker can pick one record out of the bag randomly. This may be an over-simplification in analyzing the risks of cluster-based parallel coordinates and scatter plots, because, to know the connection among the elements within a cluster, or in other words, all the two-dimensional coordinates, an attacker has to guess the correct configuration, i.e., the orientation of the records within the cluster. The probability of a correct guess is given by the following equation:

$$P(c_k) = \frac{1}{n_{c_k}} \tag{17}$$

5.7.2.5 Uncertainty Reduction

Uncertainty about a cluster configuration can be reduced from the knowledge of free points. If the attacker knows that the number of free points on one of the axes (the j^{th} axis in the following equation) is zero, then there has to be real edge elements that define the edges of the cluster. This would imply that the configuration is edge-only as no pivot configurations or mixed edge configuration possible. If c_k^j is a cluster on the j^{th} axis, then it follows:

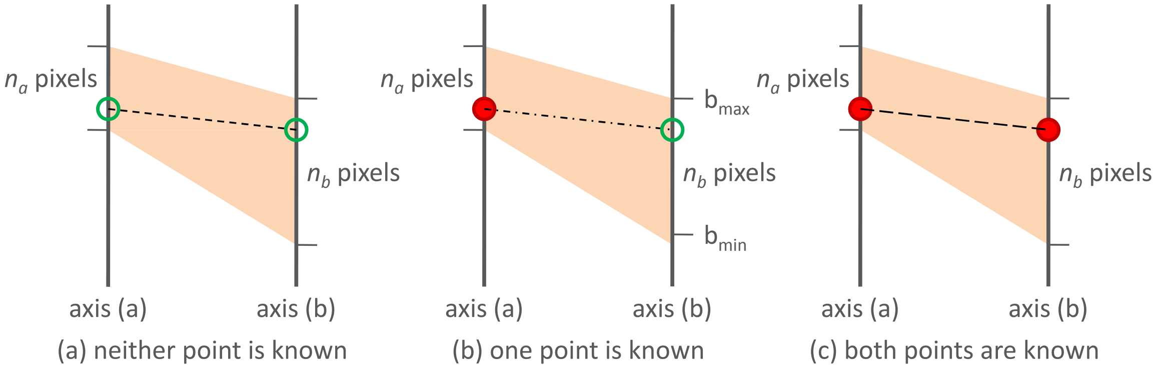
$$P(c_k^j|f^j = 0) = \frac{1}{n_{el_k}} \quad since \;\; n_{ep_k},\, n_{em_k} = 0 \tag{18}$$

This implies that the attacker has to make only a limited number of guesses to know the connections among the coordinates as shown in Table 7. We can see that for increasing k, the possible number of edge-only configurations remain constant at *seven* and thus poses a higher risk than the other two types of configuration.

If an attacker knows that the number of free points is non-zero but less than two on any axis, then it cannot be a pivot configuration, because a minimum of two free points is needed to define both edges of a cluster, as discussed earlier. Thus the number of possible configurations is given by the following equation:

$$P(c_k^j|f^j < 2) = \frac{1}{n_{el_k} + n_{em_k}} \quad since \;\; n_{ep_k} = 0 \tag{19}$$

As shown in Table 7, the increase in number of pivot configurations is much steeper than that of mixed-edge configurations as k goes higher than 6. In case of pivot configurations, although an attacker can tentatively guess a configuration from the distribution of free points, the connections among those points and the precise location of those would still be unknown in most cases. Even if the location of the free points are revealed by the visualization itself (that we will study in Section 5.6.3), then the configurations do not reveal information about the connection among the free points. So, the uncertainty reduced



(a) neither point is known (b) one point is known (c) both points are known

Figure 43: Three basic scenarios about attacker's knowledge about the end points of a line. This relates to spatial accuracy in guessing the exact locations of the points on the axes.

due to analysis of cluster configurations, would be mostly restricted to knowledge about the corner elements.

1 Intra-Cluster Uncertainty: Cluster Range

Cluster configurations are not affected by pixel resolution but for a detailed analysis of the attack scenarios, we have to take the pixel range of clusters into account. When an attacker has some background knowledge about some attribute values (end points in parallel coordinates) and/or is more interested in discovering attribute values, the plan of attack would be based on working through a number of combinatorial cases for overcoming the uncertainty caused by lack of spatial accuracy owing to the pixel ranges of clusters. In this section we study the disclosure risk attached with knowledge of end points. The methods for determining the amount of uncertainty or vulnerability about end points can also be extended to scatter plots, except that the notion of end points is transformed to coordinates of sample points in a scatter plots. There are also four corner points in scatter plots as illustrated in Section 5.7.2. Since for a triangular shaped cluster one end is always known, we focus our analysis on two-dimensional quadrilateral-shaped clusters between adjacent axes. Figure 43 shows three example scenarios where an attacker would try to breach privacy:

1. In relation to two attributes (a) and (b), the attacker may have the knowledge that a specific line must be in the cluster, but not much more. Hence the attacker has to make wild guesses about possible values on axes (a) and (b). These guesses translate visually to the guesses of both green end points on the two axes.

2. The attacker may have already discovered attribute value on axis (a) about this target line, and want to find out attribute value on axis (b). In other words, he needs to guess where the green end point is.

3. The attacker may have already found out attribute values on both axes (a) and (b) about this target line, and realize that there may be more than one line with these two attributes. In other words, what is the certainty or uncertainty about the hypothesis that this is the target line. Such a confirmation will be useful to the attacker when he progressively moves onto the next pair of axes.

In the following subsections, we first examine scenario (b), and provide a method for quantifying the uncertainty in relation to the number of lines in the cluster, k, and the resolution of the visualization. Building on the analysis for (b), we examine scenarios (a) and (c). We assume that the attacker has some kind of background knowledge using which the existence of a line, given two correct attribute values (i.e., both end points) can be confirmed.

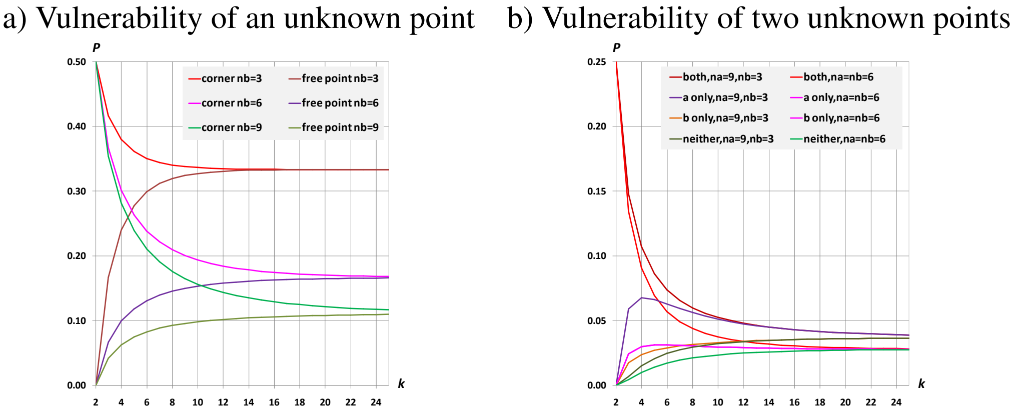

a) Vulnerability of an unknown point b) Vulnerability of two unknown points

Figure 44: Quantifying risks based on uncertainty about end points. (a) Given the attacker knows one end point of a line, the plot shows the certainty (or vulnerability) of the other end-point as a function of k and n_b where n_b is the cluster range on axis b. Whether or not the end-point is a corner point poses different risk factor. (b) Given the attacker only knows the containment of a line in a cluster but not the end points, the plot shows the certainty (or vulnerability) of both points with two different combinations of n_a and n_b.

1.1 Uncertainty about One End-point of a Specific Line

Considering Figure 43(b), let k be the number of lines in this cluster. Without losing generality, we assume that the attacker has discovered the value of attribute (a) associated with a specific line L_i. If the attacker does not gain any knowledge from the visualization about the possible location of the other end of the cluster, then the probability of a correct guess depends on a number of combinatorial cases. We can compute the number of valid combinations of k lines that pass through some of the n_b pixels, that is the cluster range on axis (b) as:

$$G(k,n_b) = \begin{cases} 1 & n_b = 1 \\ n_b^k - 2(n_b-1)^k + (n_b-2)^k & n_b > 1 \end{cases} \tag{20}$$

where the first term in the case of $n_b > 1$ is the number of all possible combinations, the second term defines the number of invalid cases where no line passes through either of the corner point, b_{min} and b_{max}, and the third term defines the number of cases where no line passes through either b_{min} nor b_{max}.

Consider two numbers that define the number of combinations of k lines with at least one line passing b_{min} and b_{max} respectively. The two numbers can be defined by using the same recurrence function:

$$
\begin{aligned}
&B(0,n_b) = 0 \\
&B(1,n_b) = 1 \\
&B(2,n_b) = 2n_b - 1 \\
&\quad\cdots \\
&B(k,n_b) = n_b^{k-1} + (n_b - 1)B(k-1,n_b)
\end{aligned}
\tag{21}
$$

Therefore, for the target line L_i to pass through either b_{min} or b_{max}, the number of combinations will be $B(k-1,n_b)$, since there must be at least one other line that passes through the other corner point.

Meanwhile, for L_i to pass through a free point, $b_{min} < b_x < b_{max}$, the number of valid combinations is simply $G(k-1,n_b)$.

Let $L_i \multimap \rho$ denote that the line L_i passes a point on axis (b), where ρ can be one of the values, $b_{min}, b_{min}+1, \ldots, b_{max}-1, b_{max}$. The vulnerability or certainty of this specific line that is to be guessed depends on its position on axis (b). For $k > 1, n_b > 2$, we have:

$$
P(L_i \multimap \rho) = \begin{cases} \dfrac{B(k-1,n_b)}{G(k,n_b)} & \text{if } \rho = b_{min} \text{ or } \rho = b_{max} \\[2ex] \dfrac{G(k-1,n_b)}{G(k,n_b)} & \text{if } b_{min} < \rho < b_{max} \end{cases}
$$

As shown in Figure 44(a), when k is relatively small, there is a significant difference between lines that pass through corner points (b_{min} or b_{max}, and lines that do not. Such a gap closes when k increases (i.e., there are more lines) as there can be a large number of pivot configurations that are possible due to the addition of pivot lines. In general, the higher value n_b is, the lower the certainty is, except that when $k = 2$, the resolution does not affect the uncertainty.

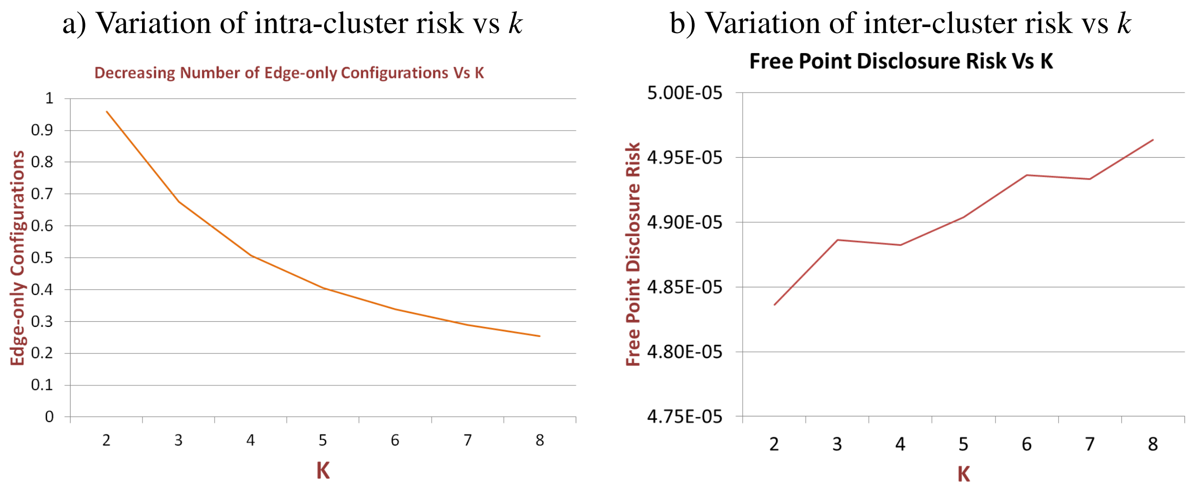
1.2 Uncertainty about Both End-points of a Specific Line

Building the above analysis, we now consider scenario (a) in Figure 43(b). Assume that the attacker knows the fact that a specific line is in a cluster, but do not know the value of either attribute. As the probabilistic distributions on the two axes are independent, we can derive the joint distribution from the distributions on individual axes.

Let $L_i \multimap\!\circ\, (\rho_a,\rho_b)$ denote the line L_i passes two points on axes (a) and (b) respectively, where ρ_a can be one of the integer values between a_{min} and a_{max}. For $k > 1, n_a > 2, n_b > 2$, the vulnerability or certainty of this specific line is to be guessed is:

$$P(L_i \multimap\!\circ\, (\rho_a,\rho_b)) = \begin{cases} \dfrac{B(k-1,n_a)B(k-1,n_b)}{G(k,n_a)G(k,n_b)} & \rho_a \text{ and } \rho_b \text{ are corners} \\[2ex] \dfrac{B(k-1,n_a)G(k-1,n_b)}{G(k,n_a)G(k,n_b)} & \text{only } \rho_a \text{ is a corner} \\[2ex] \dfrac{G(k-1,n_a)B(k-1,n_b)}{G(k,n_a)G(k,n_b)} & \text{only } \rho_b \text{ is a corner} \\[2ex] \dfrac{G(k-1,n_a)G(k-1,n_b)}{G(k,n_a)G(k,n_b)} & \text{neither is a corner} \end{cases}$$

Figure 44(b) shows the vulnerability in the case of both end points are unknown. It shows two situations when $n_a = 9, n_b = 3$ and when $n_a = n_b = 6$ respectively. It is useful to note that the first situation is slightly more vulnerable than the second, though the sums of the pixel



a) Variation of intra-cluster risk vs k

b) Variation of inter-cluster risk vs k

Figure 45: Number of edge-only configurations decreases with increasing k, while the disclosure risk for free points can increase with increasing k due to the uncertainty involving cluster splits.

resolutions on both axes are the same. This indicates that the lower pixel resolution on the right axis incurs more risks. When the sums of the pixel resolutions on the left and right axes are the same, a cluster patch of a trapezoid shape is more risky than a parallelogram. Nevertheless, in comparison with Fig. 44(a), the risk is noticeably lower.

1.3 Uncertainty about a Line with Disclosed End-points

After an attacker has discovered both end-points of a specific line, the next step will naturally be moving to the next pair of axes to repeat the above actions. However, discovering the two end points is not the same as an absolute certainty about the specific line. There can be two or more lines that share the same pair of end points. Knowing the probability of how many lines that may share the discovered end points would introduce uncertainty that the attacker has to deal with in the next attacking step. In other words, when the privacy of both end points on a pair of axes is breached, there may still be some uncertainty that can be used as a defence.

Let us assume the attacker has discovered two end-points, ρ_a and ρ_b, of a target line L_i.

Suppose that among the $k-1$ lines, there is at least one line that shares the same end points ρ_a and ρ_b.

Let us consider first a situation where both ρ_a and ρ_b are corner points, i.e., $(\rho_a = a_{min} \vee \rho_a = a_{max}) \wedge (\rho_b = b_{min} \vee \rho_b = b_{max})$ After excluding the 2 lines, there are $k-2$ lines left. The total number of valid cases is:

$$V_{cc}(k-1, n_a, n_b) = B(k-2, n_a)B(k-2, n_b),$$

where $B()$ is the recurrence function defined in Eq. (21). The probability for having such a line is:

$$P_{cc}(k-1) = \frac{B(k-2, n_a)B(k-2, n_b)}{G(k-1, n_a)G(k-1, n_b)}.$$

Among the remaining $k-2$ lines, it is probable that there is another line sharing the same end points ρ_a and ρ_b. The probability is

$$P_{cc}(k-2) = \frac{B(k-3, n_a)B(k-3, n_b)}{G(k-2, n_a)G(k-2, n_b)}.$$

We can obtain a series of probabilities, $P_{cc}(k-1), P_{cc}(k-1), \ldots, P_{cc}(2)$, by continuing this reasoning. We know $P_{cc}(1) = 0$, as long as $n_a > 1$ or $n_b > 1$.

Consider a *useful uncertainty* that is the probability of cases where at there is at least another line share the same end points with L_i. The probability of these is:

$$1 - \prod_{j=2}^{k-1}(1 - P_{cc}(j)).$$

Similarly, we can derive the probabilities in the other three situations. We have P_{cx} for the situation where ρ_a is a corner point, but ρ_b is not; P_{xc} for the situation where ρ_a is not a corner point, but ρ_b is; and P_{xx} for the situation where neither ρ_a nor ρ_b is a corner point.

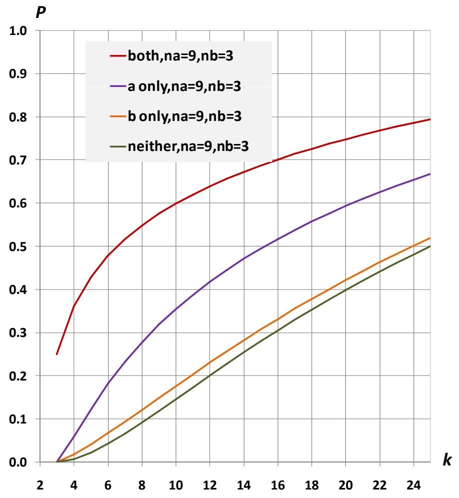


Figure 46: After both end points of a line is disclosed, there is still some uncertainty about whether or not there are other lines sharing the same end points. Such uncertainty grows rather quickly in relation to increasing *k*.

They can be computed as:

$$P_{cx}(k) = \frac{B(k-1,n_a)G(k-1,n_b)}{G(k,n_a)G(k,n_b)}$$

$$P_{xc}(k) = \frac{G(k-1,n_a)B(k-1,n_b)}{G(k,n_a)G(k,n_b)}$$

$$P_{xx}(k) = \frac{G(k-1,n_a)G(k-1,n_b)}{G(k,n_a)G(k,n_b)}$$

Hence, the uncertainty surrounding two discovered end-points in relation to a target line is:

$$P(L_i \fallingdotseq (\rho_a,\rho_b)) = 1 - \prod_{j=2}^{k-1} \begin{cases} (1-P_{cc}(j)) & \text{if } \rho_a \text{ and } \rho_b \text{ are corners} \\ (1-P_cx(j)) & \text{if only } \rho_a \text{ is a corner} \\ (1-P_{xc}(j)) & \text{if only } \rho_b \text{ is a corner} \\ (1-P_{xx}(j)) & \text{if neither is a corner} \end{cases}$$

Figure 46 shows such uncertainty associated with a situation when $n_a = 9, n_b = 3$. One can notice a continuing increase of this useful uncertainty when *k* increases. Such uncertainty is particular useful for slowing down the attacker's reasoning when there are overlapping clusters at axes.

