Radial Diagrams for the Visual Analysis of Wind Energy Production Data

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Visual Computing

eingereicht von

Wolfgang Ludwig
Matrikelnummer 0526080

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Mitwirkung: Dipl.-Ing. Dr.techn. Harald Piringer

Wien, 10.10.2018

(Unterschrift Verfasser) (Unterschrift Betreuung)

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Radial Diagrams for the Visual Analysis of Wind Energy Production Data

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Visual Computing

by

Wolfgang Ludwig

Registration Number 0526080

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Assistance: Dipl.-Ing. Dr.techn. Harald Piringer

Vienna, 10.10.2018

(Signature of Author) (Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Wolfgang Ludwig
Hardeggasse 55/6/28, 1220 Wien

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

(Ort, Datum) (Unterschrift Verfasser)

Acknowledgments

This thesis was written at the VRVis Research Center[1] in Vienna, Austria. I want to thank the VRVis Research Center for providing me a nice work place and for being open to student projects. Great thanks go to Harald Piringer and his team at the research center for managing and supporting this work. Many thanks to Eduard Gröller and the Vienna University of Technology for giving me the chance to work on such an interesting project. By teaching me in the field of Visualization, Eduard Gröller provided me with solid knowledge which I was able to apply throughout this thesis. Harald's and Eduard's ideas for the visualization and interaction design for the analysis of wind energy production data were important contributions to this thesis.

Additional thanks go to Thomas Mühlbacher and Christian Möllinger for all their support regarding the use of the software framework. They always provided me with helpful answers to questions regarding software development as well as user interface design.

[1]VRVis is funded by BMVIT, BMDW, Styria, SFG and Vienna Business Agency in the scope of COMET - Competence Centers for Excellent Technologies (854174) which is managed by FFG.

Abstract

Wind energy production is a fast growing sector in the field of renewable energy production. In the process of energy production, more and more data is produced and recorded every year. This data is usually worthless without further exploration, analysis, and presentation. This thesis presents a design study of the visual analysis of wind energy production data. The goal is to provide data analysts with tools to carry out common tasks in the field of wind energy production more efficiently. As the data commonly contains directional information of winds and gusts, analysis techniques need to take the circular nature of such data into account.

This work proposes a set of techniques for the visualization and interaction with circular data in radial diagrams. The diagrams operate in the polar coordinate system and thus are well suited to solve the problems of maintaining the natural coherence and circular closure of circular data. The thesis discusses important design decisions and gives practical guidance how to implement novel features into an existing software system. Implementation details on how to ensure large data scalability are presented. The work evaluates the results in a case study with real data carried out by an expert in the field of wind energy production. The results indicate an improved work flow of common tasks and a successful system integration. The reported deployment at a national power grid operator further demonstrates the system's user acceptance and importance. The thesis also reflects on the iterative design process and the within collected expert feedback.

Kurzfassung

Die Produktion von elektrischer Energie aus Windenergie ist ein rasch wachsender Sektor auf dem Gebiet der erneuerbaren Energien. Im Prozess der Energiegewinnung werden jedes Jahr immer größere Datenmengen aufgezeichnet. Die große Menge an gesammelten Daten ist jedoch wertlos ohne weitere Untersuchung, Analyse und Präsentation. Ziel ist es, Datenanalysten Werkzeuge zur Verfügung zu stellen, mit denen sie übliche Aufgaben im Bereich der Windenergieerzeugung effizienter durchführen können. Da die Daten häufig Richtungsinformationen von Winden und Böen enthalten, müssen die Analysetechniken die zirkuläre Natur solcher Daten berücksichtigen.

Die Arbeit beschreibt eine Reihe von Visualisierungs- und Interaktionstechniken für zirkuläre Daten auf Basis von radialen Diagrammen. Die Diagramme arbeiten im Polarkoordinatensystem und sind daher gut geeignet, um die Probleme von zirkulären Daten - das Beibehalten natürlicher Zusammenhänge und des zirkulären Schlusses - zu lösen. Die Arbeit diskutiert wichtige Designentscheidungen und gibt praktische Hinweise, wie neue Funktionen in ein bestehendes Softwaresystem implementiert werden können. Implementierungsdetails, welche die Skalierbarkeit bei großen Datenmengen gewährleisten, werden vorgestellt. Die Arbeit evaluiert das Design und die Implementierung anhand einer Fallstudie mit realen Daten. Die Fallstudie wird von einem Experten auf dem Gebiet der Windenergieproduktion durchgeführt. Die Ergebnisse weisen auf einen verbesserten Arbeitsablauf bei üblichen Aufgaben und auf eine erfolgreiche Systemintegration hin. Der berichtete Einsatz der vorgestellten Lösung bei einem nationalen Stromnetzbetreiber verdeutlicht die Akzeptanz und Bedeutung des Systems für die Benutzer. Die Arbeit reflektiert außerdem über den iterativen Entwurfsprozess und die darin gesammelten Expertenrückmeldungen und Erkenntnisse.

Contents

CHAPTER 1

Introduction

This chapter serves to give a brief introduction into the process of wind energy production and will demonstrate the potential and growth trend of the wind energy sector. Within this chapter, the challenges in wind energy production will be discussed. As it will be explained that those challenges can be addressed by the extensive collection and analysis of huge amounts of data, this will give the motivation to use techniques that emerged from the research fields of visualization and visual analytics to support experts in the wind energy sector.

1.1 Wind Energy Production

Wind energy production is the process of converting kinetic wind energy into electrical energy. Kinetic energy is provided by air in motion as air has mass and thus moving air contains the energy of that motion. In fact, this wind energy is a converted form of solar energy originating from the sun's radiation. Wind is generated when the sun's radiation unevenly heats up the world's air (directly) or surfaces (indirectly). This uneven heating comes from different angles of the rays hitting the earth and day and night cycles. In addition, surfaces reflect, absorb, and yield this radiation at different rates. All this causes regions of the atmosphere to heat up differently. As hot air rises into higher layers of the atmosphere, the air pressure in this region drops and cooler air from the surrounding regions is drawn into this low pressure region. The resulting global and local phenomena of air motion driven by temperature differences is called wind.

As wind is not used more quickly than it is produced and continues to regenerate, it is a viable source of energy. Wind energy is known to be one of the most cost effective types of renewable energy. Most importantly, the process itself does not create any pollution, greenhouse gases, or waste [34]. Indirect greenhouse-gas emissions from wind power are among the lowest of any renewable-energy technology [47].

Wind power can be categorized into three major branches of application: *Offshore wind power* uses large wind turbines constructed on platforms in (shallow) waters around the world. *"Small" wind power* uses turbines with usually less than one-hundred kilowatts in capacity to directly power homes or small businesses. *Utility-scale wind power* uses onshore turbines to contribute to national and global power grids [5].

Many countries and companies incorporate wind power into their short and long term energy strategies to reduce their carbon dioxide footprint. In addition, renewable energies strengthen the independence from fossil fuels. These strategies involve the process of building new *wind turbines* and *wind farms*. These constructions are discussed briefly below.

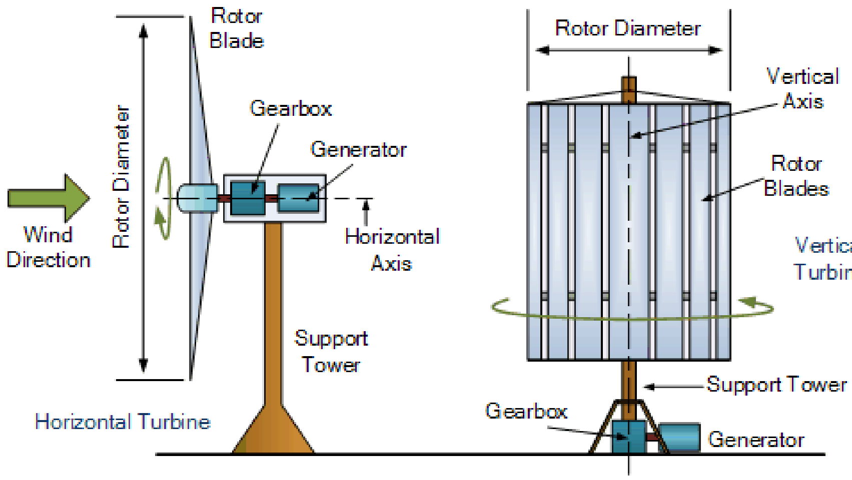


Figure 1.1: A schematic illustration of the two types of wind energy turbines and their most important components. A *Horizontal Axis Wind Turbine (HAWT)* design is shown on the left, a *Vertical Axis Wind Turbine (VAWT)* design on the right side. Image courtesy of Alternative Energy Tutorials [98].

Wind Turbines

A wind turbine is a device that turns the kinetic energy provided by wind into mechanical energy and finally, through a generator, into electrical energy. This electrical energy can then be used for general purposes.

Modern wind turbines are the result of advances in engineering and development of windmills which started about two millennia ago in Persia. Those windmills were used mainly for pumping water or for grinding. However, the first wind turbine known that was used to create electricity was installed in 1887 by James Blyth to power electric lights in his home in Scotland [10]. Over the years, different designs of wind turbines emerged all of which can be classified into the two fundamental types of wind turbines: turbines that rotate around a horizontal axis and turbines that rotate around a vertical axis. Figure 1.1 shows both types and the arrangements of common important components such as the tower, the gearbox, the generator, and the rotor.

The horizontal axis wind turbines (HAWTs) are the most efficient type of wind energy conversion device and thus the most popular one [34]. Today's industrial horizontal axis turbines usually range from 60 to 90 meters in height with blade lengths ranging from 20 to 40 meters. The total power output under ideal conditions commonly ranges from 1.5 to 5 megawatts. The record for the tallest wind turbine is held by the commercial turbine *Vestas V164* installed in 2013 in Denmark. The turbine went into operation in 2014. This offshore wind turbine is 220 meters in height and also holds the world record for being the most powerful one, i.e., with a capacity of 8 megawatts [76].

Wind Farms

(a) A photo of an onshore wind farm. The wind turbines' towers require only a minimal amount of space thus the area below can still be used for agriculture. Image source Pixabay Public Domain Images [43]

(b) A photo of an offshore wind farm. Image adapted from Pixabay Public Domain Images [44]

Figure 1.2: The figure shows photos of typical horizontal axis onshore (a) and offshore (b) wind farms.

The term *wind farm* or *wind park* defines a set of wind turbines in the same geographic location. A number of turbines are locally connected to form a grid of turbines, which provides electricity to the power grid. The power grid then acts as the larger "network" that transports electricity to where it is actually needed. This transport of electricity is called *transmission*. *Transmission lines*, distributed around the globe, move electricity from wind farms to end-users [5].

Wind farms are sometimes also called *wind power plants* as they are used as a unit to be compared against other power plants or sites. For example, a wind farm may consist of one hundred wind turbines with a capacity of two megawatts each, resulting in a total capacity of two-hundred megawatts. Wind farms may be constructed onshore or offshore depending on local wind conditions and construction site options. Figure 1.2 shows typical horizontal axis wind turbine parks on- and offshore.

Although offshore wind parks allow for much bigger wind turbines and offer a greater energy potential, the harsh marine conditions make erecting new turbines and their maintenance more difficult. Installations are more expensive and grid connections need to be developed subsea, i.e., fully submerged. Offshore projects are dominated by northern European countries, followed by China. With 1.9 % (2012) of the market share, offshore projects represent only a small portion of the world's total installed wind capacity [47].

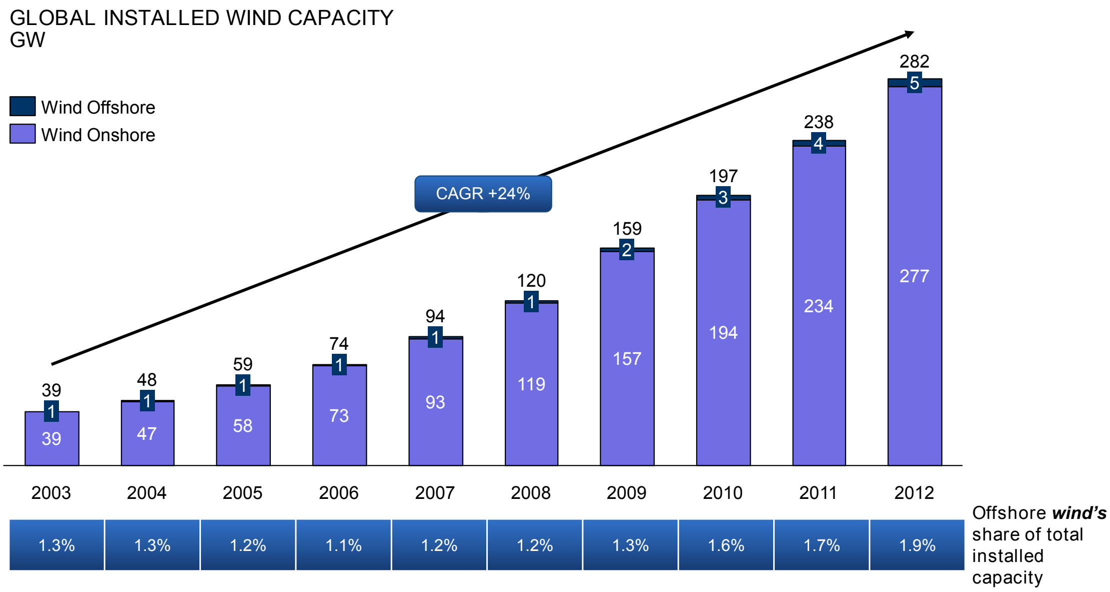


Figure 1.3: The diagram shows the trend of the global installed wind capacity of onshore and offshore projects in the period from 2003 to 2012. The installed capacities have grown at an annual rate of 24 % driven by onshore technology. Capacities are given in gigawatt (GW). Image courtesy of SBC Energy Institute [47].

Potential and Growth Trend

According to the American Wind Energy Association [5], a current estimate of the wind energy potential in the United States shows that it would be possible to produce ten times the annual power consumption of the entire country. They further state that wind power has increased exponentially and the amount of wind energy capacity in the United States has increased by a factor of sixteen between the years 2000 and 2012. In fact, the global wind-power capacity has increased by an average of 24 % per year between 2002 and 2012, as shown in Figure 1.3. This ongoing growth trend is expected to continue as demonstrated by Figure 1.4. In 2017 wind power is estimated to contribute a total capacity of 500 gigawatts to the global energy market.

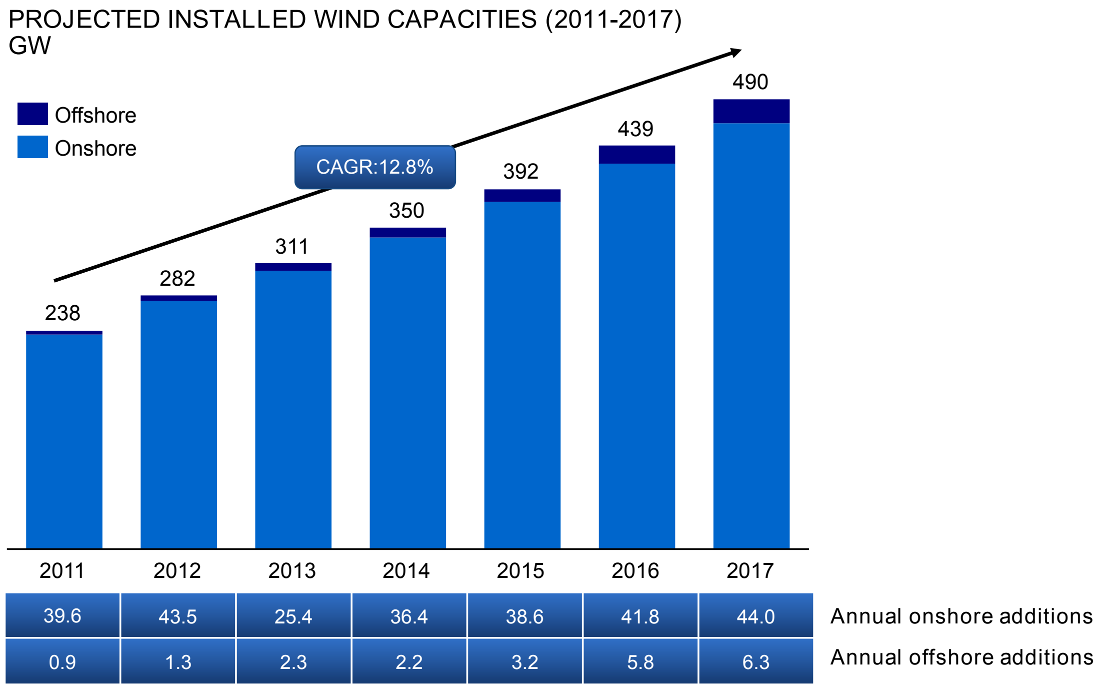


Figure 1.4: The diagram shows the projected total installed wind capacity of onshore and offshore projects for the period 2011 to 2017. The prognosis results in a compound annual growth rate (CAGR) for this period of 12.8 %. Capacities are given in gigawatt (GW). Image courtesy of SBC Energy Institute [47].

The overall global theoretical potential of wind exceeds the current global electricity production [47]. Extensive work on the wind potential in various countries has been published over the last decade. Additionally, work has been conducted to reveal the potential of wind in rural areas by using data extrapolation with the help of computer models [22, 33].

Thanks to this potential and growth trend, wind power will play an important role in the power mix of the future in limiting the effect of global warming. Although wind power poses great potential, it comes with certain challenges. These are discussed below.

Challenges

New but also existing wind power projects face various kinds of economical and environmental challenges. These challenges include:

The Placement of New Wind Farms and Turbines. As wind is a strongly local physical phenomenon, the location of a wind turbine is a crucial decision contributing significantly to the turbine's power output characteristics. When constructing or extending wind farms, the spacing between turbines has to be optimized so that turbulence and slipstream effects are minimized. However, not only the wind potential has to be considered when picking new installation sites. The access to land and transmission lines, including questions of ownership and transport roads, also play an important role when appointing new areas for wind farms. An optimal wind farm location is close to where energy is consumed as the length of transport is associated with power loss. Additional environmental concerns include the avoidance of bird travel routes as well as the low frequency noises developed by the rotor blades, which might disturb wildlife or annoy inhabitants [47].

The Orientation of Wind Turbines. As mentioned earlier, the majority of wind turbines are represented by the horizontal axis wind turbine design. These turbines are commonly pointed upwind as the tower may produce turbulence. This makes it necessary for the rotor to be turned, with the help of a yaw motor, into the wind's main direction. Therefore, it is essential to know, or be able to forecast, the wind's main direction.

Health Monitoring of Wind Turbines. Existing wind power installations need health monitoring to remain operational and to reduce the need for maintenance. This monitoring may include overseeing mechanical loads, temperatures of components, and sensor drifts, among many other factors. Automatic shutdowns of components or other countermeasures must be triggered in time accordingly [47].

Grid Integration. Another challenge is to meet the power network's requirements such as predicting the power output and maintaining stability to be able to satisfy demand and supply. However, wind as a physical phenomenon depends on many factors making it highly variable and imperfectly predictable. Weather and wind forecasts are difficult and generally less accurate over longer time frames. As the output depends on the weather, it can change very rapidly resulting in power ramping effects. Additionally, wind turbines may ramp down to an output of zero to protect themselves against high speed winds. As these ramping effects need to be compensated for, the challenge is to stay independent of fossil energies for buffering or during times of peak demands [47].

Wind Energy Production Data

To address the aforementioned challenges and solve the problems associated with them, various types of data are collected. This *wind energy production data* often comes from multiple heterogeneous sources and different types of sensors. The data set may include recorded weather data containing temperature and humidity readings, recordings of precipitation, levels and wavelengths of incident solar radiation. The data includes recordings of wind and gust speeds along with their dominant directions. The wind speed is usually defined as the average air velocity over a chosen time frame, whereas the gust speed is defined by the highest recorded speed in this time frame. Additionally, barometric pressure levels from different locations may be recorded to estimate and analyze the development of winds. As the weather actively influences the power output of a wind park, understanding its influences and trends is important to network and power plant operators. For instance, the gathered weather data may be used to generate forecasts to be able to plan network loads ahead of time. Furthermore, values such as energy output, stability, and the effectiveness of individual wind turbines, as well as stress and temperature levels of components, may be gathered over time. Along with these, network loads and demands are monitored and recorded. Additionally, computer models for weather and climate estimation may produce local weather prognosis data to aid future planning.