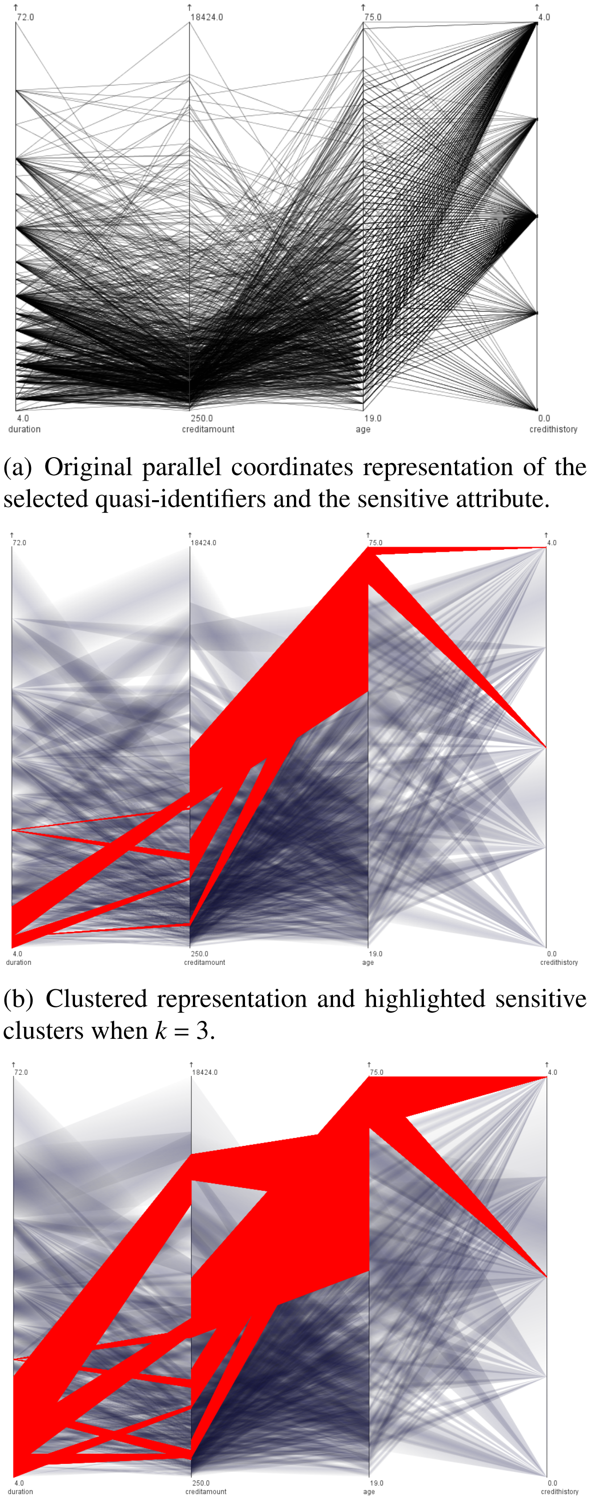
5.8 Case Studies

The German Credit dataset [52] has 1000 instances which classify bank account holders into credit classes *Good* or *Bad*. Each data object is described by 20 attributes that include 13 categorical and 7 numerical attributes. In our experiments we consider the *credithistory* as the sensitive attribute, however we assume that good credit history is not a sensitive value, but bad credit is, so we want to protect the value 4 for *credithistory*. For the other attributes we use a subset of the original attributes. We choose a mix of numerical and categorical attributes among the ones which show maximum information gain and are deemed selectable [90]. Those are: existing checking account *status*, *duration of loan*, *credithistory*, *savings accounts status*, *credit amount*, *personal* status (depends upon gender and marital status).

We also use the *Diabetes* dataset [52] to illustrate the application of our metrics. This dataset has 768 records and consists of 6 dimensions: *number of times pregnant*, *blood pressure*, *serum insulin level*, *body mass index (BMI)*, *age*, and the binary attribute *class*. The sensitive dimension is the *class* attribute, all others are considered to make up the quasi-identifier attribute.

5.8.1 *k*-anonymity

Figure 48(a) shows the default parallel coordinates layout for this dataset. And in Figure 48(b) and Figure 48(c) we see the *k*-anonymized version for $k = 3$ and $k = 8$. Although

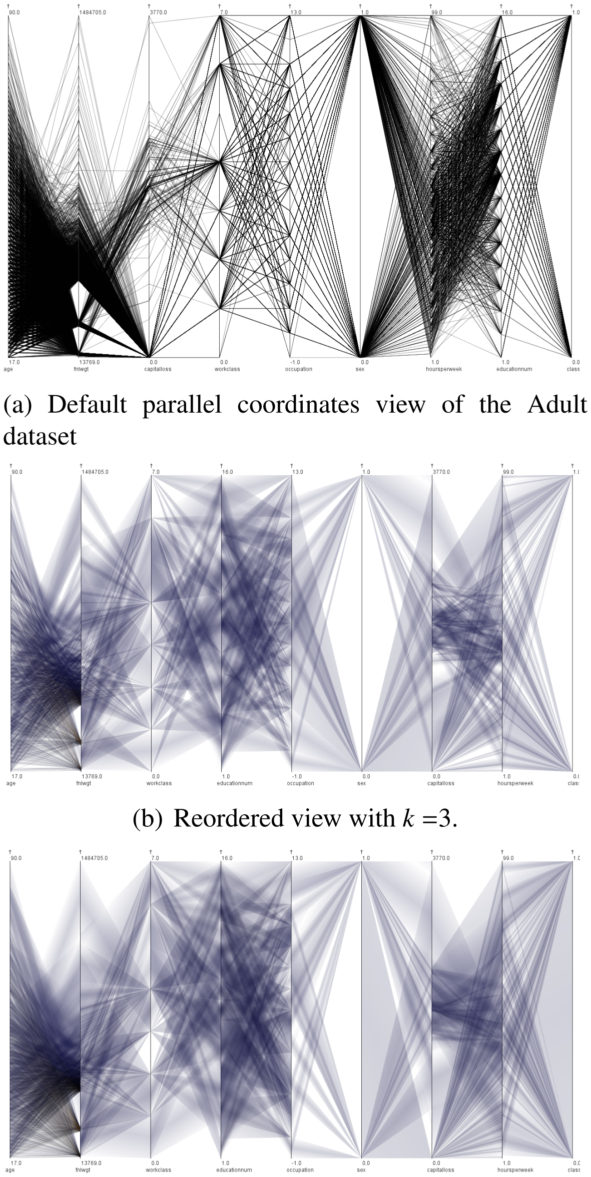


(a) Original parallel coordinates representation of the selected quasi-identifiers and the sensitive attribute.

(b) Clustered representation and highlighted sensitive clusters when $k = 3$.

(c) Clustered representation and highlighted sensitive clusters when $k = 7$.

Figure 47: With higher k the number of cluster splits become higher, as also the number of overlapping clusters on adjacent axis pairs. This can lead to disclosure of more free points and also more no-record areas within the clusters.



(a) Default parallel coordinates view of the Adult dataset

(b) Reordered view with k =3.

(c) View with k = 8.

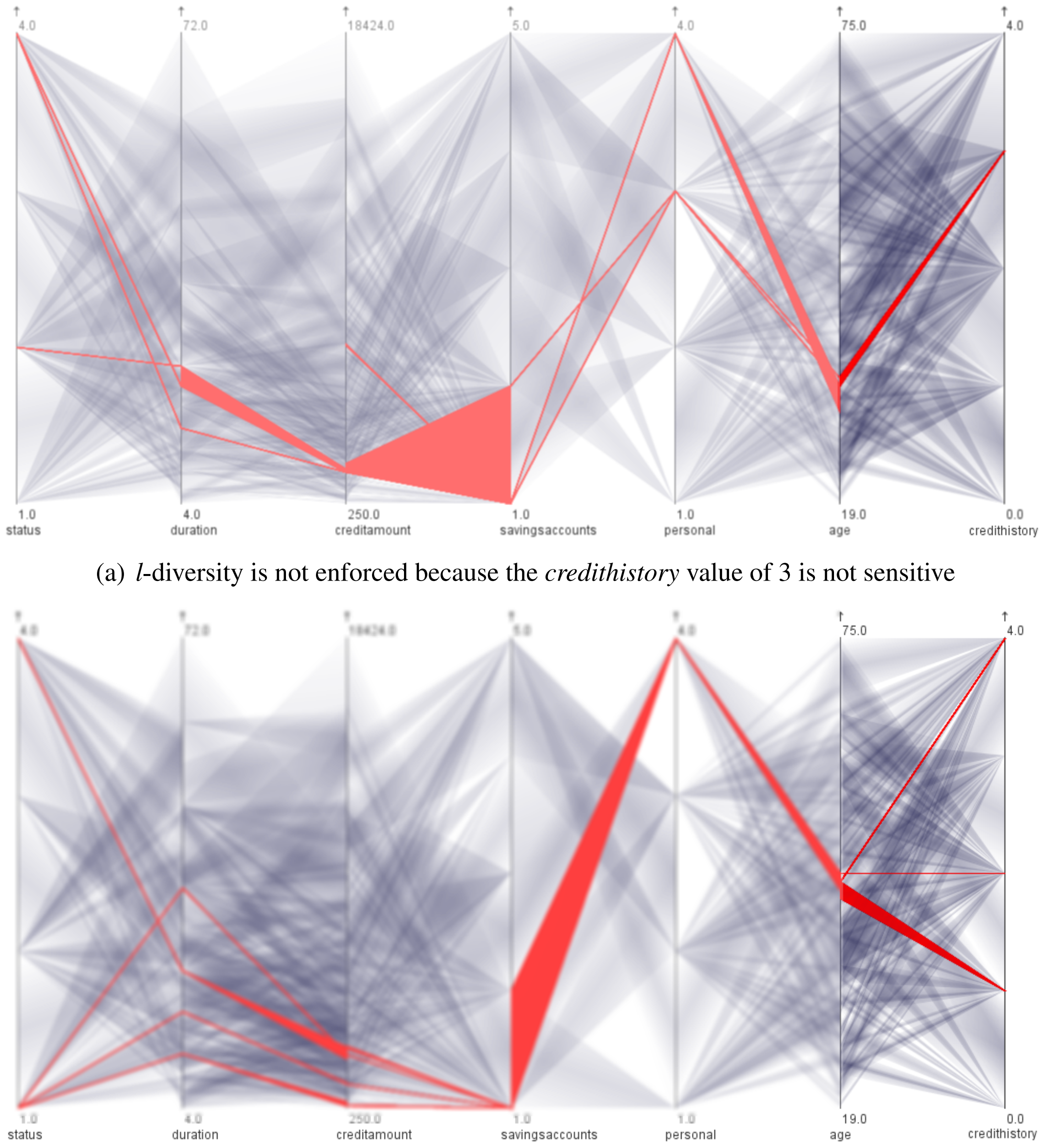
Figure 48: Illustrating k-anonymity after reordering the axes

the individual values cannot be seen, the overall structure of the parallel coordinates is largely retained. For lesser values of *k*, we get better discernability but lower privacy because the cluster ranges are small. For higher *k*, we get larger cluster ranges which make loss of precision very high and therefore guarantees more privacy.

5.8.2 *l*-diversity

Cluster splits and overlaps are an artifact of axis-pairwise clustering. As mentioned in Section 5.3.5, the splits lead to uncertainty. Splits are especially pronounced in the case of numerical axes and higher values of *k*. Even if we know the exact values of the quasi-identifiers, there are splits among the different axes which introduce uncertainty for guessing the exact value for a particular record (Figure 49). For the adjacency condition of a quasi-identifier and a sensitive attribute, we selectively enforce *l*-diversity. In Figure 49(a) we retain the highlighting of clusters based on just *k*-anonymity because the data owner has determined that only persons with a poor credit (in this case with a *credithistory* value of 4) should be protected. In Figure 49(b) we show that *l*-diversity is applied and we cannot tell three different values of *credithistory* apart.

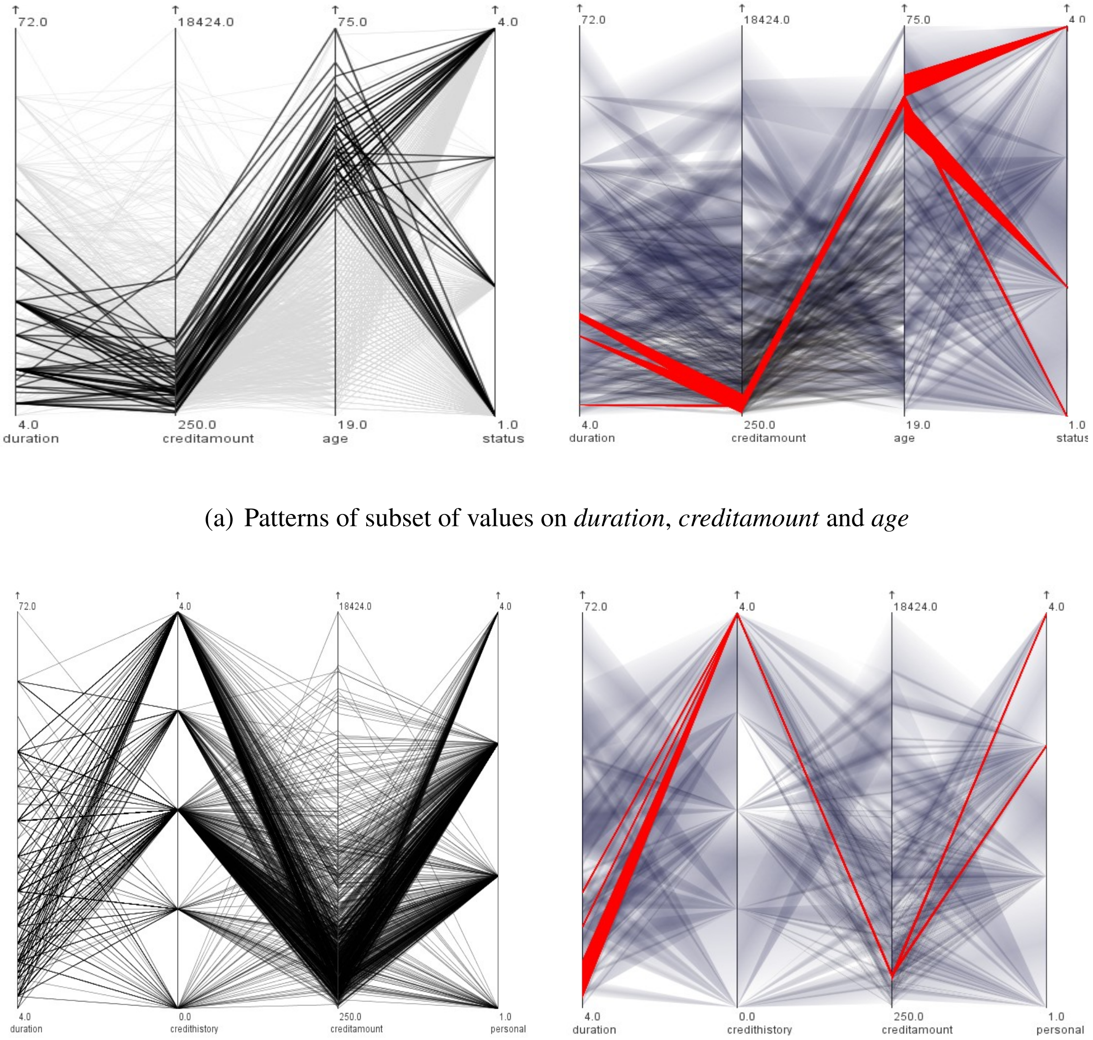
In case of reordering, if the sensitive dimension is not the last axis, then we have cluster splits on both sides of that dimension. But this reduces the utility to a great extent because we have the diversity on either side of the sensitive dimension leading to a lot of cluster splits. This reduces the meaningfulness of such a view. In the *German credit* dataset there is a single sensitive attribute, but our model can easily handle the case of multiple sensitive attributes by ensuring sufficient diversity in the corresponding axis pairs.



(a) *l*-diversity is not enforced because the *credithistory* value of 3 is not sensitive

(b) *l*-diversity is enforced because the *credithistory* value is 4 which is *sensitive*

Figure 49: l-diversity demonstrated in the rightmost axis pair. Top: l-diversity is not applied between *age* and *credithistory* because the value 0 is not sensitive as determined by the data owner. Bottom: l-diversity is applied because we want to protect clusters which have records with a sensitive 4 value. In this case, *l* is set to be 3, so we cannot tell apart among 3 different values of *credithistory* for the selected cluster.



(a) Patterns of subset of values on *duration*, *creditamount* and *age*

(b) Patterns between *credithistory*, *creditamount* and *personal*

Figure 50: Illustrating utility of clustered view with respect to different reordering configurations of the raw data.

1.4 Utility of Clustered View

Compared to the conventional data-based approach of multi-dimensional clustering, axis-pairwise clustering produces much more discernible clusters as we had shown in Figure 22. This fact is demonstrated by the graph in Figure 51 where cluster ranges are much smaller in axis-pairwise clustering used in our technique than the multi-dimensional clustering, thereby helping in clutter reduction.

By applying the *k*-members clustering algorithm, we protect the privacy of records, so that the user can only visualize cluster-level information. We cannot show individual record values in the form of lines, but the overall multivariate distribution among the different dimensions can still be visualized. In Figure 48 we see that for different values of *k* most relationships in the raw data are discernible in the clustered view.

We also demonstrate by showing different configurations of axis reordering that a user can still see the different trends and patterns. Figure 50(a) and Figure 50(b) show two different configurations of raw data on the left and the corresponding clustered configurations on the right. Patterns can also be seen between: a) *duration*, *creditamount* and *age*: low duration values corresponding to low credit amount and higher values of age are visible on mouse-over interaction in the clustered view; and b) *credithistory*, *creditamount* and *personal* exhibit a band of clusters that are seen in both views.

1.5 Different Reordering Configurations

The common ways to interact with parallel coordinates are to: a) hover over different lines to trace their path across the different dimensions, b) reorder the axes to see the patterns for different configurations of adjacent axes, and c) brush over different records on

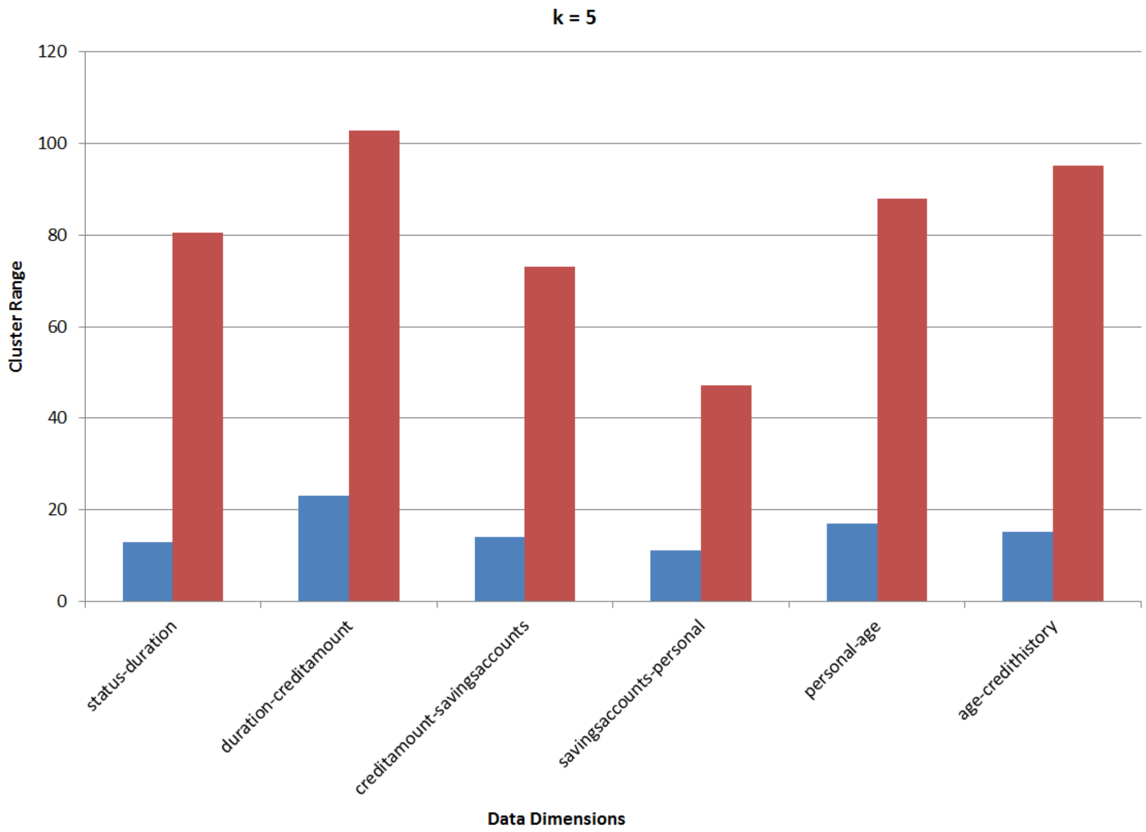


Figure 51: Multi-dimensional clustering (red bars), as opposed to axis-pairwise clustering (blue bars), leads to high cluster ranges that cause occlusion as shown in Figure 22.

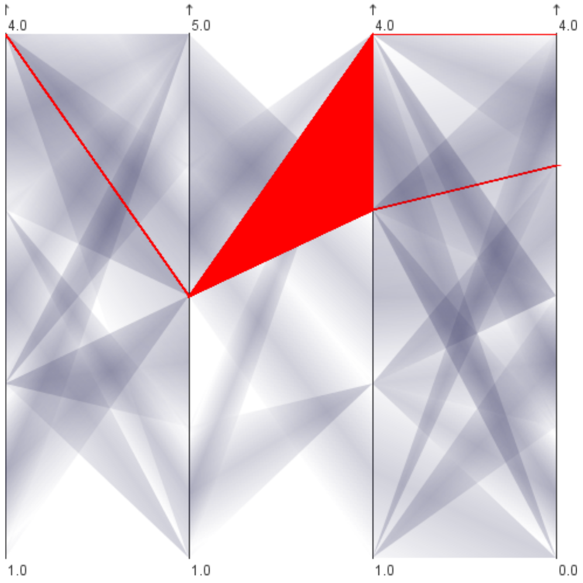


Figure 52: A configuration like this is avoided because categorical dimensions tend to produce clusters with splits at the edges, which when placed adjacent to a sensitive dimension, may lead to disclosure.

one axis to see the patterns of the subset of the data on other axes. Since the clusters already represent aggregated values, we ignore brushing for this paper and focus on the first two aspects.

High mutual information between two data dimensions A and B imply that the uncertainty about A is highly reduced in presence of B. We had shown earlier that mutual information is an effective screen-space metric for parallel coordinates. In the privacy context, we use lack of mutual information as a measure of high uncertainty [19]. If there is high mutual information between two axes, there might be a skewed distribution which makes it easier for an intruder to breach the privacy if he has some background knowledge about the person. This relates to attribute disclosure and is susceptible to both attack scenarios. When reordering the axes, we enforce the constraint that the quasi-identifier attribute with the least mutual information should be adjacent to the sensitive attribute. This ensures that even with interaction, the attacker cannot exploit the strong correlative effect between the two dimensions.

In case of a mix of categorical and numerical attributes, our technique puts numerical and categorical axes alternately, adjacent to each other. We avoid a configuration like the one shown in Figure 52 for two reasons. Firstly, cluster ranges can be very high between two categorical dimensions reducing the utility. Secondly, placing a categorical attribute adjacent to a sensitive dimension can reduce the intended privacy because the cluster edges represent actual data values and most values for a categorical cluster may lie on its edge and give away information. This relates to the prosecutor identification scenario, where an attacker knows an individual exists in the database and can select the appropriate cluster to gain knowledge about the attributes that describe the individual.

5.8.3 Measuring Privacy

Privacy is measured in terms of both encoding and decoding uncertainty as outlined in Table 1. On the encoding side, cluster range (γ) and cluster summary error (η) are higher in case of data-based clustering than visual clustering. This implies that precision and granularity uncertainty are higher in case of data-based clustering, signifying higher privacy. However, as shown in Figure 35b, decoding uncertainty due to overlaps, as measured by entropy, is much higher in case of visual clustering. Here, we observe that the uncertainty measured in terms of the overlap entropy of the axes (computed based on equation 7), increases for increasing k, which signifies higher uncertainty for guessing exactly which cluster a record belongs to. This metric not only measures the entropy in the static image, i.e, when highlighting/brushing is not available, but also covers cases where a user can select certain clusters by interaction. In our technique, when several clusters overlap on a pixel bin, we only highlight the smallest cluster. The attacker would thus not be certain whether a record which he/she is trying to guess the value of, belongs to that particular cluster.

In our experiments we have also found that the number of clusters, whose boundaries exactly coincide (the 'meets' condition in Allen's interval algebra), also decreases with increasing k, because with increasing k, there is a greater loss in precision of the values on the axis. This is beneficial from a privacy point-of-view because for smaller k, cluster boundaries meeting precisely on a pixel can potentially cause disclosure of the the data-points on these boundaries [38].

Traceability uncertainty is also much higher in case of visual clustering due to the higher average split count than in case of data-based clustering, where there are no cluster splits and the clusters have a one-to-one correspondence with continuing clusters on adjacent axes. As we discuss in Section 5.7.1, this creates the lack of *l*-diversity problem.

The effect of higher cluster range and cluster summary error in case of data-based clustering is offset by the higher overlap entropy and higher average split count in case of visual clustering. This is reflected in the graphs for net privacy (m_p) in Figure 35c, which shows that privacy achieved in case of visual clustering is higher on average, than data-based clustering.

5.8.4 Measuring Utility

Utility is expressed in terms of the different metrics for decoding uncertainty, mainly clutter and pattern complexity. High mutual information between adjacent dimensions maximizes utility. Figure 35a) shows the variation of mutual information for increasing k for four different axis pairs of the *Diabetes* dataset. The difference of mutual information between the two types of clustering is very pronounced due to the large cluster ranges in case of data-based clustering, which ensures that a two-dimensional data point is overlapped by multiple clusters on both axes in most cases.

With increasing k, it is expected that patterns in the visualization will get distorted and will be more difficult to discern. We are able to have better utility in terms of screen-space clarity in case of visual clustering because of less pattern complexity. For increasing k the effect of parallelism (computed based on our discussion in Section 5.6.7) gets reduced. However, the decrease happens in quite small increments and therefore does not degrade the visualization much. In our experiments we have observed that for converging-diverging structures, there is no significant variation with changing k. With very small k, like $k = 2$, the patterns are almost same as in raw parallel coordinates. With increasing k, the number of converging/diverging lines do not increase or decrease significantly. This is because we use convergence-divergence as a criterion for seeding the clusters and there is not much change in choice of seeds with changing k.

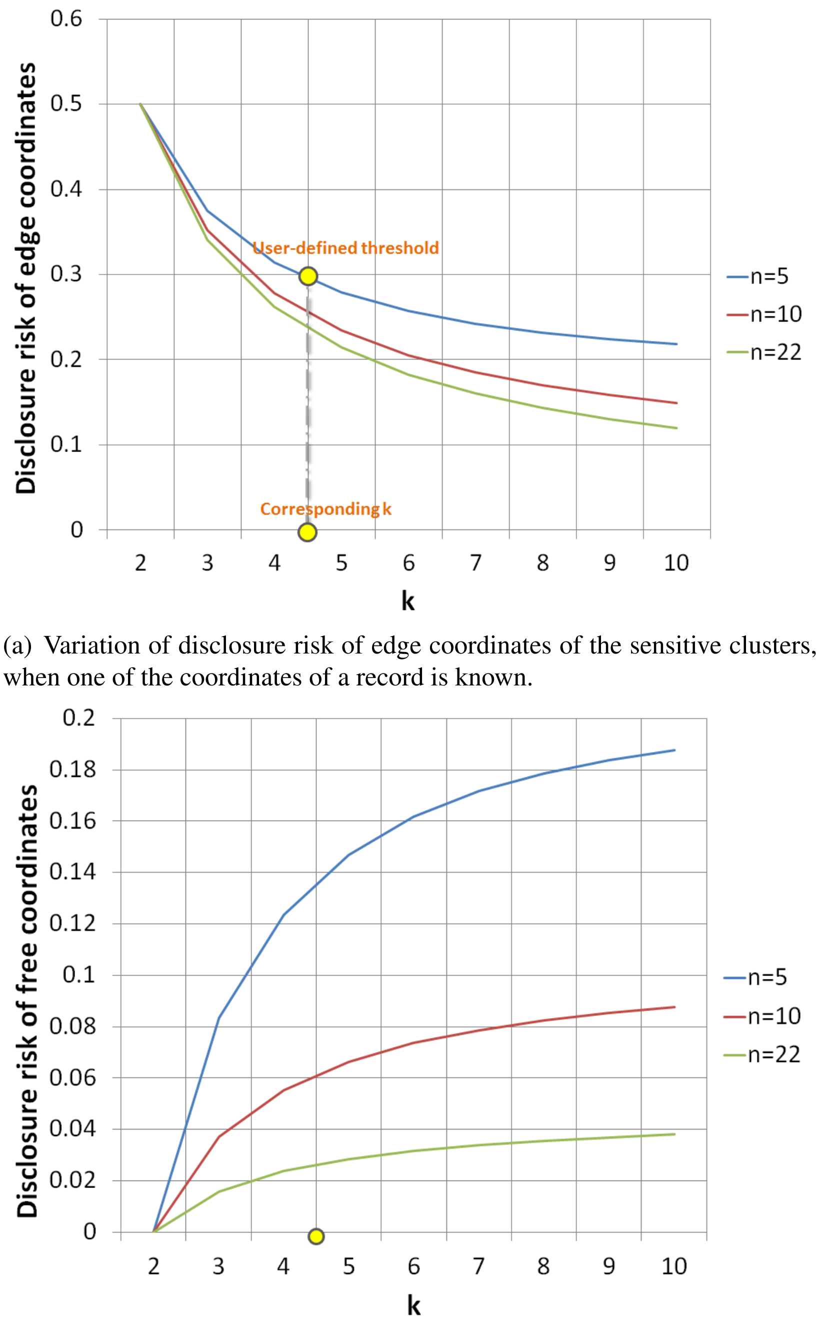
Due to higher traceability uncertainty in case of visual clustering, the utility is reduced. However, the effect of the other uncertainty components is much more significant when m_t is computed. The comparison for net utility is shown in Figure 35d, where we can observe that higher net utility in case of visual clustering than data-based clustering.

5.8.5 Devising an effective k

We have shown that all the various uncertainty measures do not increase or decrease monotonically with k. Especially for overlap entropy and mutual information, there is a high degree of variability across different k (Figures 35a and 35b). A similar pattern is also reflected in the graphs for m_p and m_t in Figures 35c and 35d. This implies that higher k does not necessarily signify higher privacy and/or lower utility in the screen-space. This is an important difference from privacy preservation in the data space, where variation of k is directly proportional to the privacy achieved or utility lost. Metric-based analysis of visual uncertainty, therefore, will enable visualization designers to choose the effective k based on the requirements for privacy and utility, that we discuss next.

1.6 Multi-criteria evaluation

The quantification of different types of visual uncertainty in the preceding sections serve as a feedback loop by providing a set of rules for iterative refinement of the design of



(a) Variation of disclosure risk of edge coordinates of the sensitive clusters, when one of the coordinates of a record is known.

(b) Variation of disclosure risk of free coordinates of the sensitive clusters, when one of the coordinates of a record is known.

Figure 53: By analyzing disclosure risks with respect to assumptions about the attacker's background knowledge, data owners can use thresholds for disclosure probability and choose a *k* accordingly. In this case, a data owner sets the threshold for diclosure probability to be less than 0.3 and therefore the initial choice of *k* is 4.

privacy-preserving parallel coordinates and scatter plots. We aim to perform a probabilistic analysis of the disclosure risks when an attacker uses background knowledge and interaction for breaching privacy. Explicit modeling of background knowledge is beyond the scope of this work, and we would like to point the readers to privacy literature in data mining for further reference in this area.

In Figure 45a the number of edge only configurations, computed using Equation 4 is plotted for increasing *k*. The decreasing number of edge-only configurations mean that the probability of the number of pivot configurations get higher with increasing *k*. While this direct correlation with *k* signifies more privacy, in Figure 45b we see that the probability of knowing the location of free points, computed by Equation 2, can increase with increasing *k*. While this can be counter-intuitive, given our model and metrics, it is not difficult to reason about this. One reason for this is the increasing number of pivot configurations with increasing k, and another factor is that the number of overlaps cases increase with greater k. This two factors lead to a higher probability of free points being known with increasing *k*. This necessitates additional measures, that would objectively compute the probability of disclosure given such uncertain patterns with higher k-anonymity.

Assumption about background knowledge: In this dataset, we show the utility of the metrics by assuming some knowledge about the dataset from the relationships of the dimensions. In many real world datasets, certain types of knowledge can be pre-conceived. For example, in a disease dataset, with the sensitive value being breast cancer, the attacker already knows about the most probable gender of the patient, and can use that knowledge for breaching privacy further. In this dataset, we make the assumption that the attacker knows, very young or very old people are more likely to have bad credit history. We examine some of the clusters that belong to people of this age group and has the sensitive value. In Figure 47 we see the lines for the raw data, and the clusters with increasing k. We can observe that with increasing k, the number of known locations of free points become higher, as annotated in the figure.

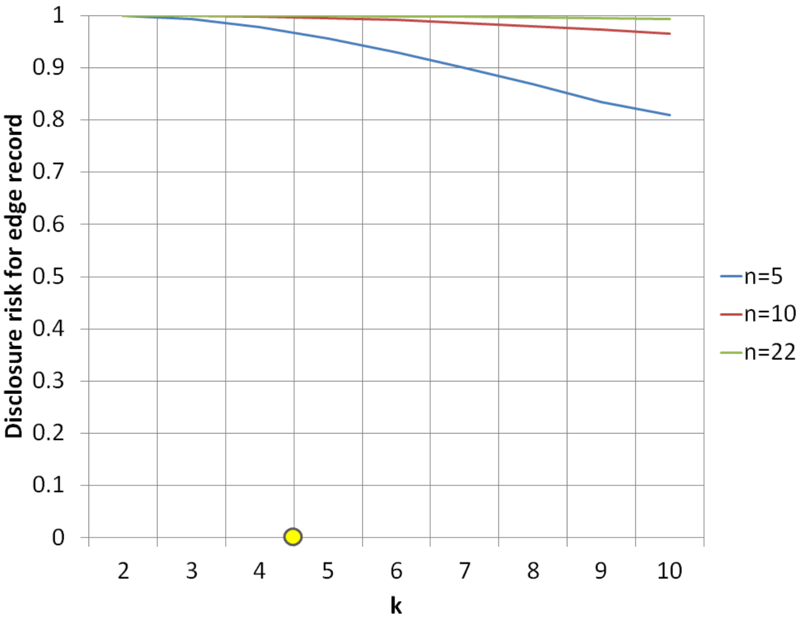
Sensitive Clusters: We use some of the computations to examine the probability of vulnerability given such background knowledge. For this dataset we compute the interquartile range of the cluster ranges on the axis for the sensitive clusters. Then based on equations 8,9 we perform a detailed analysis for deciding the k that can be used for an axis pair or for the dataset. In this case the average interquartile range for the chosen set of dimensions is 5 to 22, with the median being 10. Since disclosure risk is a function of both k and cluster range, we examine the nature of the functions for disclosure risk vs k, for the cluster range values of Based on these graphs the data owner decided a *k* to be used.

It should be noted that by selecting a smaller subset of other probable sensitive clusters, we are assuming that an attacker has been able to break through the inter-cluster uncertainties and are faced with the challenge of negating intra-cluster uncertainty. As discussed earlier, with reference to Figure 37, intra-cluster uncertainty can be reduced by knowing adjacent clusters, that is, by reducing inter-cluster uncertainty, and vice versa.

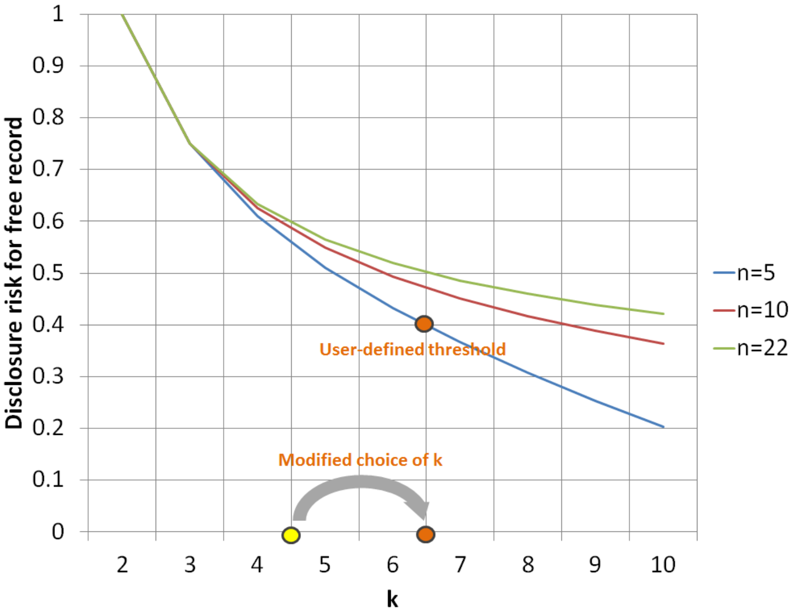
Analysis of disclosure risks: In Figure 53(a), the disclosure risk of edge coordinates vs k, when one of the coordinates is known by the attacker, is plotted. By looking at the graph, the data owner or the visualization designer sets a thresold of probability= 0.3, and he desires that the disclosure risk of edge points to be below this probability. It can be observed in the graph that $k = 4$ gives that desired probability for this dataset. In Figure 53(b), the disclosure of non-edge or free coordinates is plotted for varying k. While the disclosure risk for those points increase with increasing k, the disclosure probability is only 0.1 when $k = 4$.

However, when we assume that an attacker knows both the coordinates of a record, the uncertainty is much lower as shown in Figure 54(a). Due to precision and granularity uncertainty, there is still some uncertainty whether several records overlap on the identified record. This condition leads to attribute disclosure, but the uncertainty has a bearing on identity disclosure, as typically, knowledge about more than two quasi-identifiers are required to know the identity of an individual. As shown in Figure 54(a), the disclosure risk increases only gradually for increasing k for the edge records. Since increasing k to a very large value would degrade the utility, one would further inspect the vulnerability of the free points when a record is known. As shown in Figure 54(b), such disclosure risk decreases much faster with increasing k. After observing the graph, in this case a data owner decides to set the disclosure probability threshold to 0.4, and therefore k is changed from 4 to 6.

The high probability of disclosure of edge records, although poses a risk, as shown earlier in Figure 45a, the number of edge-only configurations decrease sharply with increasing k, which balances out the risk factor to some extent. As observed in the figure, for $k = 6$, the probability of the number of edge-only configurations is between 0.3 and 0.4. However, as an additional defense mechanism, data owners or visualization designers can optimize the layout with respect to adjacency of the dimensions, with respect to the output of the metrics, as we discuss below.



(a) Disclosure risk of edge records of the sensitive clusters, when both coordinates of a record is known, is quite high.



(b) Disclosure risk of free records of the sensitive clusters, when both coordinates of a record is known, is relatively lower and degrades much steeply with higher *k*.

Figure 54: By analyzing disclosure risks by assuming that an attackers knows both coordinates of a record, in this case a data owner changes his/her choice of *k* from 4 to 6, so that the disclosure probability of free records are less than the desired level.

1.7 Controlled Interaction

Adjacency of quasi-identifiers with sensitive attributes is critical from a privacy point-of-view [38]. Cluster splits at the edges, triangular clusters and low cluster resolution can all lead to disclosure of a sensitive value. A visualization designer needs to evaluate which attributes can pose low risk, by using the risk quantifications, for example the one described in Equation 14 and thereby restrict reordering in parallel coordinates or scatter plot matrix. The cluster configurations should also be evaluated using the knowledge about free points as given by Equations 18 and 19 and brushing can be restricted for clusters with edge-only configuration.

In particular we evaluate which axis pair formed by a chosen dimension among the quasi-identifiers have the fewest clusters with edge-only configurations and no-record areas. That dimension is selected to be adjacent to the credithistory dimension. In this case the dimension is age. In particular, two categorical dimensions are not placed adjacent to each other, as most of them have fewer values on the axis, leading to edge-only configurations and no-record areas on the clusters.

1.8 Choice of visualization technique

Scatter plots and parallel coordinates are comparable in terms of the amount of privacy-protection that can be achieved by clustering. One advantage of parallel coordinates is that since there are the additional overlaps between axis, as discussed in Section 5.6.3, they can create more uncertainty due to clutter than scatter plots. This implies visual quality of scatter plots is higher than that of parallel coordinates. Parallel coordinates, on the other hand, enables one to trace the path of the clusters across multiple dimensions. In case of large average number of splits per cluster, getting an idea of the multivariate distribution would be difficult. Moreover in parallel coordinates one can know about the cluster configuration from the overlaps across axis pairs, which is difficult in scatter plots. So a designer has to carefully analyze the anatomy of splits before publishing the visualization.