As this recording of data is an ongoing process, data sets are continuously getting bigger and more complex. This vast amount of often raw data is usually worthless without further exploration, analysis, and presentation. The research fields of (Computer-) *Visualization* and *Visual Analytics*, which are discussed below, have developed solutions for exactly these kinds of problems.

1.2 Visualization

Visualization is the computer-aided technique of creating images or animations in order to communicate a message to a viewer. Visualization uses the remarkable perceptual abilities of the human's visual system and the brain's *visual cortex*. The visual cortex is the part of the brain responsible for processing any visual information. Humans can scan, recognize, and recall images in a fraction of a second. The brain is able to detect changes or patterns in size, color, shape, movement, or texture [87].

Visualization is valuable in many different application domains by providing a valuable assistance for data analysis and decision making tasks [95]. Depending on the source and purpose of the data, which is to be visualized, the research field is traditionally subdivided into the two areas *Scientific Visualization* and *Information Visualization*, which are both discussed below.

Scientific Visualization is the research field of generating a graphical representation of physical phenomena, which aims to assist scientific investigations. The goal is to discover things that might not be apparent in numerical form [75]. Scientific Visualization involves scientific data with an inherent physical component [95], i.e., data that represents or describes objects in the physical world [20]. Common data which is visualized includes spatial data (two- or three-dimensional geometries, tensorial fields, or vectorial data) but also spatial data with a temporal component, e.g., medical tomography scans recorded over time. Common visualization techniques include direct volume rendering, ray tracing or projection, two- or three-dimensional flow visualization and many more. Applications are found in every area where large amounts of data with a physical component are created and need to be processed. This includes medicine, archeology, aerospace engineering, biology, genetics, geology, satellite imaging, and many more. As an example for a scientific visualization, Figure 1.5 shows a two-dimensional flow visualization of weather data which was recorded during a hurricane event. The research in scientific visualization is mostly focused on how to effectively display huge data.

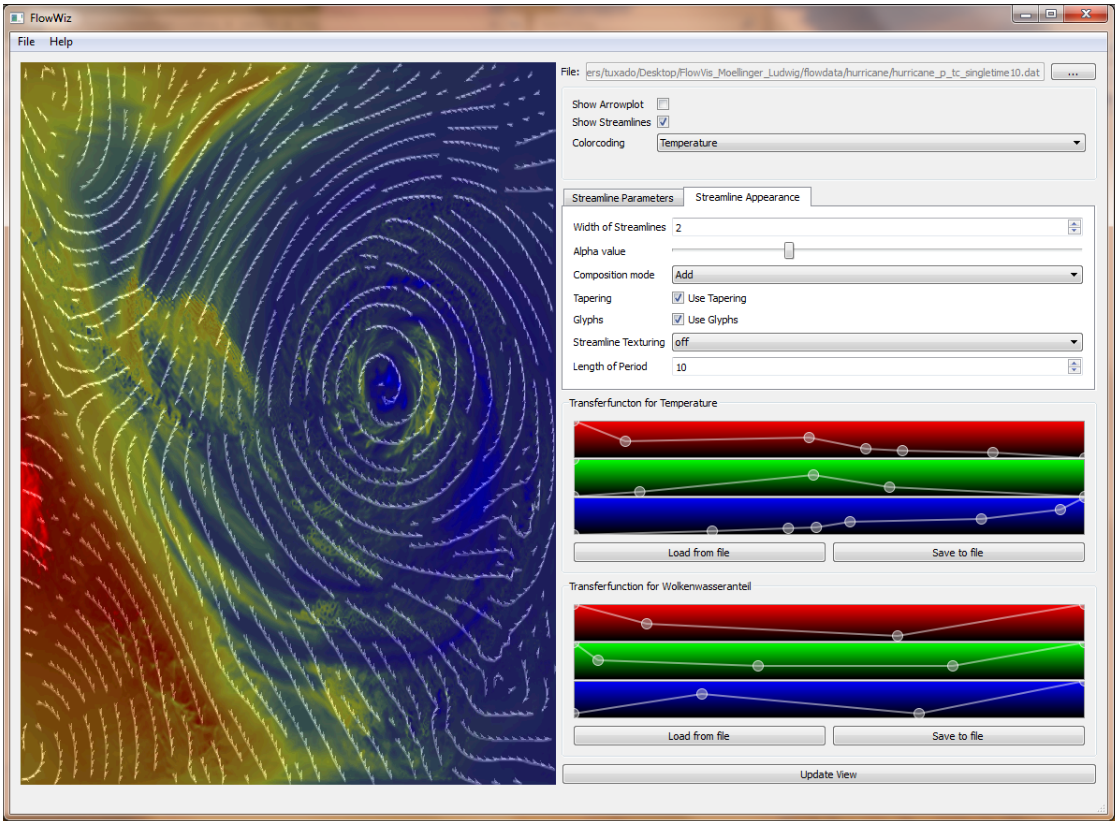


Figure 1.5: Scientific Visualization techniques are used to analyze data recorded during a hurricane event. The shown two-dimensional flow visualization illustrates wind flow directions and temperatures over a time frame. Image courtesy of Möllinger and Ludwig [71].

Information Visualization is the research field of creating images from abstract data that has, in strong contrast to data in scientific visualization, no explicit spatial reference. This type of data has no natural mapping and thus no trivial display space. Temporal or spatial components may occur but the data exists in an abstract, conceptual data space [20]. Furthermore, data often consist of more than just three, sometimes up to hundreds of dimensions. Hence the research involves the development of new visualization techniques to handle complex, multidimensional data with more than three dimensions. Common data sets which are visualized include, for example, stock market data, poll results, network graphs, and social webs. The challenge is how to effectively filter and then map and render this kind of data on the computer screen. Figure 1.6 illustrates an approach how information visualization techniques are used to gain insight into a complex and large data set.

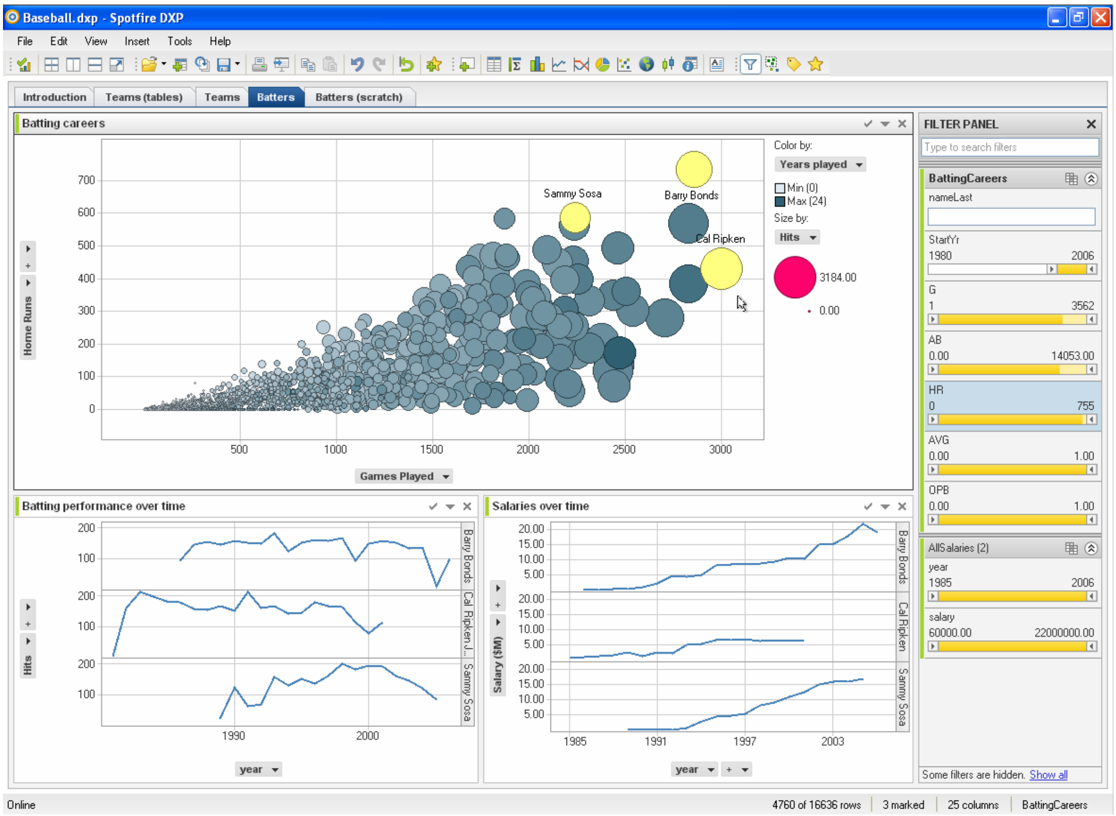


Figure 1.6: Information Visualization is used to gain insight into a large baseball sport data set. A scattergram is used to display some information of all recorded baseball players. Additional information on selected entities are visualized in time series plots below. Complex data filtering can be applied through the panel on the right. Reprint from Ben Shneiderman [88].

About Taxonomies

Rhyne et al. [81] discussed if the existing dichotomy of visualization provides a useful mechanism for progress or, in contrary, is creating confusion. In their discourse, Matt Ward [81] points out that Information and Scientific Visualization share the common goal of the visual communication "for the purpose of the presentation and exploration of data, concepts and relationships, and processes". Depending on the researcher's field of view, a separation might be hard or just artificial due to the fact that a lot of research topics are found in both research fields. Tory and

Möller [95] provided insight into how different visualization areas relate and overlap. Their work concluded that the used terminology is ambiguous and too fuzzy hence making them propose a new model-based, high-level visualization taxonomy [96]. Their novel taxonomy classified visualization based on the used algorithms instead of the underlying data.

Chris Johnson [81] explained that recorded scientific data often is of high dimensionality and thus not suited for scientific visualization methods. Furthermore, data sets may consist of data with spatial but also non-spatial components. For example, wind energy production data may include directional data, e.g., wind directions, but also non-spatial data such as the number of clients connected to the power grid or measurements of air temperatures or power outputs. In conclusion, research areas overlap and researchers are encouraged to break down the barriers between information and scientific communities and instead work together to create "scientific-information" visualization software systems [81].