Encoding uncertainty is identical in parallel coordinates and scatter plots as we get the same values for the metrics. On the decoding side, however, clutter and traceability produce different results. Although clutter is less in scatter plots than parallel coordinates, as was demonstrated in Figure 32, it cannot be readily concluded that in terms of perception , the former is better. This is because the degree of distortion of the visual structures is higher in scatter plots than parallel coordinates. In scatter plots, points (zero-dimensional entities) are transformed to rectangles (two-dimensional entities). In case of parallel coordinates, lines (one-dimensional entities) are transformed into polygons (two-dimensional entities). Therefore the structural properties are much less distorted.

CHAPTER 6: VISUALIZING HIGH-DIMENSIONAL, TIME-VARYING DATA

6.1 Problem Description

Visualization of time-varying, multivariate, large data is a complex problem because of multiple challenges, like maintaining scalability and fidelity of the visual representation, and communicating the temporal changes effectively through an interactive visual interface. Scalability of the technique is essential for efficiently handling the large number of data points. Fidelity is important for capturing the salient functional relationships among multiple variables. Visual communication of the temporal trends through an interactive interface in a perceptually beneficial manner is another significant challenge. Visual approaches that integrate the automated methods with user-driven analysis, in the context of time-varying data, are still in their infancy. To fill this gap, we propose an integrated temporal parallel coordinates-based framework that combines screen-space metrics for investigating multivariate temporal relationships and multiple coordinates views for facilitating interactive user-driven analysis.

The use of screen-space metrics belongs to the explicit encoding category [57] of enabling visual comparisons across objects. Since the metrics are computed based on the screen resolution and not on the cardinality of the data space, they are more scalable than purely data-based statistical metrics. Data abstraction in the form of metrics results in loss of data fidelity in exchange for more focused information. By picking metrics that are motivated by perceptually meaningful structures, we make it easier to relate them to the information displayed on screen. Moreover, we facilitate contextualization of the metrics by using coordinated multiple views, which supports the sense-making process on the analyst's part. The key analysis questions our system helps to address are summarized as follows:

Q1, Section 6.5: Which variables show stable or unstable behavior over time?

Q2, Section 6.6: How does the univariate and bivariate distribution of a variable change over time?

Q3, Section 6.6: What multivariate patterns are there and how do they change over time?

Q4, Section 6.7: Which subset of dimensions and data points exhibits the most interesting patterns in which particular time steps?

Following Munzner's nested model [88] for visualization design and validation, the characterization of these domain-specific questions enables us to focus on the other three levels proposed in the model, that are described as follows:

Abstraction design: we have extended previously proposed screen-space metrics to support analytical abstraction for temporal data analysis and devising new ones for quantifying properties and temporal trends.

Encoding design: while we use parallel coordinates as the main view, our system constructs univariate, bivariate and multivariate views of the data to gain different perspectives on temporal trends.

Interaction design: we enable analysts to build their own parallel coordinates configurations from rank-ordered paired dimension sets and explore subspaces within the data with the help of semantic brushing based on the metrics.

Algorithm design: we have designed algorithms for the metrics, their visualizations and for guiding the multi-way interactions among the views.

Multi-level validation: we illustrate our approach qualitatively, through several examples using a bioremediation dataset and a detailed case study and discussion of performance at the end.

6.2 Reducing Pattern Complexity Uncertainty Through Metrics

The choice of metrics is motivated by Amar and Stasko's recommendation [6] of the general analysis tasks that a user performs with a visualization. Among those, characterizing univariate distribution, finding bivariate relationships in terms of correlation, detecting hidden clusters in subspaces and examining outliers are relevant to the work reported here.

For quantifying univariate distribution, our goal is to characterize the data density in terms of the locus (where, on the axis, most data values are located) and randomness (amount of disorder or uncertainty among the values). For this purpose we propose two new metrics for characterizing the data distribution, the *density median* and *axis entropy*, both of which are computed from the one-dimensional axis histogram. Density median is indicative of the degree of skewness of an unimodal distribution. The median and entropy metrics are not restricted in their application, to parallel coordinates only. They also apply to point-based representations like scatter plots. In terms of dispersion or data disorder, entropy and variance are popular statistical measures. While there is no direct correlation between entropy and variance, it has been shown that entropy is more flexible in capturing dispersion as its location is independent of the mean, unlike variance [47].

When there are multiple modes, median alone is not an accurate estimator of density. A multimodal distribution implies data has a higher likelihood of being clumped at subspaces. To explore if there are hidden clusters in subspaces that are not distinguishable from the overall patterns, we have developed a clumping metric, based on the two-dimensional distribution based on an axis pair. This is especially of relevance for temporal data, because at many time steps, certain data-points tend to cluster/clump together in local neighborhoods. Clumping metric captures this behavior.

For quantifying the change in relationship between adjacent dimensions, we quantify the linear relationship and relative information content that are computed based on the one-dimensional distance histogram and two-dimensional histogram respectively. For the former, we adapt the parallelism metric and for the latter we use mutual information between dimensions. In addition, number and angle of line crossings are also used to find inverse correlations and clusters respectively.

In the following sections, we first discuss general applications for high-dimensional data in terms of dimension order optimization and then discuss our system for analyzing time-varying, multivariate data.

6.3 Dimension Order Optimization

Using the above metrics, we are able to optimize the display for the analyst. In general, finding an optimal ordering of axes for parallel coordinates is equivalent to the traveling salesman problem, and thus NP-complete [71]. Using a branch-and-bound algorithm and considering the special properties of parallel coordinates, we are able to find optimal solutions in much less time in general. Our binned data model of parallel coordinates also reduces many the computations (reducing the impact of the number of data items) and even

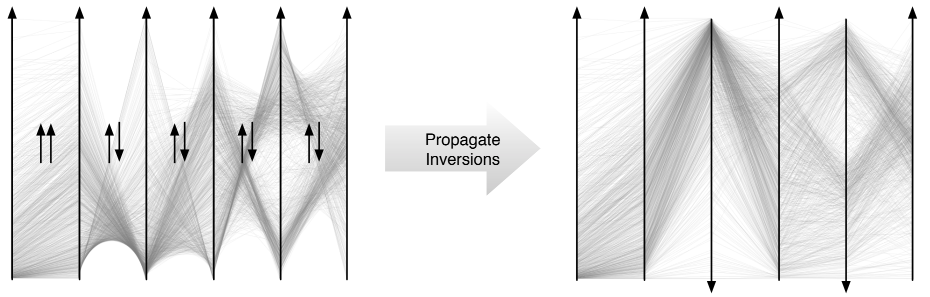


Figure 55: Axis inversions are handled locally during the optimization, and only considered per axis pair. They are later propagated through all dimensions from left to right to determine which axes end up being inverted.

handles axis inversions as local decisions, leading to a very efficient solution.

We use a branch-and-bound algorithm to find the optimal order of axes. We first build a matrix of all axis pairs and the cost associated with them. The cost can be a combination of several metrics, using weights selected by the user. This computation is only performed once, and is the only step that depends on the number of records in the dataset. All subsequent steps are performed on the basis of this matrix and thus only depend on the number of dimensions.

6.3.1 Axis Inversions

Axis inversions are handled per axis pair as part of building the cost matrix. For each axis pair, the cost for both the inverted and the non-inverted situation are computed across all the desired metrics. The lower value is used in the matrix, and the program records which of the two values that was.

Axis inversions can be handled locally because we consider them only between the two axes in each axis pair. Inverting one axis pair does not have an immediate effect on neighboring axis pairs. Only once the optimal solution has been found, the inversions are propagated across the axes from left to right (Figure 55). Since users more likely prefer axes

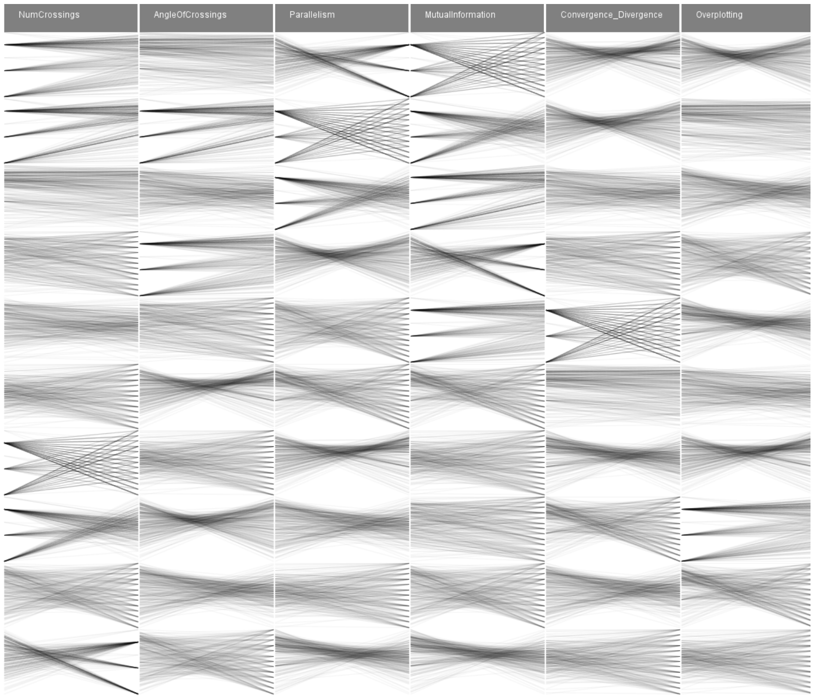


Figure 56: Parallel coordinate views of the cars dataset, ranked by the ascening order of the different metrics. In our implementation, the user is shown the names of the axis when mousing over a plot, and the display also highlights all instances of the same plot the user is pointing at.

pointing up, the program also performs a global inversion that flips all axes if the majority of them are pointing down after the propagation.

While axis inversions can dramatically reduce the number of crossings and increase parallelism, they also make reading the visualization more difficult, because they require the analyst to look up the axis orientations and remember which ones are inverted and which are not. The optimization therefore provides the option not to use inversions in the process.

6.3.2 Branch-and-Bound Optimization

We implemented the optimization as a branch-and-bound algorithm that uses a priority queue and best-first search. A key issue in branch-and-bound implementations is how tight the bounds can be estimated when the decision is made about whether to cull a sub-tree or

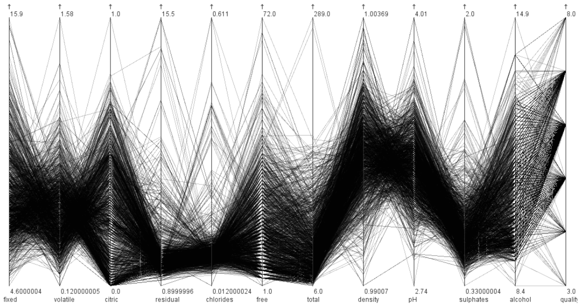


Figure 57: Initial layout of the wine dataset.

not. In our case, these estimates are based on a full cost matrix, and are very precise.

Based on our case studies, we find that the best-first search very quickly finds the optimal solution after only a few complete solutions (i.e., having walked through to a leaf of the search tree). This is not surprising given the reduction of the metrics to axis pairs, rather than looking at the entire visualization at once. In most cases, the optimization ends up only picking the best next axis from the ones still available, rather than having to evaluate a much larger number of permutations.

6.3.3 Performance

For practical use, metrics need to be calculated in reasonable time. The fact that all Pargnostics metrics are based on histograms simplifies computation, and reduces the real-world complexity by reusing intermediary results.

All histograms are created in one pass, with complexity of $O(n)$ (n being the size of the dataset). In our implementation, the one- and two-dimensional histograms are computed at the same time, which further reduces overhead. The calculation of line crossings (and crossings angles) is the most computationally expensive, with a complexity of $O(h^4)$ or $O(n^2)$ (where h is the size of the display, see Section 4.1.1). All others are $O(h^2)$ or $O(h)$, with $h << n$ (i.e., $O(n)$ would in many cases be more expensive in real-world terms).

The optimization requires the calculation of the relevant metrics (which the user picks) for all axis pairs. They are used to build a cost matrix, which the branch-and-bound optimizer needs for building candidate solutions. Once the matrix is built, the optimization is very efficient. For a dataset with 12 dimensions, it typically queues several ten thousand partial solutions, but only evaluates a few (less than ten) complete ones. That means that our culling criterion is very good, and that the best-first search has a high chance of immediately finding the best overall solution.

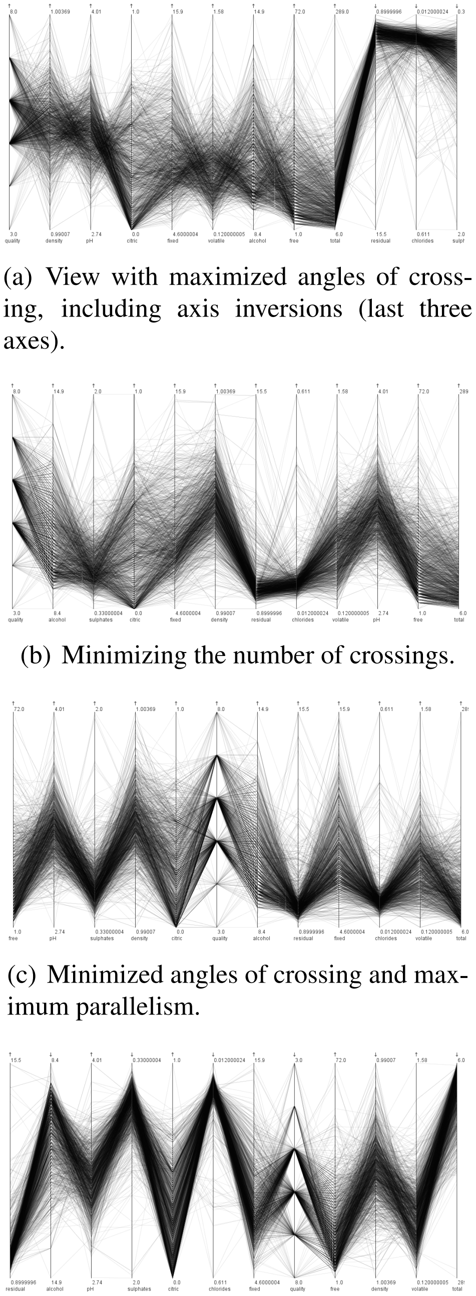
Multiple metrics are handled in the matrix building stage, and have no direct bearing on the performance of the optimization itself. Having to evaluate several metrics of course slows down the matrix construction. Axis inversions similarly are handled at the first stage and do not increase the complexity of the search space. Since our implementation efficiently computes the inverted value together with the non-inverted value for most metrics, the added cost for considering inversions is very low.

6.3.4 Examples from Datasets

We demonstrate the use of our metrics with two example datasets, one describing car models and another one about wines.

6.3.5 Cars Dataset

The cars dataset [52] consists of 392 values and six attributes: *MPG*, *horsepower*, *cylinders*, *weight*, *acceleration*, and *year*. Pargnostics metrics can be used for both user-centered and automated optimization. Based on the common scatterplot matrix, we have developed a parallel coordinates matrix view that shows all combinations of axes in the lower left half of the matrix with both axes pointing in the same direction, and with inverted axes in the



(a) View with maximized angles of crossing, including axis inversions (last three axes).

(b) Minimizing the number of crossings.

(c) Minimized angles of crossing and maximum parallelism.

(d) Maximized number of crossings and minimized angles of crossing, and including inversions.

Figure 58: Optimization for different sets of criteria using the wine dataset.

upper right. A similar matrix view shows the distance and angle histograms for each axis pair (Figure 59).

The small parallel coordinate plots show the overall structure of the axes pairs, while the histograms give some insight into some of the metrics. In our prototype implementation, the user can construct the parallel coordinates display by picking axis pairs from any of the matrices. There is also a ranked view of parallel coordinate axis pairs (Figure 56), which allows the user to directly find parts of the display that exhibit certain structures.

Selecting from any of those views is based on visual structure rather than particular data dimensions, and thus frees the user from preconceived ideas about the data. Adding dimensions is typically done by selecting them by name; in our model, the display can be built directly from interesting visual structures. The resulting display makes maximum use of the power of the visualization.

6.3.6 Wine Dataset

The wine quality dataset [52] consists of 4898 rows and 12 dimensions: *fixed acidity*, *volatile acidity*, *citric acid*, *residual sugar*, *chlorides*, *free sulfur dioxide*, *total sulfur dioxide*, *density pH*, *sulphates*, *alcohol*, and *quality*. Figure 57 shows the initial view of the data set, with the dimensions in the order in which they appear in the data file.

Single-Metric Optimization

In Figure 58(a), parallel coordinates is optimized by high crossing angles, taking inversions into account. As we can see, most of the lines between the axes tend to cross at close to 90 degrees and this helps to reduce clutter. In Figure 58(b), parallel coordinates is optimized by minimum number of crossings. This tends to produce a less cluttered display,

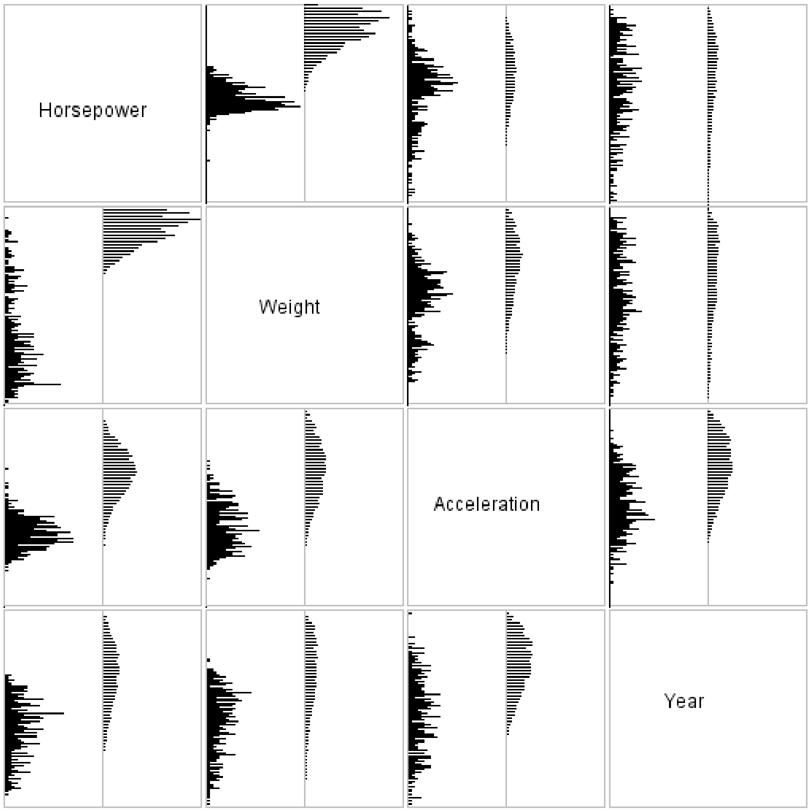


Figure 59: Histogram matrix for the cars dataset. The left histogram in each cell shows the distance, the right one line crossing angles.

but at the same time produces high parallelism as observed on *density* and all dimensions to the right of it.

Multi-Metric Optimization

In Figure 58(c), parallel coordinates is optimized by low crossing angles and high parallelism. This configuration produces clusters on either side of the categorical quality variable. Thus this configuration is not only useful to see the correlations and clusters, but also enables one to see where the intense concentration of records, or the mode [118], occurs. In Figure 58(d), parallel coordinates is optimized for high number of crossings and low crossing angles, taking axis inversions into account. There are many inverse correlations observed between most of the axes. A large number of crossings generally tends to produce inverse correlated structures and with low angles, the clusters can be seen clearly.

6.4 System Design for Visualizing Temporal Data

Our system [41] is composed of three coordinated views that enable analysts to explore temporal data from different perspectives. Metrics, abstracted in the form of time-series, are displayed in those views. Interaction with these views enables analysts to seamlessly navigate through interesting time steps and data dimensions. In this section we provide an overview of the metrics and introduce the different views.

6.4.1 Multiple Views

We use the screen-space metrics as the basis for designing coordinated multiple views (Figure 60) that show different properties of the data. The views facilitate exploratory data analysis based on: a) getting an overview of the temporal trends that evolve over time and b) interactively explore patterns at time steps of interest and drill down to details. Our design

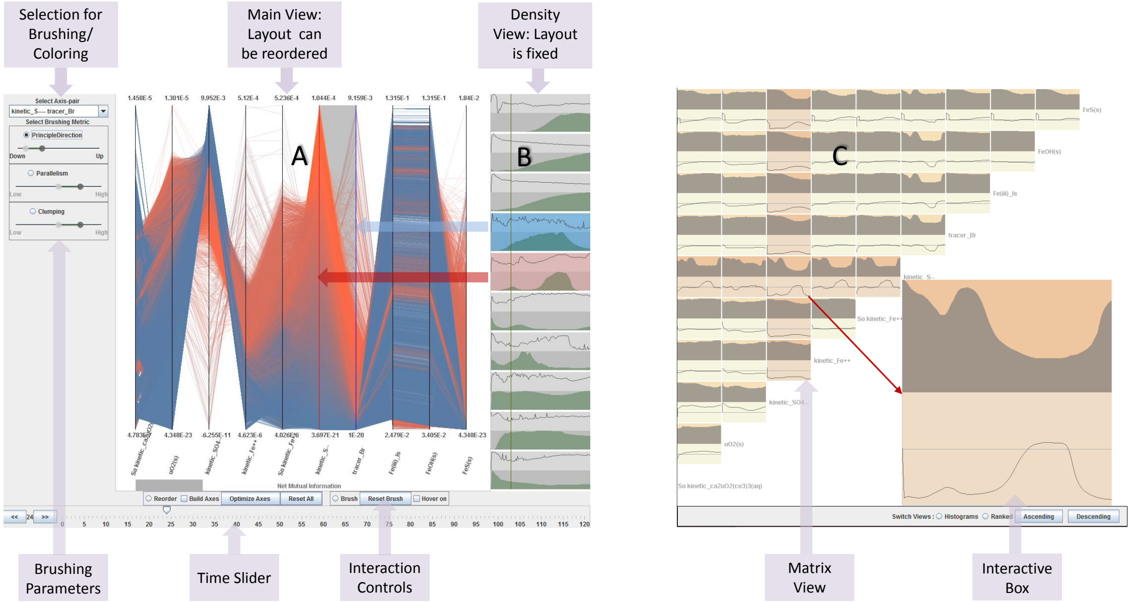


Figure 60: The different components of our parallel coordinates-based framework. The three views of the data are: A) Main view, which is a time-varying view and re-orderable, B) Density view, which shows univariate temporal distribution with the selected axis pair being highlighted, and C) Matrix view, which shows the bivariate correlations as time-series.

of the coordination among the views follow Schneiderman's visual information seeking mantra [102] by enabling the analysts to seamlessly switch between gaining overview and exploring details.

For a time-varying dataset D with n records, d dimensions and t time steps, the cardinality of the dataset is given by $|D| = n * d * t$. In the context of the bioremediation dataset that we use in this work, $n = 96000$, $d = 10$ and $t = 120$. So $|D| = 115,200,000$. For such a high cardinality, a conventional parallel coordinates representation of data dimensions on the vertical axes and the records as poly-lines is not a good fit due to clutter and scalability issues. To overcome these problems, we use visual abstraction in the form of screen-space metrics for creating an effective temporal summary that conveys the salient time-varying behavior with respect to all the data dimensions. Using these metrics, we build a *meta*

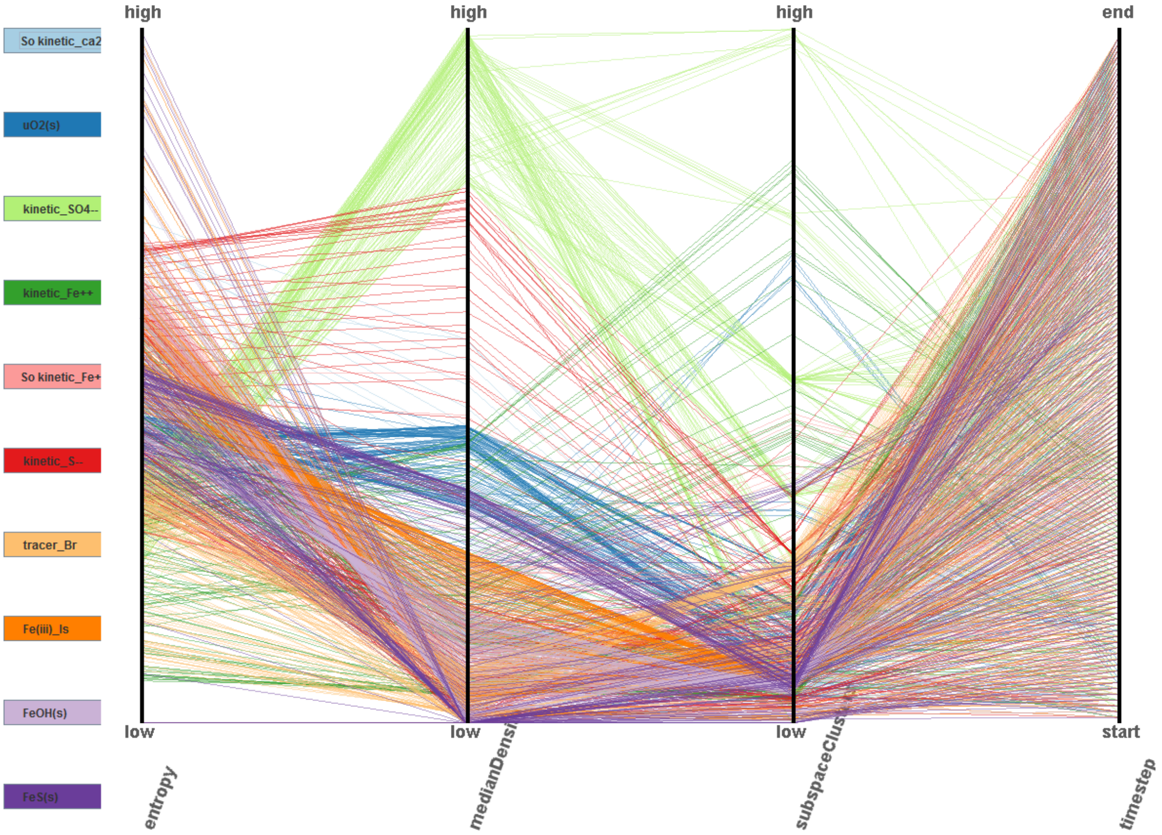


Figure 61: Meta Parallel Coordinates show the dimension-level view: The vertical axes represent the metrics while each polyline represent the value of the metrics for a given data dimension at any given timestep. Users can filter by a single dimension or multiple dimensions by making selections in the left panel. Specific time steps can also be selected by brushing.

parallel coordinates view (Figure 61) [42] in which the metrics are represented by the vertical axes and each poly-line represents the values of those metrics for a color-coded data dimension, for a particular time step. Thus this view serves as a *meta* view for the conventional parallel coordinates display. We thus reduce the number of data points in the meta view, to $d * t = 1200$ data points and thus we alleviate the scalability problem. The MPC is coordinated with the conventional parallel coordinates view (Figure 60A) that shows the data plot for a particular time step. Interaction between the MPC and the conventional view enables a user to explore the data at multiple levels of granularity by seamlessly switching between exploring temporal changes at the dimension level, and then looking at the details, at the record level. The density view (Figure 60B) and matrix view (Figure 60C) comprise the data views that represent summarized representations of temporal behavior.

This approach of separating the dimension level view (meta parallel coordinates) and the record level view (conventional parallel coordinates) is similar to that proposed by Turkay et. al. [113] who suggested a dual analysis model involving the dimension space and item space, by using standard statistical measures computed in the data space. In this work we use screen-space metrics, as they are computationally more efficient than purely data-based metrics. This is because after the initial data transformation the screen-space metrics are affected by only pixel resolution and are independent of the data cardinality.

6.4.2 Interaction

Exploration of temporal changes is facilitated by multi-way interaction. Relating patterns with the corresponding metrics is achieved by interaction starting in the main view, while using metrics to drive the analysis process is achieved by interaction beginning in the density view and the matrix view. These are explained in detail as follows:

From Main View: Interaction originating in the main view is labelled as *A* in Figure 62. The main view is used to explore the patterns in detail. When an analyst selects an axis pair (*b*-*d* in the figure), the color gradient is applied and the axis pair is highlighted. Additionally, the corresponding boxes *b* and *d* in the density view, are also highlighted. Note that color of the highlighted boxes correspond to the color of the axes: we use red as the color of the left axis and purple as that of the right. At the same time, the box *b*-*d* is highlighted in the matrix view and enlarged box is drawn, with the vertical line representing the time step also being drawn. The time-slider always gets updated to the current time step.

From Matrix View: Interaction originating in the matrix view is labelled as *B* in Figure 62. Each box in matrix view can be selected and observed in details at the bottom right. The enlarged box is interactive with the other two views. The configuration for the corresponding time step is shown in the main view and axes get reordered according to the adjacency of the values selected, with a grey highlighting for the corresponding axis pair. Axis orientation of bottom to top translates to a left to right orientation in the main view. The corresponding boxes are also highlighted in the density view and the vertical line denoting the time step is shown.

From Density View: Interaction originating in the density view is labelled as *C* in Figure 62. When a box *b* is selected, in addition to that box, the one that is adjacent and to the right of *b*, that is *d*, is highlighted too. We treat the selected axis as the left one in the main view and color it as red. The corresponding axis pair in the main view is highlighted, color gradient is applied and time-slider is advanced to the time step computed from user's mouse coordinates. The matrix view is similarly updated like in case of the interaction

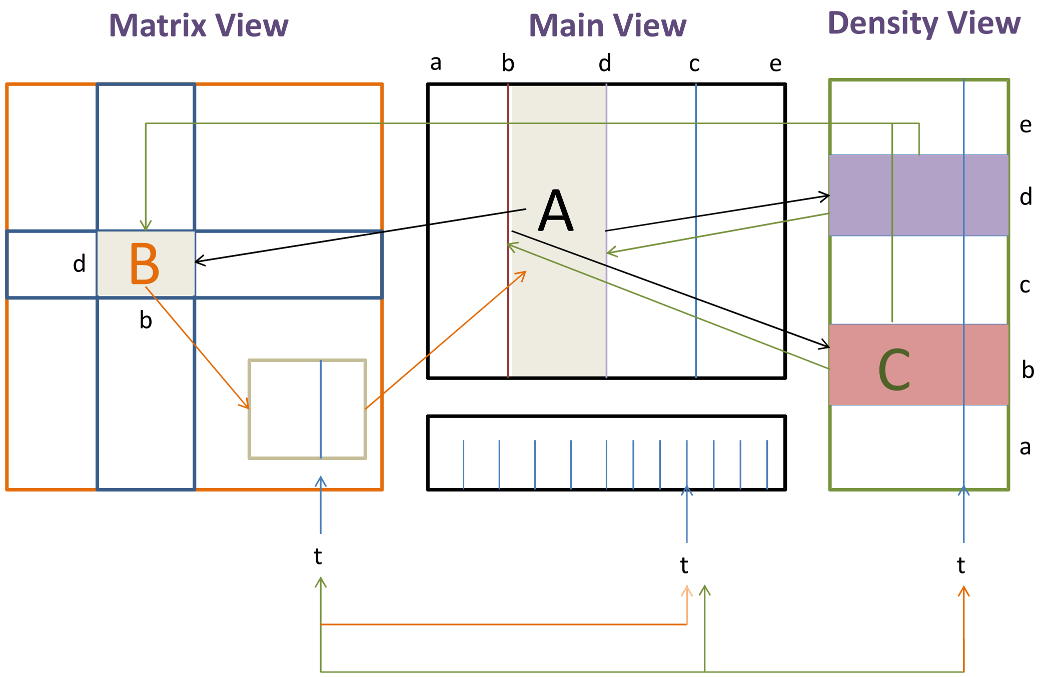


Figure 62: Coordination among the different data views. Arrows are differently colored according the origin of interaction. Black: Interaction starts in main view, Blue: Interaction starts in matrix view and Red: Interaction starts in density view.

from main view.

In the following sections we describe the metrics, views and the associated interactions among them in detail, with respect to the questions (Q1, Q2, Q3, Q4) we set out to address using our tool. We use simulation data from bioremediation experiments for illustrating our approach and refer to some of the variables in course of describing our system. We present the details about the dataset in Section 6.8.

6.5 Identifying Stable and Unstable Behavior (Q1)

Finding stable and unstable variables (Q1) gives the analysts an overview of the overall temporal behavior. We define stability as the degree to which the data distribution remains unchanged over a period of time. This question is addressed by metrics for data density and getting an overview from the density view and main view, that are discussed below.

6.5.1 Metrics for Data Density

Both the density median and axis entropy metric are based on univariate properties, meaning, they are independent of axis adjacency in parallel coordinates. The two components of our data density metric are : the density median determining the locus of skewness, and the nature of density in terms of data disorder or randomness.

Computation: Density median ($\tilde{D}$) is computed from the median of the frequencies of the pixel-bins in a one-dimensional axis-histogram. The location, that is the pixel coordinate of the median ($\tilde{\beta}$), is then plotted over time. A high value of the median at a particular time step means dominant values at that time step are the high ones and a low value means dominant values are the low ones.

Entropy measures the uncertainty or disorder within the data values and we use Shannon entropy [32] to capture this characteristic. We consider each axis in the screen-space as a random variable. We use the probability of intersection of a data record with a pixel-bin, computed from the frequency of each pixel-bin on an axis as the basis for computing Shannon entropy. Entropy for an axis (A) in terms of its pixel bins ($a1, a2, \ldots a_h$) is given by:

$$H(A) = -\sum_{k=1}^{h} p(a_k) log(p(a_k)) \tag{22}$$

where $p(a_k) = \frac{1}{\beta_k}$. When the entropy (H(A)) is plotted over time, the trajectory of the time-series indicates the overall stability or instability of the variable.

Visualization: Figure 16 shows the main locus of data density which is implied by the concentration of the lines. If the majority of the lines start/end towards the top of any axis (in this case, the left axis in the left image), then data density is skewed towards the higher values. Mostly uniform distribution of the lines, as shown in the left image in Figure 17 do not convey any significant patterns within the data. This implies there is more uncertainty in the data and therefore higher randomness. The right image shows a trend associating high and low values on either axis. The increase and decrease in data disorder/dispersion is captured by our axis entropy metric. Line graphs that are more or less flat, like the ones in the top two boxes in Figure 60B indicate there is no significant change of behavior for the variable, while a jagged graph indicates potentially interesting behavior. Moreover, high entropy values indicate a uniform distribution and low entropy indicates more skewness and also more recognizable patterns.

6.5.2 Density View

The density view is composed of sets of vertically stacked boxes (Figure 60B), where each box corresponds to a dimension and contains a line-plot and an area-plot. The line plot represents the axis-entropy and the area represents the density median, time-axis being horizontal. The configuration of this view is invariant to the order of axes in the main view. Different colors (red for left axis and purple for right axis) are used to represent the selected axes. Even without interaction, the entropy and density median plots give an overview of which variables are stable or unstable.

Figure 63 illustrates how the patterns are interpreted using the density view. Table 8 lists the formulas and names of the chemical species that we use in our application. The back-

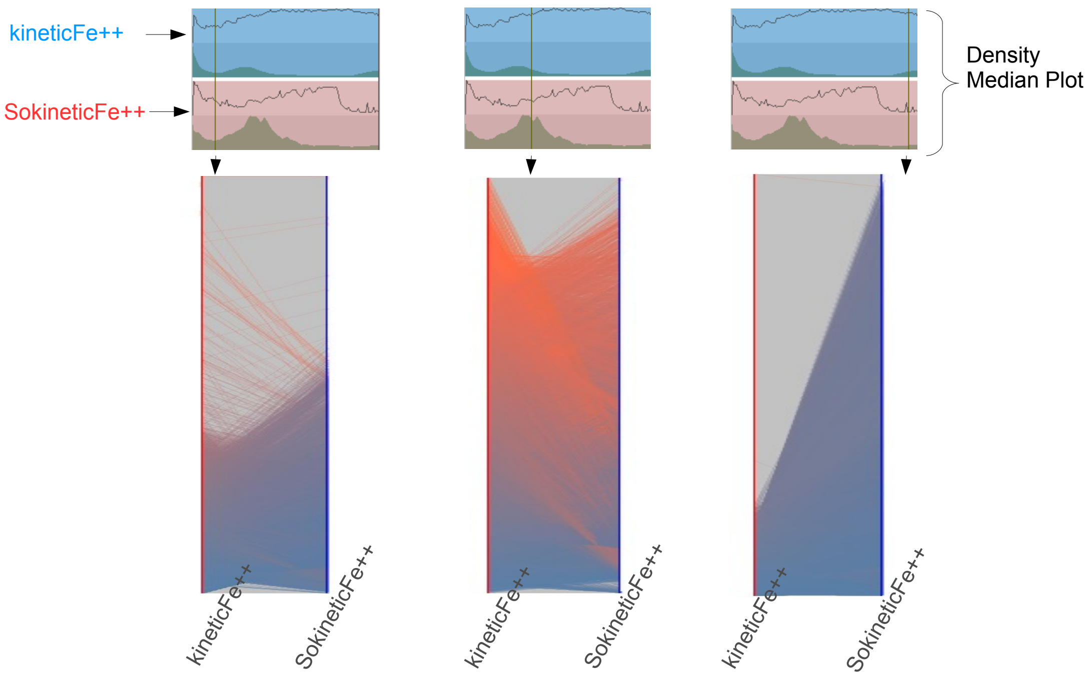


Figure 63: **Illustrating the data density metrics**: Three different configurations of the axis pair (kinetic iron and sorbed kinetic iron) on interaction with the density view. Each blue box represents the density median plot for kinetic iron and the red box represents the same for sorbed kinetic iron. Low entropy clearly leads to more recognizable patterns.

Chemical Formula	Chemical Name
$So\ kinetic\_ca2uo2(co3)3$	uranium carbonate complex
$UO_2(s)$	solid uraninite
$kinetic\ SO_4$	kinetic sulfate
$kinetic\ Fe++$	kinetic iron
$So\ kinetic\ Fe++$	Sorbed kinetic iron
$kinetic\ S--$	kinetic sulfide
$tracer\ Br$	tracer bromide
$Fe(iii)\ Is$	iron sillicate
$FeOH(s)$	solid iron hydroxide
$FeS(s)$	solid iron sulfide

Table 8: Chemical formula and name of the different variables, whose interactions are studied through the bioremediation experiment.

ground about the generation of the data and the experiment are explained in Section 6.8.

The three boxes and their parallel coordinates representation are for three different time steps for kinetic iron and sorbed kinetic iron. For the first case, kinetic iron (red box) exhibits low density median and low entropy and same for sorbed kinetic iron (blue box). This is represented by highly dominant and less dispersed blue lines (signifying dominance of low data values). In the second case density median for left axis is higher and axis entropy for right axis is higher. This is demonstrated by more orange lines, that signifies higher concentration of kinetic iron being dominant. While high entropy on the right axis leads to a high dispersion among the lines. Had the entropy been low, we would have seen a cluster of lines going from top of the left axis to bottom of the right axis. In the last case, low density median and low entropy is indicated by the dominant blue lines originating from kinetic iron. The difference from the first case is the lines are highly dispersed, shown by high entropy value for the right axis.

6.5.3 Main View

The main view helps in addressing *Q1* by enabling the analysts to explore the features shown by the density view, that is, going from overview to the details. In this view we show the default parallel coordinates layout for our dataset, as shown in Figure 60A. The configuration is synchronized with the time-slider.

Global Scaling: We choose a global scaling for the variables, i.e., we compute the maxima and minima for the different dimensions over all the time steps and scale the data points accordingly. This helps us in handling ranges that can vary a lot from an initial time step to a later time step in a particular domain, which enables us to show how trends change within a fixed data range.