The Goals of Visualization

Visualization acts as a tool to enable a user to gain insight into data. The mathematician Richard Hamming predicted this practice in 1962 by hinting that "The purpose of computing is insight, not numbers". In 1997, Daniel Keim [50] defined three major goals of visualization techniques:

Explorative Analysis
The user is confronted with data of which the user has no a priori knowledge or hypotheses about. The user interactively explores the data to find a visualization of the data which provides hypotheses about that data. This process is usually an undirected, thus unbiased, search.

Confirmative Analysis
At the beginning of the analysis the user already has one or more hypotheses about the data. To verify or reject hypotheses the user then examines the data in a goal-oriented way. The result of this analysis is a visualization, which confirms or disproves a hypothesis.

Presentation
The user has already found facts that should be presented. The challenge is to select the most efficient presentation technique, i.e., the one which conveys the learned insight best, and create a high-quality visualization, which is then presented to an audience.

Visual Information Seeking

In the year 1996, Ben Shneiderman [87] analyzed visual data exploration projects and discovered a basic principle all projects had in common. He called it *The Visual Information Seeking Mantra* that reads as follows:

"Overview first, zoom and filter, then details-on-demand"

It describes the process of going from an overview on the data top-down to the details of a data item. The user first gains an overview of the entire data, then identifies regions of interest and zooms into this region. The user may filter out uninteresting items and retrieve details of a specific entity.

With data sets continuously growing in size and getting more and more complex, it became clear that giving an overview over all data may not be feasible or efficient. It was this insight that marked the birth of a new research area called *Visual Analytics*.

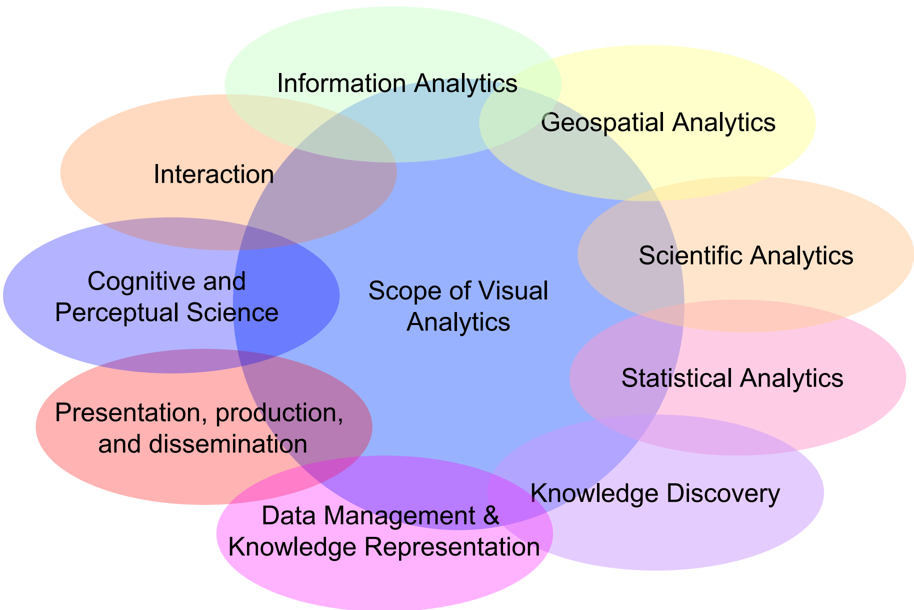


Figure 1.7: The scope of Visual Analytics as an interdisciplinary field of research. Reprint from Keim et al. [55].

1.3 Visual Analytics

Thomas and Cook [93] defined Visual Analytics as "the science of analytical reasoning facilitated by interactive visual interfaces". However, this definition did not reflect the usage of really big databases, so Keim et al. [52] gave a more specific definition by stating that "Visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets".

Visual Analytics combines the strong human capabilities of visual data processing with electronic, algorithmic data processing. Many scientific disciplines, as illustrated by Figure 1.7, contribute to the field of Visual Analytics, making it a highly interdisciplinary research field. Application areas include every area where large data have to be processed and analyzed, e.g., physics, biology, astronomy, geology, security, medicine, business intelligence, health care, the analysis of consumer or social data, and many more [53].

The Information Overload Problem

Nowadays, data is often collected and stored without filtering or refinement for later use. As storage space is not a problem anymore, the amount of data to deal with in nearly every industry or business is increasing very rapidly. At the end of the day, the goal is to extract information contained within this (usually) raw data which, for most applications, has no value in itself [52].

The *Information Overload Problem* refers to the difficulty for a human or a system to understand or make decisions about data caused by the presence of too much information. When confronted with too much information, the observer may get lost in data which is irrelevant to the current task. This holds also true for data that is processed or presented in an inappropriate way. The result of this information overload is that time and money is wasted, and opportunities are missed [53].

The field of Visual Analytics provides possibilities to overcome this problem. Keim et al. [55] underlines this by stating that “The goal of visual analytics research is to turn the information overload into an opportunity”.

The Goals of Visual Analytics

Keim et al. [53] stated that “Visual Analytics aims at making data and information processing transparent” for an analytic discourse. Further, they specified that the goal is to create tools, i.e., software, and techniques, e.g., algorithms, presentations, and interactions, to enable users to perform the following tasks:

- *“Synthesize information and derive insight from massive, dynamic, ambiguous, and often conflicting data”*
- *“Detect the expected and discover the unexpected”*
- *“Provide timely, defensible, and understandable assessments”*
- *“Communicate these assessment effectively for action”*

The Visual Analytics Process

Complex and large data sets cannot be rendered into a readable visual representation without analyzing the data before mapping it in a visualization. In the field of Information Visualization, it is the duty of the system designer to incorporate knowledge on the data to create readable and meaningful graphical presentations. This can, for example, be achieved by using knowledge on the likely topology of data or by composing visual abstractions. With a priori design choices, however, there is the risk of creating a representation that favors a specific interpretation. In fact, every representation that is not a presentation of the raw data will somehow favor an interpretation [55]. The process of Visual Analytics presents a solution to this problem by coupling automatic data analysis techniques with visual data exploration through human interactions [53].

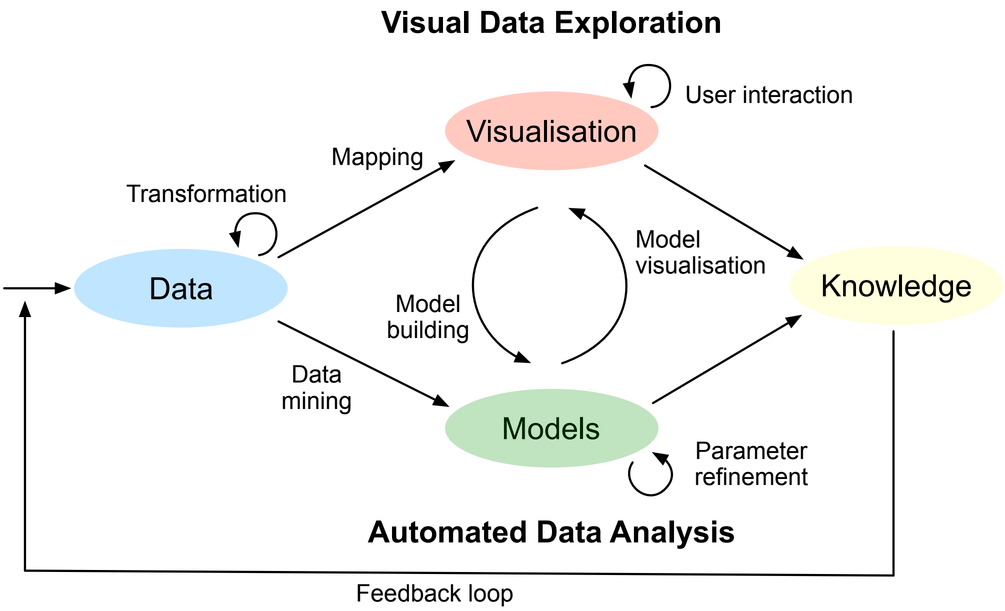


Figure 1.8: A flow diagram showing the process of Visual Analytics which promotes interaction between data, visualizations and models in order to gain insight into complex and big data. The feedback loop enables the user to confirm hypotheses from previous iterations. Reprint from Keim et al. [53].

The process, as illustrated in Figure 1.8, introduces a *feedback loop*. It allows the user to gain insight into the data and its graphical presentation likewise. This is achieved by giving the user the power to interactively manipulate the visualization as well as tune parameters that drive the automated data analysis behind it. Both manipulations result in new knowledge about, and new *views* on the data, i.e., adapted visual representations.

Alternating between visual data exploration and automatic data analysis methods allow the user to refine and verify preliminary results. Due to the fact that misleading intermediate results can be discarded at an early stage, the process leads to better results and a higher confidence, i.e., less uncertainty about what data is hidden or discarded for the graphical representation [53]. Each iteration in the feedback loop helps the user "to go beyond the visual and ultimately confirm hypotheses built from previous iterations" [52]. It was this process which led Keim et al. to rethink the information seek mantra and propose *the visual analytics mantra* [55]:

"Analyse First -
Show the Important -
Zoom, Filter and Analyse Further -
Details on Demand"

In contrast to Information Visualization, the process gives a "higher priority to data analytics from the start and through all iterations" [52].

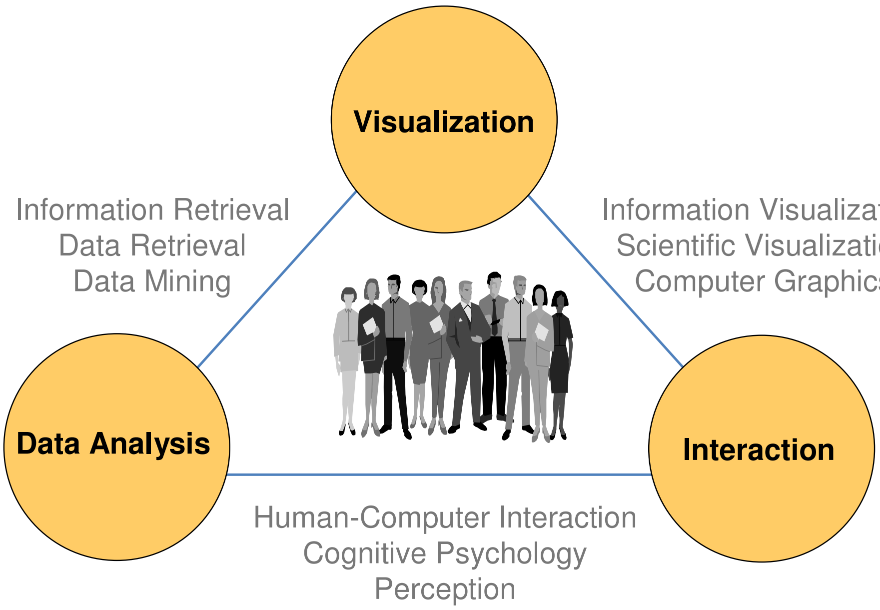


Figure 1.9: The constant interplay between data analysis, visualization, and interaction methods. Reprint from Keim et al. [56].

In conclusion, knowledge is extracted not only from visualizations or automatic analysis but also from the interactions between visualizations, data models, and the analyst [53]. Figure 1.9 illustrates this constant interplay.

1.4 Aim of this Work

The application background motivating this work is the need for better visualizations in the energy sector to address common challenges within the special field of wind energy production. The aim of this work is to improve and/or facilitate real tasks of real users based on real data in the context of this application domain. For this purpose, the work shall propose novel visualization and interaction techniques, which are especially well-suited for wind energy production data with its circular characteristics. Evaluated by a case study, and based on expert feedback on their deployment, these techniques shall provide analysts with a more intuitive way to visually analyze circular data such as data that contains wind directions.

The work shall reflect on the design process and design decisions to identify lessons learned, which may be applicable outside of the work's application domain. Moreover, the work shall discuss aspects of implementing the new techniques as parts of a larger visual analysis framework that is already in use by the domain experts. Therefore, the software's design and implementation details shall be presented and discussed.

All these actions shall provide data analysts with valuable insight into their data. Ultimately, this work shall save analysts time and hence companies resources and money.

1.5 Thesis Organization

Chapter 2 starts by providing an overview of existing information visualization and common interaction techniques. This is followed by the definition of a visual analysis framework and a discussion of radial visualizations as well as an explanation of two data types - *circular and cyclic data* - and an explanation of applicable radial methods.

Chapter 3 gives a problem characterization and abstraction by analyzing the target application domain, i.e., the visual analysis of wind energy production data.

Chapter 4 presents novel radial diagrams and interaction techniques, proposed by the *2D Radial View*, and discusses important decisions on their design. System integration and implementation details are given in Chapter 5. The proposed 2D Radial View is evaluated in Chapter 6 through a demonstration of its real-world application on wind energy production data.

Finally, the Chapters 7 and 8 provide the reader with a design reflection, discussion, and outlook on future work. Chapter 9 draws a conclusion about this thesis. Additionally, Appendix A provides mathematical fundamentals on the involved coordinate systems and transformations.

CHAPTER 2

Fundamentals and Related Work

This chapter explains the fundamentals and terms that are of importance to this thesis. After a brief introduction to common visualization and interaction techniques, a complete section is dedicated to the survey of previous work in the field of radial visualization. It is followed by an in-depth explanation of circular and cyclic data and a presentation of radial methods for these special kind of data.

2.1 Information Visualization Techniques

The consecutive subsections give an overview on visualization techniques that will be used throughout this thesis. Due to the fact that the research field of Information Visualization has brought up a vast number of different types of visualizations, for some of which additional derivations exist. Only the techniques most important for this thesis are discussed below.

Scatter Plots

A *scatter plot*, or *scatter chart*, or sometimes *scatter graph* is a type of diagram which visualizes the values of variables of a data set. In the case of only one data variable, the diagram plots each value as a point or ring along the axis of the data dimension. Such a very simple plot, also referred to as a *strip chart* because of its look, is depicted in Figure 2.1.

In the more common case of two or three variables, the related values forming an item are usually plotted within a Cartesian coordinate system. This visualization technique is often used in the field of statistics to visually analyze data dispersion, find correlations, or to quickly identify *outliers*. In the field of Information Visualization, the technique is used to present multiple data dimensions in a single visualization. Additional information can be added to each point by

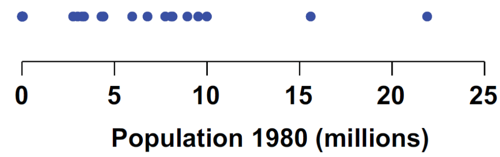


Figure 2.1: A one-dimensional scatter plot showing the population of different cities in the year 1980. Reprint from the book The Grammar of Graphics by Wilkinson et al. [101].

employing coloring, transparency, or point sizes, which extends the number of possible data dimensions that can be displayed in such a diagram. Scatter plots are only suitable for visualizing very large data sets when transparency or decluttering techniques are used due to the problem of overlapping or occlusion of the large number of plotted values. A three-dimensional scatter plot using coloring for displaying a fourth variable and rendering a medium-sized data set is shown in Figure 2.2 illustrating this problem.

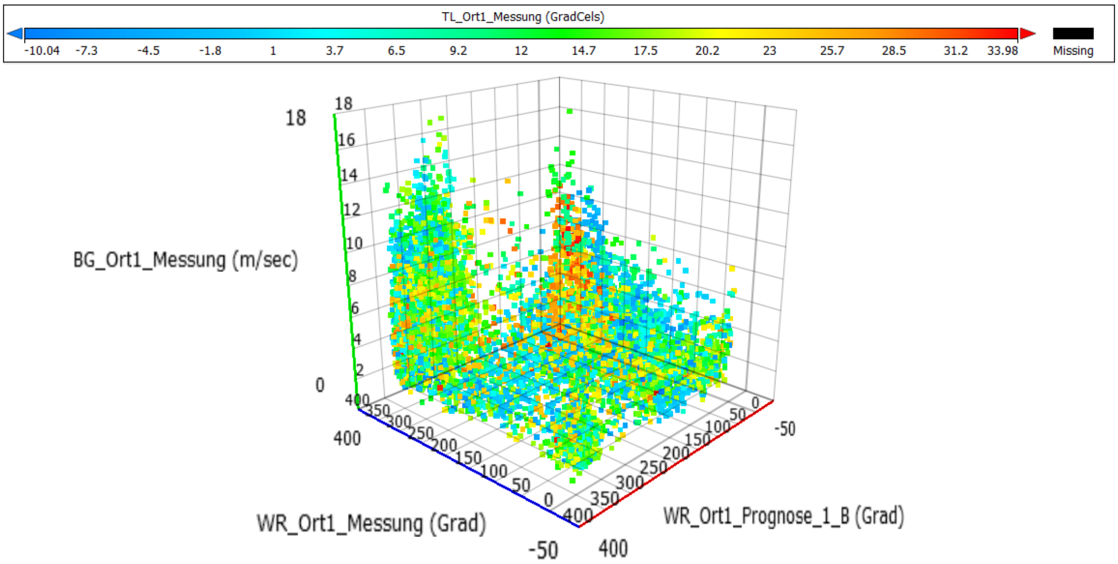


Figure 2.2: A three-dimensional scatter plot, which uses coloring for displaying a forth variable. Even the rendering of a medium-sized data set already results in overlaps and occlusions.

Histograms

A histogram is a graphical representation of the frequency distribution of a variable of continuous data. The continuous data is divided into non-overlapping and adjacent ranges, i.e., the bins, and the occurrences of variables falling into a bin are counted. Thus the histogram is a two-dimensional visualization incorporating the two dimensions count and the continuous data it relates to.

Commonly, the count of each bin is visualized using an erected rectangle or a bar, which has the width of the corresponding data range. The bins are usually of equal size but this is not required by design. Independent of the bin size, i.e., the width of the range, the area of the rectangle or bar is proportional to the frequency of occurrences in the bin. The count may be normalized and thus show the relative frequency instead of the absolute count for each bin to underline this property. Choosing a suitable number of bins is a crucial task for the expressiveness of the final representation as this directly influences the overall shape of the histogram. Figure 2.3 illustrates this by showing four histograms of the same data, but with different bin numbers. The shape of a histogram provides an estimation of the variable's probability distribution.

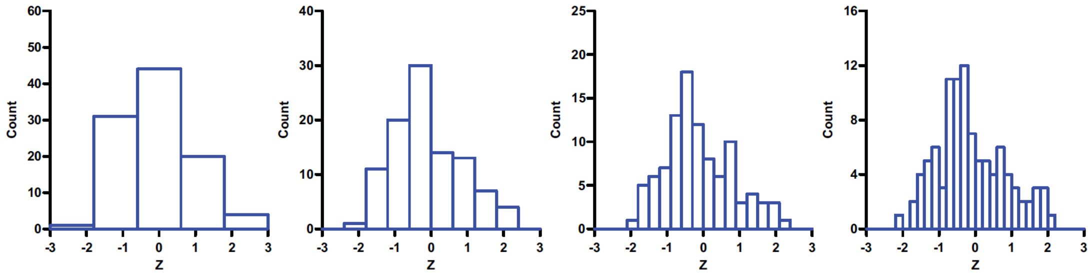


Figure 2.3: Four histograms of the same data but each with a different number of bins for the variable z. The histogram's shape changes from left to right but the area of each erected rectangle is always proportional to the count within each histogram bin. Reprint from the book The Grammar of Graphics by Wilkinson et al. [101].

Bar Charts

A *bar chart* or *bar graph* is a type of diagram, which visualizes one or multiple numerical properties of groups of data. Usually, filled rectangles are used to visualize the magnitude of this numerical property, thus the name bar chart. The bar's length therefore is proportional to the property's value. This way, visualizing bars of multiple data groups allows viewers an easy comparison among the presented values. The bar's width is usually arbitrary and gaps may be placed between bars.

Visualizing multiple numerical properties of one or more data groups, i.e., rendering bars for a group side-by-side, results in a *clustered bar chart* as demonstrated in Figure 2.4. Orthogonal to the technique of showing multiple numerical properties of a group side-by-side they can also be displayed in one single bar by stacking them. The result is a *stacked bar chart*, which shows such cumulative bars.