Color Gradient: We use a continuous color gradient of blue to orange, to indicate the transition from low to high values on the axis. The color gradient is applied to an axis pair selected by a user, which is the pair with kinetic sulfide (kinetic S–) and tracer bromide (tracer Br) in Figure 60A. The higher concentration of blue lines on most other axes give an overview of difference in high and low concentrations of multiple variables even without adjacency. This also helps in reducing clutter when there are a large number of line crossings and makes the trends stand out.

Axis Order: Appropriate axis-order is important for revealing interesting patterns in parallel coordinates. We use the mutual information metric which helps find non-linear correlations [37], for guiding the adjacency of the axes. The grey horizontal bar below the main view indicates the sum of the mutual information values of the individual axis pairs. If at any time step, the net mutual information content of the default layout is very low, as indicated by the width of the bar, user can choose to optimize the configuration by maximizing the metric. The underlying optimization algorithm is a branch-and-bound algorithm. This way we choose a user-driven approach to optimize the layout.

6.6 Exploring Temporal Changes from Different Perspectives (Q2, Q3)

Temporal changes include change in univariate distribution and bivariate/multivariate relationships. We illustrate how we address the problem of quantifying these changes and relating those to the multivariate properties of the data at different time steps (Q2 and Q3). These questions can be addressed using the metrics for correlation, data density, and multi-way interaction among the views. Moreover, the ability to perform semantic brushing on records by selecting the upper and lower ranges of the metrics, helps analysts explore patterns within subset of data points. In this section, we first introduce metrics for correlation and the matrix view and then describe the different interaction mechanisms.

6.6.1 Metrics for Correlation

We use the parallelism metric[37] to quantify the linear relationship between dimensions. The parallelism metric is composed of two elements: the range depicting the degree of correlation (if most lines conform to the trend or are loosely scattered) and the median or principal direction from left to right between two axes indicating if parallel lines are going up, down or staying horizontal.

Computation: To compute parallelism, a distance histogram is first constructed that records the distribution of pairwise vertical distances between data points on adjacent axes. From this histogram, the median distance value indicates the direction of parallelism, if lines are staying horizontal or going upward or downward. The direction is given by the median

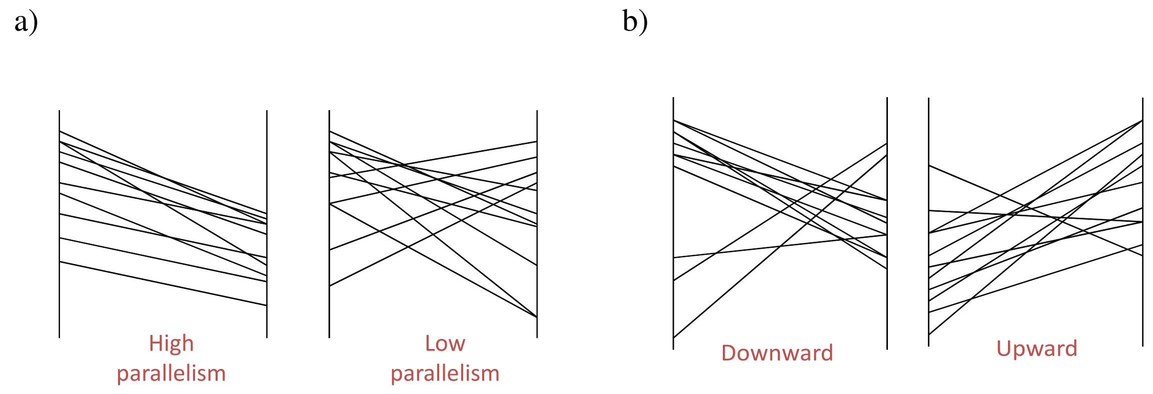


Figure 64: a): Parallelism range indicates how strongly the lines are parallel. b): Principal direction of parallel lines indicate association of high values (on the left axis) with low values (on the right axis) and vice versa.

M_P, which is not normalized (the direction only makes sense in pixel coordinates):

$$M_P = q_{50} \tag{23}$$

where q_{50} represents the 50% quartile of the distance distribution.

The extent of parallelism is given by the interquartile range: a narrow interquartile range implies high parallelism. We normalize the distances between 0 and 1, by dividing by the highest possible distance. We then compute parallelism P_{norm} as follows based on the interquartile range between the 25% and the 75% quartiles, q_{25} and q_{75} .

$$P_{\text{norm}} = 1 - |q_{75} - q_{25}| \tag{24}$$

The subtraction is done to get a higher parallelism value for a higher degree of parallelism (and thus a smaller interquartile range). M_P and P_{norm} are used to draw line-plots for the median and range for all the time steps that models the temporal trends over time. These are illustrated in Section 6.6.2

Visualization: In Figure 64 we illustrate the parallelism metric with respect to the extent of parallelism (Figure 64a) and its direction (Figure 64b). In Figure 64a, the left image denotes a high value of P_norm and the right image denotes lower value. In Figure 64b, the first image denotes a negative value for M_P because of dominance of lines going downward while the second one denotes a positive value because of the dominance of lines going upward. As shown in these figures, we can use the parallelism metric to explore the linear relationship among a subset of values on adjacent dimensions by using semantic brushing.

6.6.2 Matrix View

A problem with parallel coordinates is that ordering of the variables has to be effective enough to convey the different conceivable properties that exist. This becomes an even bigger challenge for temporal data, because it is difficult to track the temporal pattern of all combinations of variables with the default layout. To address this issue, we build matrix layout (Figure 60C) similar to a scatterplot matrix, and show the parallelism range and median plots in that view.

The upper part of the box (Figure 65), which is a filled area, shows spread of the parallel lines denoted by P_{norm}, defined in Section 6.6. Line plot in the bottom one depicts their direction, denoted by M_P. A large area under the curve corresponds to high parallelism (high P_{norm}) , i.e., less spread and smaller area means less parallelism (lower P_{norm}), i.e. more spread.

The lower part of the box shows the line plot for M_P. An indicator horizontal line through the middle of the lower part indicates an M_P value of 0, i.e., lines remaining horizontally parallel to each other between adjacent axes. The line plot going above the indicator line

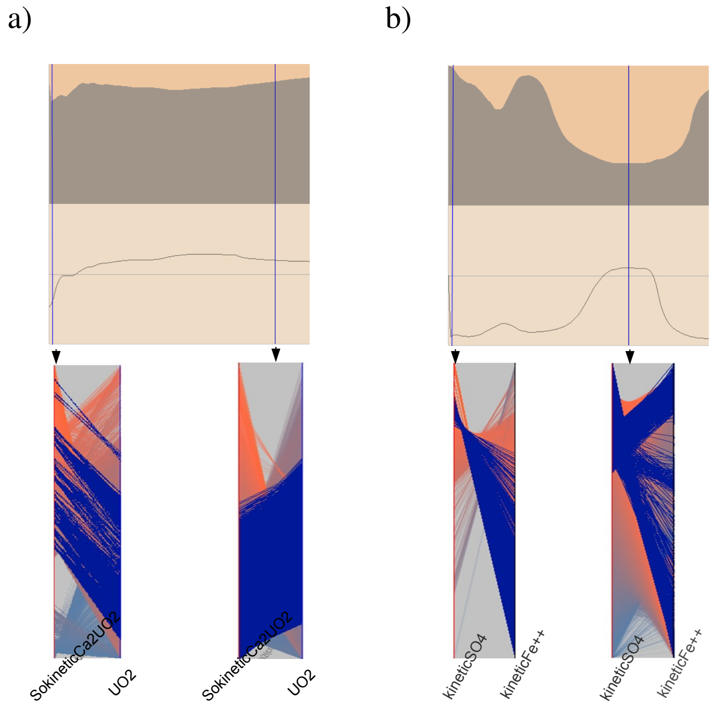


Figure 65: Illustration of the parallelism metric, for two different axis pairs, when the user performs brushing by high parallelism.

denotes most lines in the parallel coordinates plot going upward ($M_P > 0$) and when below the indicator line, that denoted most lines going downward ($M_P < 0$).

In Figure 65a, P_{norm} is consistently high as shown by the unchanged, large area under the curve. In the first selected time step indicated by the arrow-head, M_P is less than zero. This results in the brushed lines being strongly aggregated and going downward. In the second selected time step, the plot for M_P moves upward, over the indicator line. This results in the brushed lines still strongly aggregated but going upward. In Figure 65b, P_{norm} is high initially and M_P is less than zero. So we see strongly aggregated lines going downward. Later on, P_{norm} exhibits lower value but M_P increases later on. Therefore we see lines going in both directions and not as tightly aggregated.

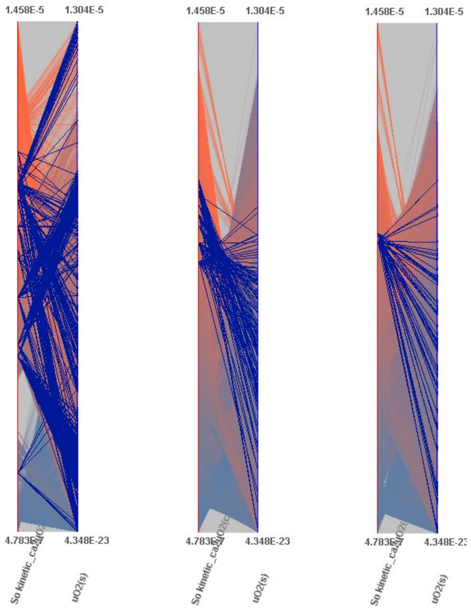


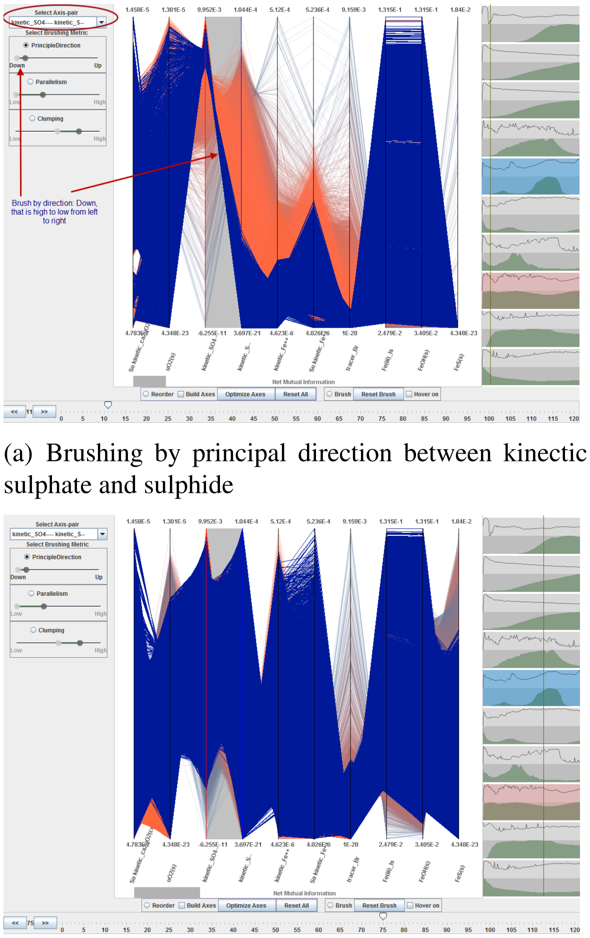
Figure 66: Different degrees of clumping exhibited between uranium carbonate and uraninite at subsequent time steps. The clumping metric (C_f) returns a higher value when there are more clumped regions as in the leftmost image.

6.7 Exploring Subspaces (Q4)

In this section we address subspace analysis (Q4) through rank-ordered views and the clumping metric. Subspaces relate to a) subset of the different dimensions and b) subset of the data-points, exploring both of which are of interest to the analysts. First, we describe how we enable analysts to explore dimensions of interest at different time steps and then we demonstrate our clumping metric. Note that analysts can explore features for subset of data points also by brushing by the parallelism metrics as demonstrated earlier in Section 6.4.2.

6.7.1 Rank-Ordered Axis Pairs

The ranked view orders axis pairs according to ascending or descending values of the metrics. For the sake of clarity on the screen, we restrict the number of axis pairs to be



Brush by direction: Down, that is high to low from left to right

(a) Brushing by principal direction between kinectic sulphate and sulphide

(b) Sulphate concentrations fall and sulphur concentrations rise as the reaction approaches completion

Figure 67: Brushing by principal direction enables an analyst to observe the association between low and high values of adjacent dimensions.

shown, at a time, to five. The ranked view is also time-sensitive and different rankings are produced at different time steps. This view enables an analyst, who might not be familiar with the data in advance, to have a guidance for what the interesting patterns are at particular time steps as demonstrated in Section 6.8.2). Different axes can be selected and a parallel coordinates configuration can be built from the ranked view.

The ability to select axes according to interesting patterns and exploring a subset of the dimensions through the ranking system helps in overcoming scalability issues. In case of datasets with dimensions of the order of hundreds, this method can be used effectively to select a subset of the dimensions to build a parallel coordinates layout. The system also has a capability of building a parallel coordinates visualization by selecting axis pairs in the matrix view.

Visualization: Clumping factor for an axis pair at different time steps are shown in Figure 66. The greater the clumping value, the more is the number of regions with higher number lines converging to or diverging from that region. Brushing by high clumping on an axis pair enables an analyst to visualize the behavior of the clusters across multiple axes and examine the temporal change of those neighborhoods.

6.8 Case Study

We applied our new integrated analysis methods to a bioremediation [50] dataset in order to demonstrate the utility of our system. In performing this case study, we worked closely with a team of subsurface scientists with expertise in subsurface geology, chemistry, and numerical modeling. The specific goals of our collaboration were to identify the utility of our analysis methods to accomplish the following: Task A) identify stable and

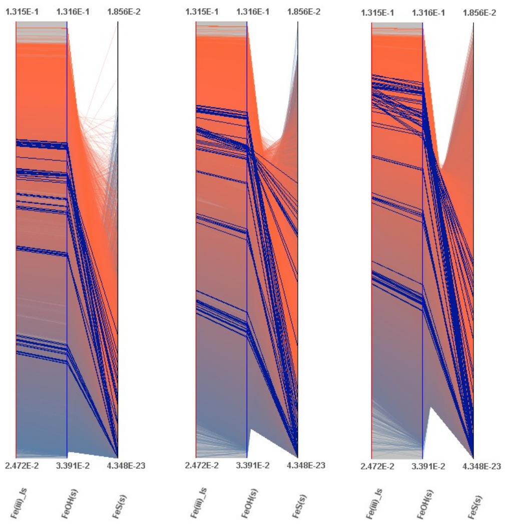
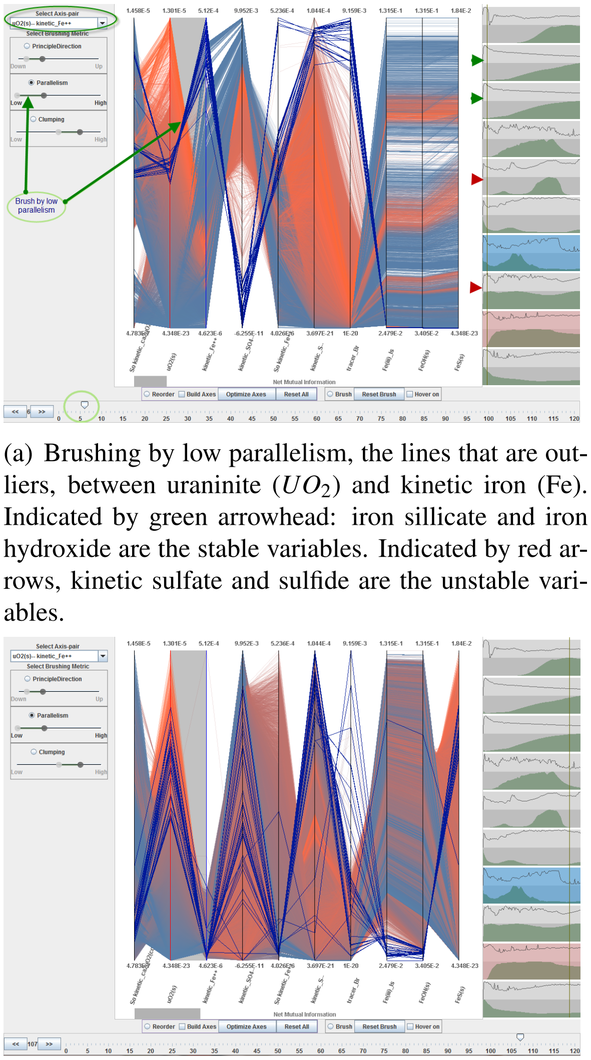


Figure 68: Clumping pattern among Iron sillicate, Iron hydroxide and Iron sulphide that was an unexpected phenomena according to the scientists' initial hypothesis.

unstable trends in variables in order to help verify certain aspects of the process models used to generate the data (Q1); Task B) build visual evidence to help confirm certain hypotheses regarding the interactions between variables (specifically uraninite, sulfates and sulfides chemical species) and discover the trends and anomalies, if any (Q2, Q3); and Task C) identify banding or clumping patterns among the different variables, verify them and observe how these patterns change over time (Q4).

6.8.1 Data Overview

This data originates from a numerical simulation that models a complex, bioremediation field experiment event that occurred over a two-month interim. The data itself consists of 10 different chemical components,i.e., variables (Table 8) that are simulated on a $3D$ grid of size 56 X 40 X 43 (approximately 96000 records). With respect to the temporal aspect of the data, there are approximately 120 time steps where the simulation outputs data for the 10 variables. The complexity of the bioremediation process itself is significant.



(a) Brushing by low parallelism, the lines that are outliers, between uraninite (UO_2) and kinetic iron (Fe). Indicated by green arrowhead: iron sillicate and iron hydroxide are the stable variables. Indicated by red arrows, kinetic sulfate and sulfide are the unstable variables.

(b) Outliers conforming to the trend between uraninite (UO_2) and kinetic iron (Fe++) towards the end of the simulation.

Figure 69: Gaining an overview from the bivariate view and the univariate view allows an analyst to select outliers in the main view. Density view indicates stable and unstable variables.

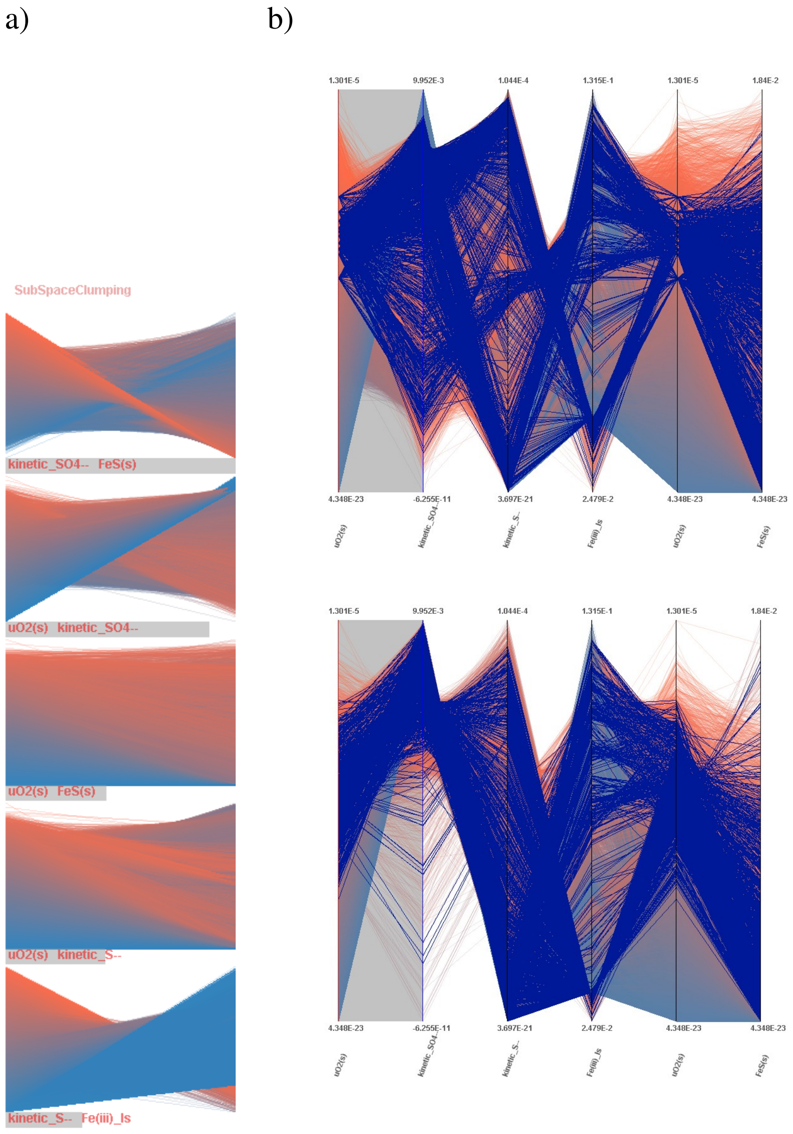


Figure 70: a) Ranked axis pairs according to descending order of clumping. Grey bars indicate the degree of clumping. b) An analyst builds a parallel coordinates configuration from the ranked view and explores the changing behavior of the clumped subspaces between uraninite (UO_2) and kinetic sulfate.