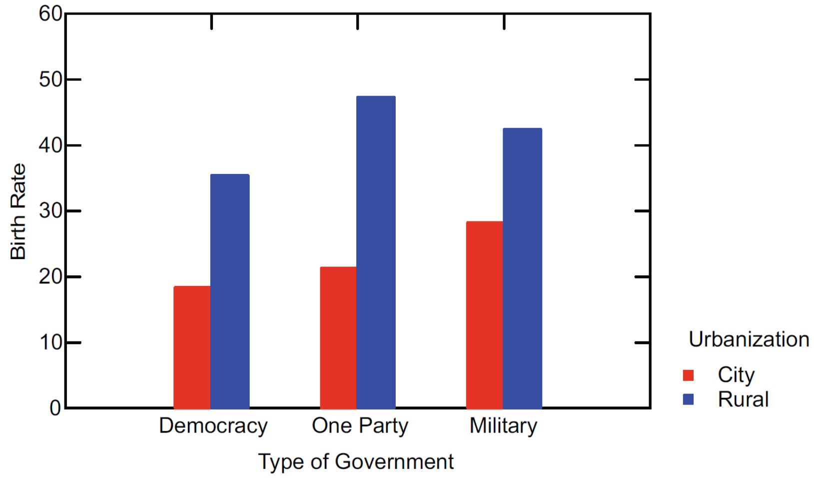


Figure 2.4: A clustered bar chart displaying the value of two properties for three different data groups. Reprint from the book The Grammar of Graphics by Wilkinson et al. [101].

Box Plots

In the field of statistics, a box plot is a visualization of a *schema*. A schema is a collection of relations such as points and intervals representing data distribution or density [101]. Hence, the box plot is a special form of a *schematic plot*.

The basic box plot, first conceptualized by Tukey [97] in 1970, conventionally visualizes five values from a set of data: the upper and lower extreme, the upper and lower hinges (quartiles), and the median [67]. The *inter-quartile range* (IQR), that is the range between the lower and upper quartile, is plotted as a box, thus the name. In its original form the width of the box is fixed and carries no information although variants with variable width boxes and notched boxes are possible [67]. The median is drawn as a line splitting the box. The inter-quartile box is extended by line segments to show the complete data range. Due to the resulting appearance, the box plot is also known as a *box-and-whisker plot*.

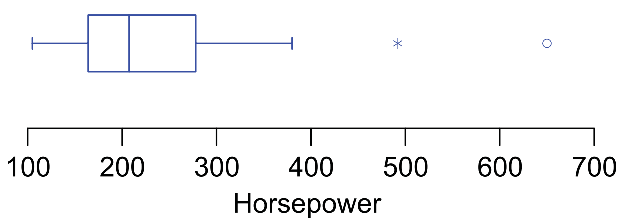


Figure 2.5: A box plot showing a distribution of horsepower among cars in a data set. A strong outlier is plotted as a star and an extreme outlier is plotted as a circle. Reprint from the book The Grammar of Graphics by Wilkinson et al. [101].

It is common practice to limit the shown data range, e.g., to a range from the fifth percentile to the ninety-fifth percentile, to exclude statistical outliers. Outliers may then be drawn by glyphs to attract attention to them. Figure 2.5 shows a typical box plot with two emphasized outliers.

Box plots are widely used as a graphical technique in exploratory data analysis and when producing visual summaries as they offer a way for the user to identify important statistical values quickly and gain insight into the distribution and deviation of data.

2.2 Common Interaction Techniques

Interaction plays an important role in the fields of information visualization and visual analytics. Whereas the roots of visualization techniques lie in the field of computer graphics, interaction techniques emerged from the field of human computer interaction (HCI). Although visual representations are often discussed completely separately from their interactions, without interaction a visualization becomes static and limited in its usefulness [103]. Interaction supports the user in the process of analytic reasoning. Hence, interaction techniques are an essential part of any information visualization system. The most common interaction techniques are presented below.

Selections

A selection describes the partitioning of a data set into exactly two subsets, i.e., into the set of selected entities and the set of unselected entities. By applying Boolean operations two sets can be used to generate a new set. Interactively defining selections is an essential tool for the visual analysis of large data sets as it enables analysts to refine a big data set down to a smaller data set and thus focus on the entities, which might be of higher interest. This *interactive filtering* of data can be accomplished by either *browsing* or *querying* data. Browsing is the process of performing a direct selection of a subset in a visualization. This is carried out mainly by the user by interacting with the visualization with the mouse and/or keyboard. Querying, on the other hand, describes the process of specifying properties, i.e., rules, for which the software then automatically produces the subsets based on these specifications [51].

Data Brushing and Visualization Linking

Data brushing describes the technique of interactively selecting a subset of visualized data through a painter's brush analogy. An input device, usually a computer mouse, is used to paint, thus define, a selection in a visualization. The brush's shape (e.g., a rectangle, a circle, or a cylinder) and its coverage can vary based on the visualization space used and/or interaction mode. Data brushing is therefore an advanced form of *browsing* data.

Brushing is an extremely powerful tool when it is combined with *visualization linking*. Linking is the technique of presenting how data brushed in one visualization is integrated in another visualization. For this purpose, brushed data is highlighted, for example, in realtime, or immediately after the painting process in all visualizations that display corresponding data. The corresponding parts are marked in all visualizations by applying a special color or by using some other form of highlighting technique [11]. As a result, linking enables analysts to detect dependencies and correlations among multiple data dimensions. Due to all visualization techniques having weaknesses when visualizing multi-dimensional data, Daniel Keim concludes that "the idea of linking and brushing is to combine different visualization methods to overcome the shortcomings of single techniques" [51].

Displaying different projections into multi-dimensional data greatly benefits from the linking technique. For example, Figure 2.6 displays two different projections (two-dimensional scatter plots) of a multi-dimensional data set. Performing brushing in the left visualization is reflected by highlighting matching points on the right. This way visualization linking integrates scattered information "by marking corresponding parts of multiple displays" [11].

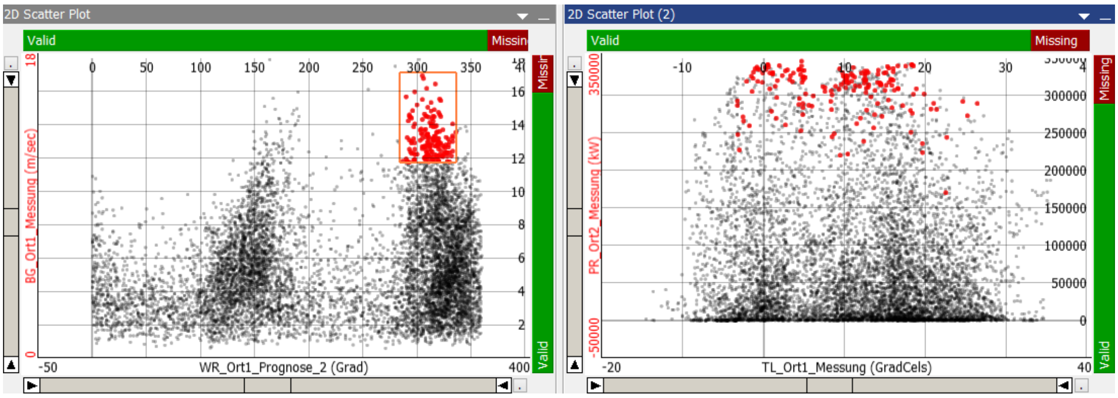


Figure 2.6: Two linked projections (two-dimensional scatter plots) of a multi-dimensional data set. Performing a brushing action in the left visualization is reflected immediately by highlighting matching points in the right visualization. This brushing and linking technique enables users to detect dependencies and correlations among multiple data dimensions.

Overview + Detail

In general, the complete information space cannot be projected in full detail onto the usually much smaller available display space. The *Overview + Detail* technique addresses this problem. The term *Overview + Detail* describes an interface design that provides the user simultaneously or sequentially with an overview and one or multiple detail views of parts of the same information space. Usually the overview and the detail view(s) are physically separated, shown in parallel, and operated individually by the user [18].

The overview visualizes the overall pattern and must neglect details for this purpose. Detail views on the other hand show only a portion of the full information space but are dedicated to preserving details. As a detail view displays only partial information, Buja et al. [11] point out that such a view "should not be regarded in isolation". They note that views "need to be linked so that the information contained in individual views can be integrated into a coherent image of the data as a whole". Consequently Overview + Detail systems usually implement visualization linking.

2.3 Visual Analysis Frameworks

A visual analysis framework is a computer software, which integrates various visualization and interaction techniques to support users to perform an effective visual analysis of data. Frameworks are highly modular and allow the user to freely choose and combine these tools and features. Figure 2.7 depicts a state-of-the-art visual analysis software in action. The requirements for a visual analysis framework as well as its key features are discussed below.

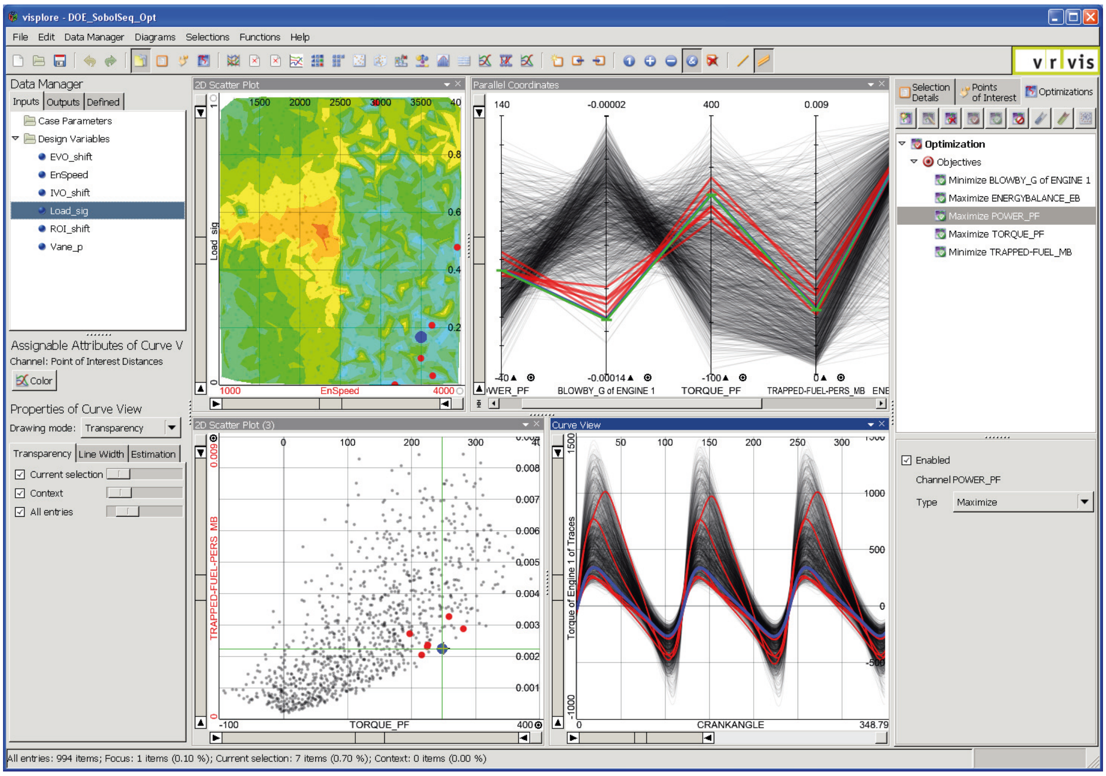


Figure 2.7: A screenshot of a typical visual analysis software framework showing multiple linked views. Image courtesy of VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH [30].