The simulation begins with the injection of acetate into the subsurface. This injection initially stimulates the growth of the bacteria Geobacteraceae, and this bacteria uses the acetate substrate to reduce iron and aqueous uranium. The reduction of uranium produces a solid called uraninite, so that the reduction process effectively removes uranium from the groundwater. As this simulation is made up of over 29 unique reactions that are temporally distinct, we limit our discussion to a few selected motivating examples.

6.8.2 Analysis

We begin analysis by using the density view to provide a high-level overview of univariate data distribution for all variables. Addressing Task A (identifying stable and unstable variables), the density view in Figure 69 shows certain chemical species, like iron silicate, iron hydroxide, and iron sulphide maintain significant stability throughout the bioremediation process as indicated by their unchanged uniform distributions over time. In contrast, other chemical species display significantly more instability, indicating these variables are more involved in the remediation process (e.g., kinetic sulfate and kinetic sulfide).

Next, we inspect the bivariate distributions between iron sillicate and hydroxide (matrix view in Figure 60C); In these distributions, note the strong parallelism between these species that remains more or less unchanged over time. This stable trend in parallelism indicates strong correlation throughout the simulation. This property was specifically of interest to the scientists who further explored the subspace between iron sillicate and hydroxide by brushing according to clumping. As shown in Figure 68, this brushing indicates strong clumping patterns between iron sillicate, hydroxide, and sulphide that remain largely unchanged across the temporal axis. The scientists identified this trait as significant in that stable clumping (in this instance) implies a low level of reactivity between these variables that is indicative of initialization errors in the simulation itself.

To address Task B (build and confirm hypotheses), we look at the density median plots (Figure 69) to get an overview of the interactions between sulfates and sulfides. In this figure, the corresponding boxes show gradually rising median trends for iron sulfide (Fes) and kinetic sulfide, while sulfate concentrations mostly remain high. From this visual information, the scientists hypothesized that cells with initially low sulfur concentrations would exhibit a rise in these concentrations, especially with respect to kinetic sulfate. This hypothesis was confirmed by brushing by principal direction (Figure 67(a)) where a selection of downward trend shows a strong cluster between high concentrations of kinetic sulfate, kinetic sulfide, and iron sulfide. This trend slowly gives way to more random patterns and the rising concentrations of sulfur are reflected with scattered brushed lines (Figure 67(b)). Our collaborators therefore confirmed their hypothesis and also concluded that concentrations of sulfate species become depleted in the middle of the reaction, and begin to rise again towards the end of the reaction.

For examining anomalies, we select the axis pair involving uraninite and kinetic iron. Both density view and parallelism views show strong initial downward parallelism from uraninite to kinetic iron. We examine the behavior of outliers by brushing using low parallelism as the criteria which showed association between high values on both axes (Figure 69(a)). While this was flagged as a potential anomaly, the cells conformed to the expected trend of association between high values of uraninite and low values of iron, towards the end of the simulation, leading the scientists re-affirm their reaction model.

The scientists were able to address Task C (identify banding pattern) through the rank-ordering feature of the application. Ranked configuration for clumping (based on a time step in the middle of the simulation) is shown in Figure 70a. Note that the top parallel coordinates configuration is built from this view. Since the scientists already knew the axis configuration they wanted, optimization by mutual information was not used here. In Figure 70, we can see that the configuration shows strong divergent clumping for uraninite, while lines are spread on the kinetic sulfate axis. Towards the end of the reaction, higher values of uraninite are associated with higher values of sulfate, and this was consistent with the previous finding.

6.8.3 Performance and Scalability

The advantage of suing screen-space metrics is that, after the initial data transformation involving pixel-based binning, the computation of the metrics is independent of the cardinality of the data. For the axis entropy and density median the time complexity is $O(h)$, while for parallelism and clumping factor, the same is $O(h^2)$. Height of the display, that is, h is most likely to be in the hundreds in most cases, while the number of data points can be in the higher thousands or millions. This implies $h << n$, and that $O(h^2)$ will be less than $O(n)$ in most cases. Since data set sizes grow much faster than display sizes, this relationship will only shift towards screen space being more efficient over time.

Our system is able to show the temporal patterns without any lag in real-time. For handling larger number of dimensions, the rank-ordering and ability to build a configuration by getting an overview, will be very useful. Currently our application is based on 10 dimensions. When the number of dimensions is in the hundreds, the rank-ordering feature can be used to reduce the number of dimensions to a manageable set for the parallel coordinates.

CHAPTER 7: DIRECTIONS FOR FUTURE RESEARCH

I have demonstrated in the preceding sections that the visual uncertainty framework not only enables us in building a theoretical foundation for bridging the machine-side operations with human-side implications, but it also gives us practical solutions to omnipresent research problems like high-dimensional data analysis and privacy-preserving data analysis. Consequently, we can extend both the theoretical and practical directions of this work.

7.1 Effectiveness of Screen-Space Metrics

The definition and use of screen-space metrics in visualization is still in its infancy. Several proposals exist like the one suggested by Bertini and Santucci [15] but a comprehensive analysis of their advantages over data-based metrics is missing in the literature. We aim to extend our work on Pargnostics by comparing with data-based metrics. As suggested by Bertino and Santucci, feature preservation metrics are can provide a way for measuring effectiveness of information visualization techniques. Li et. al. [75] provide a way for measuring for comparing scatterplots and parallel coordinates with respect to the features which are discernible in both the visualizations. Given their framework, we want to measure how well a user performs certain tasks using our metrics and using raw data-space metrics based on the recommended steps suggested by Bertini and Santucci. Our hypothesis is that screen-space metrics relate more directly to what a user sees on screen, than data-space metrics; thereby screen-space metrics are expected to be more helpful than data-space ones in identifying the features interactively. Specifically, based on the user studies, we want to address the following areas:

- Compare pattern output from data-space metrics and screen-space metrics. For example parallelism and pearson correlation coefficient are almost identical, through our experiments.

- We want to look at combinations of criteria to see which of those reflect meaningful information about the different trends and relations in the data; and if the users are comfortable using the screen-space measures. The user will not have direct control over the metrics, but those will be automatically invoked based on the tasks he performs.

- We want to confirm the effects of intended and unintended effects of information loss measures described earlier in Chapter 3 on the users' tasks. For example, a high degree of entropy due to large number of crossings may theoretically produce a high amount of information loss, but if the crossings are at the middle, a user will perceive that as inverse correlations. We want to filter out false-positive(information loss is low but user does not finds anything useful due to some other factors which we might miss) and false-negative(information loss is high but user finds something useful) cases like these based on the studies.

7.2 Guiding Design Choices

We demonstrated the utility of the visual uncertainty taxonomy from an analyst's perspective, by illustrating how information content on the screen can be described, classified and quantified, to simplify the path from data to insight. An immediate implication of the taxonomy from the visualization designer's perspective is its potential role in systematization of design choices, both in terms of choice of visual variables and that of interaction mechanisms. The smallest indivisible unit of all visualizations are the visual variables. While there has been some pioneering work involving the type and categorization of visual variables and their design implications [11, 30, 26], we still lack a proper understanding of the perceptual implications of the selection of visual variables for high-dimensional data analysis. While design choices and their motivations exist sporadically in different research papers, there is a dearth of a framework for their evaluation and comparison.

Generalization for different visual variables: Our taxonomy works for the position variable, while some parts can be generalized for other variables. For example, configuration applies all spatial variables like size, shape and orientation. Precision and granularity reflecting the resolution of the representation, can be applied for color. The fist step towards filling this gap is extension of the taxonomy of visual uncertainty and generalizing it for high-dimensional data, for all visual variables. Moreover, perception research [58] has shown that visual variables are not processed independently, but in parallel. Since encoding of high-dimensional data involves multiple visual variables, an interesting problem is how to quantify the additive affect of the uncertainty effected by the different variables. Along the lines of Table 1, the next step would be to study the trade-offs among intended and unintended effects of these different uncertainty components in the context of different tasks for high-dimensional data. These would serve as a feedback for not only the visual mapping process, but also for influencing the choice of visual variables, that is external to the pipeline.

Interaction design: In interactive visualization, different user interaction mechanisms help maximize data fidelity. For example, zooming helps in viewing the data at multiple resolutions and dimension reordering (in case of parallel coordinates) helps in get different perspectives on the multi-dimensional relationships. On the other hand, when there is inherent loss in precision, or there is uncertainty due to traceability, when there are unknown unknowns (the existence of hidden data points), it is difficult to devise interaction techniques to recover such information. Study of interaction techniques and there effectiveness in the realm of visualization has received much less attention. While there has been recent efforts [126] to bridge the machine and human side of interaction, there is still a lack of knowledge on how visual representations and interaction complement each other for analytical tasks. One application of the generalized taxonomy is to study how interaction techniques can be better informed about the causes and effects of uncertainty that can be reduced, thereby leading the development of an effective user-centric visualization optimization model.

7.3 Verification and Validation

On the other hand, a direct corollary of influencing visualization design is the application of the taxonomy for evaluating visualizations. The need for effective verification and validation methods has been widely acknowledged and received a lot of attention in discussion forums related to visualization. Controlled experiments and user studies, although are common techniques for these purposes, there conclusions are often not generalizable and difficult to model. Evaluation metrics beyond measures related to performance related metrics like those related to time and error [13] are needed to compare visual representations, which form the interface between the machine and the user [53]. Especially in applications where there is unavoidable information loss, whether intended or unintended, as in case of privacy-preserving or high-dimensional visualizations respectively, addressing the trade-off between data fidelity and perceptual effectiveness is crucial.

Perceptually beneficial designs do not always result in high data fidelity. Sometimes, certain design choices may motivated by perception, but the results can be counter-intuitive, as shown by a recent study of cluster-based variants of parallel coordinates, which shows that some of them perform worse than the ordinary line-based parallel coordinates [60]. This is mainly due to the fact that the patterns get distorted due to the pre-processing steps and therefore fidelity of the representation is affected. Translating this phenomenon in terms of the taxonomy, although edge-bundling reduces identity and traceability uncertainty by obeying Gestalt principles of proximity and continuity, it introduces pattern complexity, which is the reason for the negative result in the study.

The fact that high data fidelity do not always benefit perception, is easily conceivable in visualization. In case of large data points, we might achieve high data fidelity by mapping all data points on screen, but would be perceptually ineffective owing to clutter. In that case, although encoding uncertainty is minimized, those due to decoding are not addressed. We believe, using uncertainty metrics, these trade-offs can be quantified and the findings will be complementary to user studies, which are generally expensive and time-consuming to conduct. While our metrics are perceptually motivated, we have not yet taken the reverse step by conducting studies to verify that all of them reflect how the user perceives information. Some of the metrics like parallelism, convergence-divergence, overlap clutter etc., intuitively and directly reflect the visual structures. But some others like entropy and mutual information, being at a higher level of abstraction, can benefit from controlled user studies for validation purposes.

7.4 Privacy-preserving Visualization in the Real World

While we have implemented the technique, the rest of the infrastructure is still a proof-of-concept. If the data is present on the user's machine, there are ways for the user to bypass our program and access it. A full implementation of this idea therefore would require a client-server model where the raw data resides on a server, and the visualization client can make requests to display it. The server sanitizes the data before it is sent to the visualization front-end. Since the server is told about the user's display resolution, it can create the appropriate clustering. Client-server configuration can pose an entire new set of challenges, whereby the medium itself can be compromised and be under threat from attackers. A practical implementation needs to take all these issues into consideration.

The focus of this thesis has been to build the privacy-preserving visualization model ground-up: we have identified the different elements of the model using the uncertainty classification and using screen-space metrics to validate privacy and utility. We would like to utilize our metrics in conjunction with the client-server model in a real-world application scenario to detect any unforeseen disclosure scenarios.

Although we have modelled some aspects of the background knowledge that an attacker might possess, there are other aspects of it that can still be incorporated in the analysis, like multi-dimensional background knowledge and other types of knowledge based on attribute types. The fact that background knowledge is subjective and therefore hard to model has been widely acknowledged in the PPDM literature [44]. There has been some recent work related to modeling of background knowledge in the context of PPDM [76] and we would like to integrate our visualization model with the data-based model. We are looking to extend out work by building a Bayesian network model that would depict the dependencies among the different uncertainty-causing factors in the system, taking background knowledge of the attacker into account. Besides privacy, utility of a privacy-preserving application is another equally important factor that needs to be analyzed and quantified. As a next step, we want to perform a detailed analysis of how the different causes and sources of visual uncertainty affect the utility of different privacy-preserving visualization techniques.

CHAPTER 8: CONCLUSION

Quantifying the dynamic visual information content in interactive visualizations has been a widely acknowledged research problem. Because of the subjective nature of the exploratory analysis process, it is difficult to objectively define boundaries between noise and information. To this effect, my thesis provides a novel perspective towards building a framework for design and evaluation of visualization models and techniques, from the point-of-views of both the visual designer and the analyst. We have conceptualized the visualization pipeline as a communication channel for measuring the different characteristics of information as it traverses the pipeline from the machine to the human mind. By systematically defining screen-space metrics and using them as vehicles of this model, we have demonstrated how the problems associated with information loss can be controlled or overcome in complex analysis scenarios involving sensitive data or high-dimensional, time-varying data.

Controlled experiments and user studies are needed to further explore the characteristics of visual uncertainty in different application scenarios, with varying degrees of user expertise. These will help in throwing light into how effects of uncertainty on the perception and cognition side, like identity, traceability, pattern complexity, and lack of knowledge, can be classified further into finer levels of detail. I believe, this research direction will help furthering the seminal work on the structure of the information design space [26] by providing us with fresh insight into improving visual representations. The motivation for this pursuit stems from the demonstrated effectiveness of the visual uncertainty paradigm in building effective analytical abstractions and interaction mechanisms for the different application scenarios previously described in this thesis.

At the core of the visual uncertainty philosophy is the disparity between available pixels and number of data points to be encoded. The rate of growth of data sizes over the past few years has significantly outnumbered that of the growth in the number of available screen pixels [124]. Moreover, experiments have shown that even the availability of gigapixel displays do not guarantee statistically significant performance improvements over standard limited resolution screen sizes. Another development in display technology in recent times, has been the advent of high-resolution retina displays, starting with iPhone and iPad. This trend is expected to continue with other devices as well [121]. But simultaneously, high cardinality and dimensionality of data resulting from simulation experiments, business transactions, internet logs, etc. will continue to make the question of information loss relevant in visualization. Thus, the applicability of the research direction presented in my thesis will not be less, but only increase, in the coming years as data sizes continue to grow at an unprecedented rate.

Optimization of visual representations and controlling different parameters based on screen-space metrics ultimately leads us to the research question: how much automation is necessary and sufficient for complementing exploratory analysis process of discovering unknown unknowns within the data? While there has been some evidence of examining the complementary role of automation and user exploration [12], there is a huge scope for using evidences from future research based on my thesis to provide objective insight into striking a balance between automation and user-control in mixed-initiative visualization systems.

Finally, a significant contribution of this thesis is the treatment of sensitive data in visualization, which has been surprisingly a hitherto uncharted territory in spite of the near ubiquitous presence of privacy concerns in real world collaborative projects. As we move more and more towards open data, some portions of sensitive data have to be protected while they being still available for analysis. Effective access control to prevent unwanted disclosure within organizations and outside will be an increasingly critical challenge. Visualization has a unique way of dealing with data, where summary representations ensure information loss and facilitate effective analysis at the same time. Our conceptualization of privacy-preserving visualization and future research based on it, will have a compelling effect on effecting newer strategies for sharing of private data, whereby the privacy-aware visualization or the analysis is shared, rather than the data. We believe this will reduce the practical bottlenecks and break legality barriers in data sharing, as in this case the data itself is inaccessible as it resides on the server. Moreover, by comparing with data-based approach of privacy, we have demonstrated the effectiveness of the visual approach both in terms of privacy and utility. While interaction with the data leads to potentially higher utility, handling all interaction scenarios can be a privacy concern. The evaluation metrics based on visual uncertainty leading to to optimized representations will help guarantee that sensitive information is not leaked even with use of interaction and background knowledge of the attacker. Therefore, for both the data owner and the data consumer/ analyst, privacy-preserving visualization holds great promise.

In light of these arguments, my thesis not only contributes to the existing visualization literature by bringing a paradigm shift in understanding how visualizations work, but also prepares the ground for tackling practical research problems that are relevant in the present and the future, across various application domains.

REFERENCES

[1] Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D., and Zhu, A. Approximation algorithms for k-anonymity. *Journal of Privacy Technology* (2005).

[2] Agrawal, D., and Aggarwal, C. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the ACM Symposium on Principles of database systems* (2001), ACM, pp. 247–255.

[3] Agrawal, R., and Srikant, R. Privacy-preserving data mining. *ACM Sigmod Record 29*, 2 (2000), 439–450.

[4] Akiba, H., and Ma, K. A tri-space visualization interface for analyzing time-varying multivariate volume data. In *Proceedings of Eurographics/IEEE VGTC Symposium on Visualization* (2007), pp. 115–122.

[5] Allen, J. Maintaining knowledge about temporal intervals. *Communications of the ACM 26* (1983), 832–843.

[6] Amar, R., Eagan, J., and Stasko, J. Low-level components of analytic activity in information visualization. *Proceedings Information Visualization* (2004), 111–117.

[7] Andrienko, G., and Andrienko, N. Constructing parallel coordinates plot for problem solving. In *Proceedings Smart Graphics* (2001), pp. 9–14.

[8] Andrienko, G., and Andrienko, N. Parallel coordinates for exploring properties of subsets. *Proceedings Coordinated and Multiple Views in Exploratory Visualization* (2004), 93–104.

[9] Ankerst, M., Berchtold, S., and Keim, D. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings Information Visualization* (1998), IEEE CS Press, p. 52.

[10] Artero, A. O., de Oliveira, M. C. F., and Levkowitz, H. Uncovering clusters in crowded parallel coordinates visualizations. *Proceedings Information Visualization* (2004), 81–88.

[11] Bertin, J. Semiology of graphics: diagrams, networks, maps.

[12] Bertini, E., and Lalanne, D. Surveying the complementary role of automatic data analysis and visualization in knowledge discovery. In *Proceedings Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration* (2009), ACM Press, pp. 12–20.

[13] Bertini, E., Perer, A., Plaisant, C., and Santucci, G. Beliv'08: Beyond time and errors: novel evaluation methods for information visualization. In *CHI '08 extended abstracts on Human factors in computing systems* (2008), ACM, pp. 3913–3916.

[14] Bertini, E., and Santucci, G. By chance is not enough: Preserving relative density through non uniform sampling. *Information Visualisation, International Conference on* (2004).

[15] Bertini, E., and Santucci, G. Visual quality metrics. In *In Proceedings,BELIV Workshop* (2006), pp. 1–5.