Requirements

Visual analysis software needs to solve a number of technical challenges. For example, such a software system has to be responsive after a user has triggered a computation on the underlying data or during a redraw of a visualization. This makes parallel or multi-threaded computing techniques compulsory. Such systems are increasingly confronted with a huge amount of data, thus have to be highly scalable. Ideally, the system also has to be versatile, as it might have to handle a lot of different kinds of data, data sources, data formats, and tasks.

Furthermore, the system has to be easily extensible to allow developers to cover new user tasks by providing novel task-oriented computations and visualizations. In 1996, Ben Shneiderman made this perfectly clear by stating that any visual analysis framework that wants to be successful as a software product "will need to provide smooth integration with existing software and support the full task list: Overview, zoom, filter, details-on-demand, relate, history, and extract" [87]. The most notable and typical features of a visual analysis software are presented below.

Multiple Linked Views

State-of-the-art analysis software typically allows users to make use of an arbitrary number of simultaneously opened views. Views are automatically linked if they present any corresponding data. Figure 2.7 shows a system with four opened and linked views. The theoretical maximum number of possible views is limited only by available display space, and system resources. Modern software suites allow users to detach individual views from the main software window. Views can be placed freely anywhere on the screen. By employing multiple computer screens, a vast number of views can thus be displayed, used or observed simultaneously. However, an optimal number of views is often found in the lower single digits, as "the big picture can get lost in the details" [45]. Similar to information visualization systems data brushing techniques can be applied inside a view. Consequently, changes to selections are immediately reflected in all of the corresponding, i.e., in all linked, views. Therefore, linked views are a great way to "guide a story" and "encourage user exploration" [45].

Data Aggregations and Computations

The visual analysis process makes heavy use of data statistics and computations. Providing summary statistics, such as data aggregates, is a common way to communicate information. Data aggregations are built by combining multiple data entities into aggregated data. This aggregated data may not only be used for visualization but also for further calculations, filtering, and so on. Widely used data aggregations include the arithmetic mean (Equation 2.1) and metrics of order statistics such as: the minimum (Equation 2.2), the maximum (Equation 2.3), the statistical range (Equation 2.4), the statistical median (Equation 2.5), and other percentiles.

Data computations are the concept of applying an algorithm, i.e., the "computation", on data. In this process new data is generated from the input data and the used computation. Such data computations operate either entity-wise, e.g., calculating the logarithm for each entity, or on a wider range of entities and thus perform more complex operations, e.g., clustering or convolution. Data computations are an important tool during the stages of data transformation and model refinement in the visual analysis process.

Given all variables of a sample of size n, denoted by a_i, the order statistics, denoted by $a_{(n)}$, are defined by sorting the variables a_i in increasing order. The subscript n denotes the n-th order statistic of the sample. The arithmetic mean $\bar{a}$ is then given by:

$$\bar{a} = \frac{1}{n} \sum_{i=1}^{n} a_i \tag{2.1}$$

The minimum a_{min} is given by:

$$a_{min} = \min_{i=1}^{n} a_i \tag{2.2}$$

The maximum a_{max} is given by:

$$a_{max} = \max_{i=1}^{n} a_i \tag{2.3}$$

The statistical range a_{range} is given by:

$$a_{range} = a_{max} - a_{min} \tag{2.4}$$

The statistical median $\tilde{a}$ is given by:

$$\tilde{a} = \begin{cases} a_{\left(\frac{n+1}{2}\right)} & \text{if } n \text{ is odd} \\ \frac{1}{2}\left(a_{\left(\frac{n}{2}\right)} + a_{\left(\frac{n+1}{2}\right)}\right) & \text{if } n \text{ is even} \end{cases} \tag{2.5}$$

2.4 Radial Visualizations

Diehl et al. [21] define *radial visualizations* as visualizations that have "a dominating radial structure like a ring, an ellipse, or a spiral, in other words, visualizations in a polar coordinate system." To many radial visualizations a non-radial (usually Cartesian) equivalent exists and vice versa. The needed mathematical fundamentals for the conversion from polar to Cartesian space and its inverse are explained in a separate chapter, which is located at Appendix A of this thesis.

The term *radial visualization* was first introduced in the year 1997 by Hoffman et al. [39] although radial visualizations existed long before in the mathematical field of statistics. The first occurrence of a *pie chart* even dates back to the year 1801, i.e., Figure 2.8 shows a comparison of European nations metrics side by side using pie like representations. It uses sectors to divide the Turkish and Russian Empire into their continental affiliation. Draper et al. [24] identified popular radial statistical displays like the *pie chart*, the *star plot*, and the *radar plot* as the predecessors of today's state-of-the-art radial visualizations.

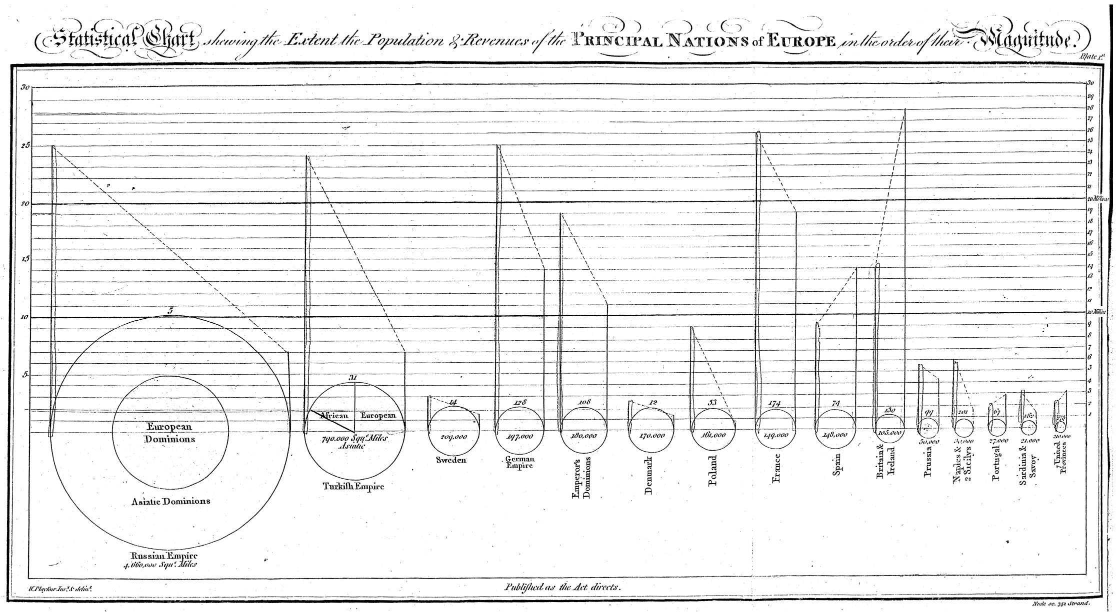


Figure 2.8: A statistical chart from the year 1801 showing extent, population and revenues of nations in Europe. Reprint from the book *The Statistical Breviary* by William Playfair [79].

An overview of the most common application domains of modern radial visualizations, a classification by visual appearance with examples, a discussion of 2D versus 3D, as well as the strengths and weaknesses of radial visualizations are given in the following sections.

Application Areas

Draper et al. [24] did a survey of radial methods for the purpose of information visualization and identified four distinct application areas for which radial visualizations seem especially well suited. They are:

- serial periodic data
- hierarchical structures
- relationships among disparate entities
- ranking of search results

Depending on the targeted application domain and involved data types a different design of the used radial visualization may be required. Thus the most common radial design patterns are discussed in detail below.

Design Patterns

Today radial visualizations exist in a variety of shapes and forms. Based on the visual appearance of all known radial methods Draper et al. [24] classified three main divisions:

- Polar Plot
- Radial Space Filling
- Ring

This rough division was not precise enough to reflect the vast abilities of all systems. Therefore, they further subdivided their scheme into seven high-level design patterns which, according to them, "describe nearly all radial visualization systems built in recent years". Table 2.1 shows the classification and its subdivision into seven distinct radial design patterns.

Scheme	**Design Pattern**
Polar Plot	Tree Star
Radial Space Filling	Concentric Spiral Euler
Ring Based	Connected Ring Disconnected Ring

Table 2.1: A classification by design for radial visualizations into schemes and design patterns proposed by Draper et al. [24].

The following gives a brief introduction to the three main groups of radial visualizations and shows examples for each design pattern within.