[16] Bertini, E., Tatu, A., and Keim, D. Quality metrics in high-dimensional data visualization: an overview and systematization. *IEEE Tansactions on Visualization and Computer Graphics 17*, 12 (2011), 2203–2212.

[17] Bertino, E., Fovino, I. N., and Provenza, L. P. A Framework for Evaluating Privacy Preserving Data Mining Algorithms*. *Data Mining and Knowledge Discovery 11*, 2 (Aug. 2005), 121–154.

[18] Bertino, E., Lin, D., and Jiang, W. A survey of quantification of privacy preserving data mining algorithms. *Privacy-Preserving Data Mining* (2008), 183–205.

[19] Bezzi, M. An entropy based method for measuring anonymity. In *Third International Conference on Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007.* (2008), IEEE, pp. 28–32.

[20] Blaas, J., Botha, C. P., and Post, F. H. Extensions of parallel coordinates for interactive exploration of large multi-timepoint data sets. *IEEE Transactions on Visualization and Computer Graphics 14* (2008), 1436–43.

[21] Brath, R. Metrics for effective information visualization. *In Proceesings, IEEE Symposium on Information Visualization* (1997).

[22] Brickell, J., and Shmatikov, V. The cost of privacy. *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '08* (2008), 70.

[23] Bu, S., Lakshmanan, L. V., Ng, R. T., and Ramesh, G. Preservation Of Patterns and Input-Output Privacy. *23rd International Conference on Data Engineering* (Apr. 2007), 696–705.

[24] Byun, J., Kamra, A., Bertino, E., and Li, N. Efficient k-anonymization using clustering techniques. In *Proceedings Database Systems for Advanced Applications* (2007), Springer, pp. 188–200.

[25] Caat, M., Maurits, N., and Roerdink, J. Design and evaluation of tiled parallel coordinate visualization of multichannel eeg data. *IEEE Transactions on Visualization and Computer Graphics 13*, 1 (2007), 70–79.

[26] Card, S., and Mackinlay, J. The structure of the information visualization design space. In *Information Visualization, 1997. Proceedings., IEEE Symposium on* (1997), IEEE, pp. 92–99.

[27] Chen, M., Ebert, D., Hagen, H., Laramee, R., Van Liere, R., Ma, K., Ribarsky, W., Scheuermann, G., and Silver, D. Data, Information, and Knowledge in Visualization. *IEEE Computer Graphics and Applications 29*, 1 (2009), 12–19.

[28] Chen, M., and Jänicke, H. An information-theoretic framework for visualization. *Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1206–1215.

[29] Chi, E. H. A taxonomy of visualization techniques using the data state reference model. In *Proceedings Information Visualization* (2000), IEEE CS Press, pp. 69–75.

[30] Cleveland, W., and McGill, R. Graphical perception: The visual decoding of quantitative information on graphical displays of data. *Journal of the Royal Statistical Society. 150*, 3 (1987), 192–229.

[31] Clifton, C., Kantarcioglu, M., and Vaidya, J. Defining privacy for data mining. In *National Science Foundation Workshop on Next Generation Data Mining* (2002), pp. 126–133.

[32] Cover, T., and Thomas, J. Elements of information theory. *Wiley* (2006).

[33] Cui, Q., Ward, M. O., Rundensteiner, E. a., and Yang, J. Measuring data abstraction quality in multiresolution visualizations. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 709–16.

[34] Dasgupta, A., Chen, M., and Kosara, R. Conceptualizing visual uncertainty in parallel coordinates. *Computer Graphics Forum 31*, 3pt2 (2012), 1015–1024.

[35] Dasgupta, A., Chen, M., and Kosara, R. Measuring privacy and utility in privacy-preserving visualization. *Computer Graphics Forum* (2012 (accepted with minor revisions)).

[36] Dasgupta, A., and Kosara, R. The need for information loss metrics in visualization. *Workshop on The Role of Theory in Visualization, Visweek* (2010).

[37] Dasgupta, A., and Kosara, R. Pargnostics: Screen-space metrics for parallel coordinates. *Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1017–26.

[38] Dasgupta, A., and Kosara, R. Adaptive privacy-preservation using parallel coordinates. *Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2241–2248.

[39] Dasgupta, A., and Kosara, R. Privacy-preserving data visualization using parallel coordinates. In *Proceedings Visualization and Data Analysis (VDA)* (2011), pp. 78680O–1–78680O–12.

[40] Dasgupta, A., and Kosara, R. The importance of tracing data through the visualization pipeline. *Beyond Time and Errors Novel Evaluation Methods for Visualization (BELIV)* (2012).

[41] Dasgupta, A., Kosara, R., and Luke, G. Screen-space metrics for exploratory analysis using temporal parallel coordinates. *Computer Graphics Forum* (2012 (accepted with minor revisions)).

[42] Dasgupta, A., Kosara, R., and Luke, G. Meta parallel coordinates for visualizing features in large, high-dimensional, time-varying data. *Large Data Analysis and Visualization Symposium* (2012(to appear)).

[43] Doleisch, H. Simvis: Interactive visual analysis of large and time-dependent 3d simulation data. In *Proceedings of the Winter Simulation Conference* (2007), IEEE Press, pp. 712–720.

[44] Du, W., Teng, Z., and Zhu, Z. Privacy-maxent: Integrating background knowledge in privacy quantification. In *Proceedings of the SIGMOD international conference on Management of data* (2008), ACM, pp. 459–472.

[45] Duncan, G. T., and Lambert, D. Disclosure-limited data dissemination. *Journal of the American Statistical Assn. 81*, 393 (1986), pp. 10–18.

[46] Dwork, C. Differential privacy. In *ICALP* (2006), Springer, pp. 1–12.

[47] Ebrahimi, N., Maasoumi, E., and Soofi, E. Ordering univariate distributions by entropy and variance. *Journal of Econometrics 90*, 2 (1999), 317–336.

[48] Ellis, G., Bertini, E., and Dix, A. The sampling lens: making sense of saturated visualisations. In *CHI'05 Extended Abstracts on Human factors in Computing Systems* (2005), pp. 1351–1354.

[49] Ellis, G., and Dix, A. Enabling automatic clutter reduction in parallel coordinate plots. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 717–724.

[50] Fang, Y., Yabusaki, S. B., Morrison, S. J., Amonette, J. P., and Long, P. E. Multicomponent reactive transport modeling of uranium bioremediation field experiments. *Geochimica et Cosmochimica Acta 73*, 20 (2009), 6029 – 6051.

[51] Forsell, C., and Johansson, J. Task-based evaluation of multirelational 3D and standard 2D parallel coordinates. *Proceedings of SPIE* (2007), 64950C–64950C–12.

[52] Frank, A., and Asuncion, A. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2010.

[53] Freitas, C., Luzzardi, P., Cava, R., Winckler, M., Pimenta, M., and Nedel, L. On evaluating information visualization techniques. In *Proceedings of the working conference on Advanced Visual Interfaces* (2002), ACM, pp. 373–374.

[54] Fua, Y.-H., Ward, M. O., and Rundensteiner, E. A. Hierarchical parallel coordinates for exploration of large datasets. In *IEEE Visualization* (1999), IEEE CS Press, pp. 43–50.

[55] Geng, Z., Peng, Z., S Laramee, R., C Roberts, J., and Walker, R. Angular histograms: Frequency-based visualizations for large, high dimensional data. *IEEE Transactions on Visualization and Computer Graphics, 17*, 12 (2011), 2572–2580.

[56] Glatter, M., Huang, J., Ahern, S., Daniel, J., and Lu, A. Visualizing temporal patterns in large multivariate data using textual pattern matching. *IEEE transactions on visualization and computer graphics 14*, 6 (2008), 1467–74.

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

[58] Green, M. Towards a perpectual science of multidimensional data visualization: Bertin and beoyond. http://www.ergogero.com/dataviz/dviz0.html, 1996.

[59] Guo, D., Chen, J., MacEachren, A. M., and Liao, K. A visualization system for space-time and multivariate patterns (VIS-STAMP). *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2005), 1461–74.

[60] Holten, D., and Van Wijk, J. Evaluation of cluster identification performance for different pcp variants. *Computer Graphics Forum 29*, 3 (2010), 793–802.

[61] Holzhüter, C., Lex, A., Schmalstieg, D., Schulz, H.-J., Schumann, H., and Streit, M. Visualizing uncertainty in biological expression data. In *Proceedings Visualization and Data Analysis* (2012).

[62] Huang, W., Hong, S.-H., and Eades, P. Effects of Crossing Angles. In *Proceedings Pacific Visualization Symposium* (2008), pp. 41–46.

[63] Hubbard, D. *How to measure anything: Finding the value of intangibles in business*. Wiley, 2010.

[64] Inselberg, A. Multidimensional detective. In *Proceedings Visualization* (1997), IEEE CS Press, pp. 100–107.

[65] Inselberg, A. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer, 2009.

[66] Inselberg, A., and Dimsdale, B. Parallel coordinates: A tool for visualizing multidimensional geometry. In *IEEE Visualization* (1990), IEEE CS Press, pp. 361–378.

[67] Johansson, J., and Cooper, M. A screen space quality method for data abstraction. *Comput. Graph. Forum 27*, 3 (2008), 1039–1046.

[68] Johansson, J., Ljung, P., and Cooper, M. Depth cues and density in temporal parallel coordinates. In *EuroVis* (2007), vol. 7, pp. 35–42.

[69] Johnson, C., and Sanderson, A. A next step: Visualizing errors and uncertainty. *Computer Graphics and Applications 23*, 5 (2003), 6–10.

[70] K. El Emam, F. D. Protecting privacy using k-anonymity. *Journal of the American Medical Informatics Association 15* (2008), 627–637.

[71] Keim, D. Designing pixel-oriented visualization techniques: theory and applications. *IEEE Transactions on Visualization and Computer Graphics 6* (2000), 59–78.

[72] Klir, G., and Wierman, M. *Uncertainty-based information: Elements of generalized information theory*. Springer Verlag, 1999.

[73] Lambert, D. Measures of disclosure risk and harm. *Journal of Official Statistics 9* (1993), 313–331.

[74] Lee, T.-Y., and Shen, H.-W. Visualization and exploration of temporal trend relationships in multivariate time-varying data. *IEEE Transactions on Visualization and Computer Graphics 15* (2009), 1359–1366.

[75] Li, J., Martens, J.-B., and van Wijk, J. J. Judging correlation from scatterplots and parallel coordinate plots. *Information Visualization 9*, 1 (2010), 13–30.

[76] Li, T., Li, N., and Zhang, J. Modeling and integrating background knowledge in data anonymization. In *In Proceedings, International Conference on Data Engineering* (2009), IEEE, pp. 6–17.

[77] Lind, M., Johansson, J., and Cooper, M. Many-to-Many Relational Parallel Coordinates Displays. *2009 13th International Conference Information Visualisation* (July 2009), 25–31.

[78] Lodha, S., Pang, A., Sheehan, R., and Wittenbrink, C. Uflow: Visualizing uncertainty in fluid flow. In *Proceedings Visualization* (1996), pp. 249–254.

[79] Loeb, D. A generalization of the stirling numbers. *Discrete mathematics 103*, 3 (1992), 259–269.

[80] Lopes, A., and Brodlie, K. Accuracy in 3d particle tracing. *Mathematical Visualization: Algorithms, Applications and Numerics* (1998), 329–341.

[81] Luzzardi, P., Freitas, C., Cava, R., Duarte, G., and Vasconcelos, M. An Extended Set of Ergonomic Criteria for Information Visualization Techniques. In *Proceedings of the Seventh IASTED International Conference on Computer Graphics And Imaging (CGIM 2004)* (2004), pp. 236–241.

[82] Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD) 1*, 1 (2007), 3.

[83] Meyer, M., Munzner, T., DePace, A., and Pfister, H. Multeesum: A tool for comparative spatial and temporal gene expression data. *, IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 908–917.

[84] Meyerson, A., and Williams, R. On the complexity of optimal k-anonymity. In *Proceedings Principles of Database Systems* (2004), ACM, pp. 223–228.

[85] Miller, N., Hetzler, B., Nakamura, G., and Whitney, P. The need for metrics in visual information analysis. In *NPIV '97: Proceedings of the 1997 workshop on New paradigms in information visualization and manipulation* (1997), ACM, pp. 24–28.

[86] Milliken, F. Three types of perceived uncertainty about the environment: State, effect, and response uncertainty. *Academy of Management review* (1987), 133–143.

[87] Muller, W., and Schumann, H. Visualization methods for time-dependent data - an overview. In *Proceedings of the Winter Simulation Conference,* (2003), vol. 1, pp. 737 – 745 Vol.1.

[88] Munzner, T. A nested process model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 921–928.

[89] 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.

[90] P. O'Dea, J. G., and O'Riordan, C. Combining feature selection and neural networkds for solving classification problems. *Proceedings of 12th Irish Conference of Artifical Intelligence & Cognitive Science* (2001), 157–166.

[91] Pang, A., Wittenbrink, C., and Lodha, S. Approaches to uncertainty visualization. *The Visual Computer 13*, 8 (1997), 370–390.

[92] Peng, W., Ward, M., and Rundensteiner, E. Clutter reduction in multi-dimensional data visualization using dimension reordering. In *Proceedings Information Visualization* (2004), IEEE CS Press, pp. 89–96.

[93] Piringer, H., Kosara, R., and Hauser, H. Interactive focus+context visualization with linked 2d/3d scatterplots. In *2nd International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV)* (2004), pp. 49–60.

[94] Purchase, H., Andrienko, N., Jankun-Kelly, T., and Ward, M. Theoretical foundations of information visualization. In *Information Visualization: Human-Centered Issues and Perspectives*. Springer, 2008, pp. 46–64.

[95] Rhodes, P., Laramee, R., Bergeron, R., and Sparr, T. Uncertainty visualization methods in isosurface rendering. In *Eurographics* (2003), pp. 83–88.

[96] Rit, J.-F. Propagating temporal constraints for scheduling. In *Proceedings of the Fifth National Conference on Artificial Intelligence* (1986), pp. 383–388.

[97] Rundensteiner, E. A., Ward, M. O., Xie, Z., Cui, Q., Wad, C. V., Yang, D., and Huang, S. Xmdvtool: Quality-aware interactive data exploration. In *SIGMOD Conference* (2007), pp. 1109–1112.

[98] Russell, S., Norvig, P., Canny, J., Malik, J., and Edwards, D. *Artificial intelligence: a modern approach.* Prentice hall, 1995.

[99] Schneidewind, J., Sips, M., and Keim, D. A. Pixnostics: Towards measuring the value of visualization. In *Proceedings Visual Analytics Science and Technology* (2006), IEEE CS Press, pp. 199–206.

[100] Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal 27* (1948), 379–423.

[101] Sharp Jr, H. Cardinality of finite topologies. *Journal of Combinatorial Theory 5*, 1 (1968), 82–86.

[102] Shneiderman, B. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings Visual Languages* (1996), IEEE CS Press, pp. 336–343.

[103] Sips, M., Neubert, B., Lewis, J., and Hanrahan, P. Selecting good views of high-dimensional data using class consistency. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 831–838.

[104] Skeels, M., Lee, B., Smith, G., and Robertson, G. Revealing uncertainty for information visualization. *Information Visualization 9*, 1 (2009), 70–81.

[105] Steed, C. a., Swan, J. E., Jankun-Kelly, T., and Fitzpatrick, P. J. Guided analysis of hurricane trends using statistical processes integrated with interactive parallel coordinates. *IEEE Symposium on VAST* (2009), 19–26.

[106] Sweeney, L. k-anonymity: A model for protecting privacy. *IEEE Security And Privacy 10*, 5 (2002), 1–14.

[107] Tatu, A., Albuquerque, G., Eisemann, M., Theisel, H., Magnor, M., and Keim, D. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. *IEEE Symposium on Visual Analytics Science and Technology* (2009), 59–66.

[108] Thomas, J., and Cook, K. A visual analytics agenda. *Computer Graphics and Applications, IEEE 26*, 1 (2006), 10–13.

[109] Thomson, J., Hetzler, E., MacEachren, A., Gahegan, M., and Pavel, M. A typology for visualizing uncertainty. In *Proceedings SPIE* (2005), vol. 5669, pp. 146–157.

[110] Tory, M., and Möller, T. Rethinking visualization: A high-level taxonomy. *Symposium on Information Visualization* (2004), 151–158.

[111] Tufte, E. R. *The Visual Display of Quantitative Information*, 2nd ed. Graphics Press, 2001.

[112] Tukey, J., and Tukey, P. Computing graphics and exploratory data analysis: An introduction. In Proceedings of the Sixth Annual Conference and Exposition: Computer Graphics 85. In *Proceedings of the Sixth Annual Conference and Exposition: Computer Graphics* (1985), pp. 773–785.

[113] Turkay, C., Filzmoser, P., and Hauser, H. Brushing dimensions; a dual visual analysis model for high-dimensional data. *Visualization and Computer Graphics, IEEE Transactions on 17*, 12 (2011), 2591–2599.

[114] V. Ciriani, S. De Capitani di Vimercati, S. F., and Samarati, P. k-anonymous data mining: A survey. In *Privacy-Preserving Data Mining: Models and Algorithms*. Springer-Verlag, 2007, pp. 105–136.

[115] Van Wijk, J. The value of visualization. In *IEEE Visualization* (2005), pp. 79–86.

[116] Wang, C., Yu, H., and Ma, K.-L. Importance-driven time-varying data visualization. *IEEE transactions on visualization and computer graphics 14*, 6 (2008), 1547–54.

[117] Ware, C., Purchase, H., Colpoys, L., and McGill, M. Cognitive measurements of graph aesthetics. *Information Visualization 1*, 2 (2002), 103–110.

[118] Wegman, E. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association 85* (1990), 664–675.

[119] Wegman, E., and Luo, Q. High dimensional clustering using parallel coordinates and the grand tour. *Computing Science and Statistics 28* (1997), 361–368.

[120] Wilkinson, L., Anand, A., and Grossman, R. Graph-theoretic scagnostics. In *Proceedings Information Visualization* (2005), IEEE CS Press, pp. 157–164.

[121] Wilson, T. How the iphone works. *How Stuff Works. http://electronics. howstuffworks. com/iphone. htm. Retrieved June 6* (2008).

[122] Wittenbrink, C., Pang, A., and Lodha, S. Glyphs for visualizing uncertainty in vector fields. *Transactions on Visualization and Computer Graphics 2*, 3 (1996), 266–279.

[123] Yang, J., Ward, M., Rundensteiner, E., and Huang, S. Visual hierarchical dimension reduction for exploration of high dimensional datasets. In *Proceedings Data Visualization* (2003), Eurographics Press, pp. 19–28.

[124] Yang, S., Chung, H., and North, C. and, F. E. The effect of presenting long documents on a large high-resolution display on comprehension of content and user experience. In *International Symposium on Electronic Theses and Dissertations* (2010).

[125] Yang-Peláez, J., and Flowers, W. C. Information content measures of visual displays. In *Proceedings of the IEEE Symposium on Information Vizualization 2000* (2000), IEEE Computer Society, pp. 99–103.

[126] Yi, J., ah Kang, Y., Stasko, J., and Jacko, J. Toward a deeper understanding of the role of interaction in information visualization. *Visualization and Computer Graphics, IEEE Transactions on 13*, 6 (2007), 1224–1231.

[127] Zhang, X., Cheung, W. K., and Li, C. H. Graph-based abstraction for privacy preserving manifold visualization. In *Proceedings of the Conference on Web Intelligence and Intelligent Agent Technology* (Washington, DC, USA, 2006), IEEE Computer Society, pp. 94–97.

[128] Zhou, H., Cui, W., Qu, H., Wu, Y., Yuan, X., and Zhuo, W. Splatting the Lines in Parallel Coordinates. *Computer Graphics Forum 28*, 3 (2009), 759–766.

[129] Zhou, H., Yuan, X., Qu, H., Cui, W., and Chen, B. Visual clustering in parallel coordinates. *Computer Graphics Forum 27*, 3 (2008), 1047–1054.

[130] Ziemkiewicz, C., and Kosara, R. Embedding Information Visualization Within Visual Representation. *Advances in Information and Intelligent Systems* (2010), 307–326.