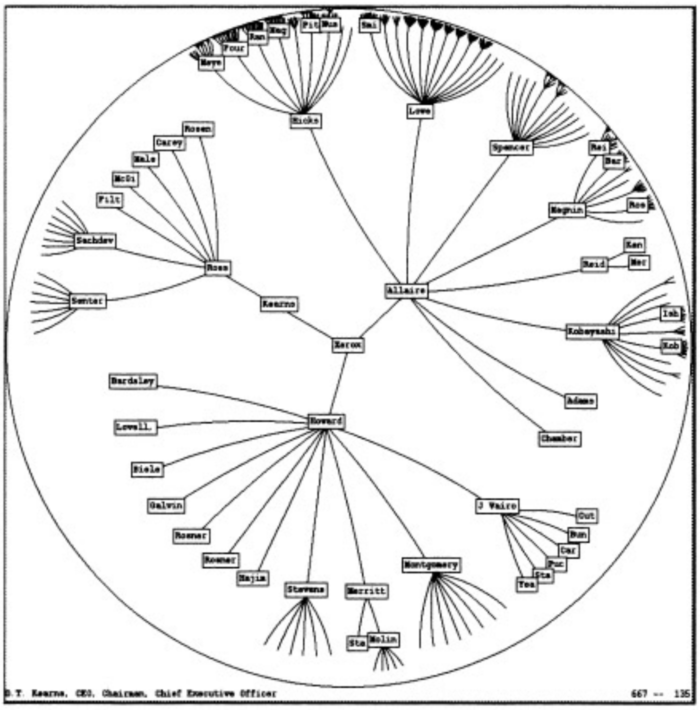
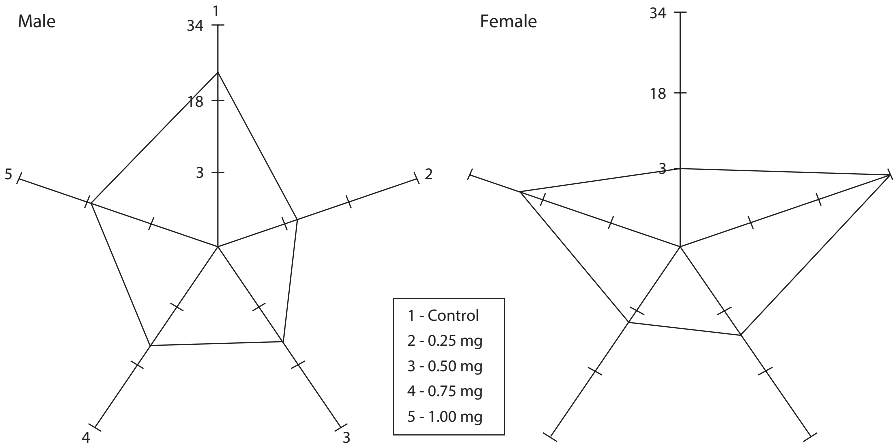


Figure 2.9: The Hyperbolic Browser system showing an organizational hierarchy. The system allows users to interactively explore very large hierarchies. Due to the polar plot design, large trees can be visualized on very little display space. Reprint from Lamping and Rao [61].

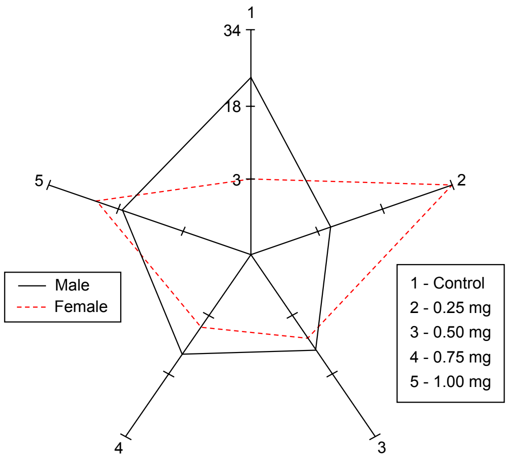
Polar Plot

In a polar plot the center of the visualization plays an important role. This central point can represent the origin of a coordinate system or the root node in a hierarchy. Other entities are placed or extended outwards from this point in a radial way. The polar plot scheme can be subdivided into the *Tree* and the *Star* design pattern, which are discussed below.

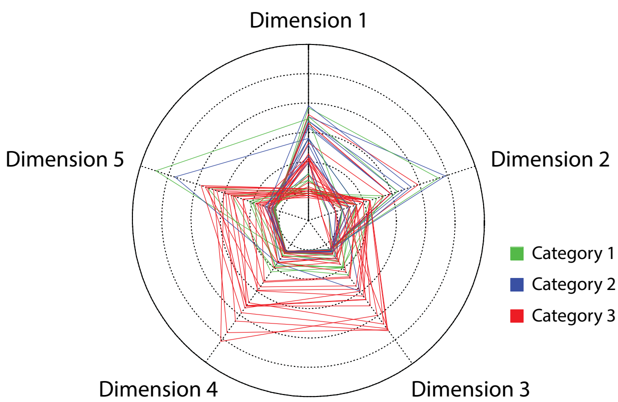
The *Tree* design pattern in principle originated from radial graph layout algorithms [7,13,80] that were designed to visualize hierarchical data structures such as trees efficiently. A tree-pattern based visualization is characterized by line segments that radiate outwards from the center node. Those line segments branch off recursively until some leaf node is reached and rendered. Intermediate nodes, which are nodes at positions where branching happens, may also be rendered if they are of importance. Most radial graph visualization systems are visualized in two dimensions but several extensions to the three-dimensional space, by using projections in or onto a sphere, or by arranging sets of cones, exist [14, 35, 48, 74, 82]. Figure 2.9 shows the *Hyperbolic Browser* [61,62] as an example for an interactive radial visualization system, which follows this design pattern. The depicted system is able to visualize large hierarchies and offers the user a focus and context technique for an interactive exploration of the graph.



(a) Two star plots used to compare side-by-side the male and female results of a clinical study. On each axis, representing a treatment group, the death rate is plotted and connected to form a geometric figure. Reprint from Joan Saary [83].



(b) The information of two star plots is rendered on top of each other. The resulting radar plot uses different line styles and colors to visually separate the two polygons. Reprint from Joan Saary [83].



(c) A typical radar plot showing multivariate data. Coloring is used to highlight the membership to a category. Figure adapted from the book *The Grammar of Graphics* by Wilkinson et al. [101].

Figure 2.10: Examples for the *Star* design pattern.

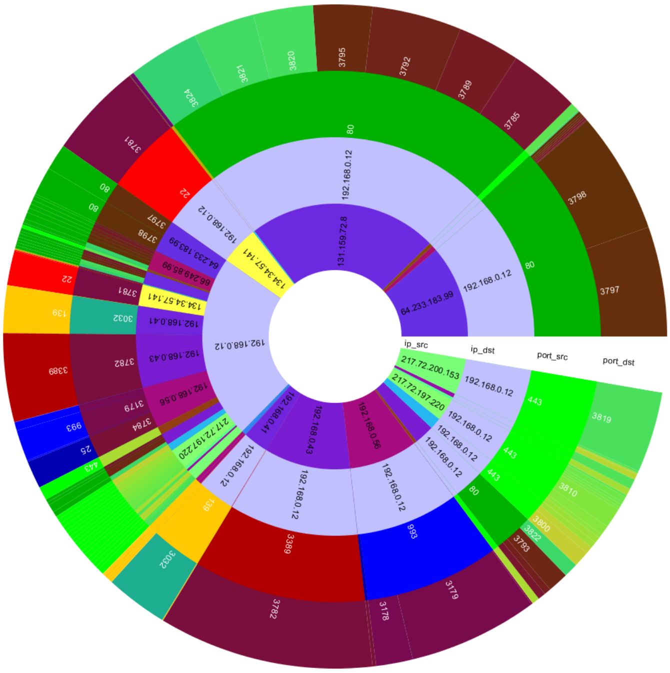
Visualizations following the *Star* pattern are best described as diagrams having “a series of spokes or rays projecting from a central point” [83]. Although both patterns use line segments radiating outwards from a mutual origin, no branching occurs in the *Star* pattern. Due to this distinctive property, the *Star* pattern is more compact and less visually challenging than the *Tree* pattern. However, Draper et al. [24] point out that the pattern therefore “[...] offers less variety in its visual representation and affords fewer options for interactivity”. Nevertheless, the pattern is often used to visualize the ranking of search results or to visualize relationships among disparate entities [24]. The design pattern has it roots in the *star plot*, which is also known as *spider web* or *kiviat diagram* [60, 73], which is very prominent in the field of statistics. Star plots are essentially the radial equivalence to parallel coordinates [46], which are designed to view and explore multivariate data sets. The parallel coordinate axes are plotted as equi-angular rays reaching out from the center of the diagram. The values plotted on those axes are connected by line segments forming an enclosing figure referred to as a *star*. Such a star presents the data with a tangible shape and size to the user. Figure 2.10(a) shows two star plots and their resulting star figures. When more than one star at a time is rendered the visualization is called a *radar plot* [83, 101]. Figure 2.10(b) combines both star plots of Figure 2.10(a) into one radar plot in favor of a better visual comparability. When the star figure is displayed as a filled polygon of its hull a comparison of multivariate data without occlusion is only possible if one figure is contained within the other or when transparency is used [24]. Moreover, with an increasing number of stars the plot will become too visually cluttered as can be observed in Figure 2.10(c).

Radial Space Filling

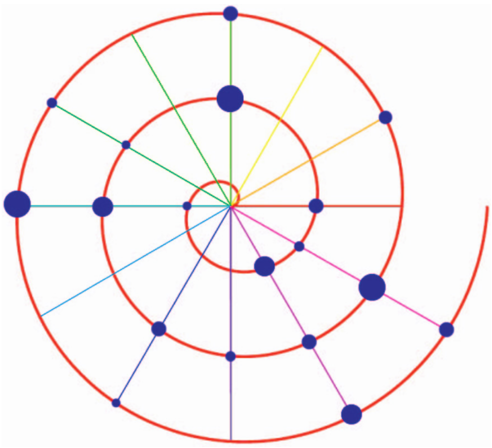
Radial Space Filling (RSF) visualization methods typically arrange data spatially with the intent of filling out all assigned space in an efficient manner. Stasko and Zhang [89] point out that the term “space-filling” might be misleading as radial techniques do not cover all (rectangular) display space. Instead, these visualization methods try to occupy all the available space of a circle as best as possible. The term *Radial Space Filling* and its abbreviation *RSF* was introduced by Yang et al. [102] in the year 2002 to describe the InterRing system. Within this category, Diehl et al. [21] identified the following three design patterns: *Concentric*, *Spiral* and *Euler*, which are discussed below.

The *Concentric* design is based on rings fanning outwards from a mutual center. Each of these rings can be divided into multiple sectors to convey information. Figure 2.11(a) shows such a visualization using concentric rings and colored sectors to give an overview of the traffic and data flow in a computer network.

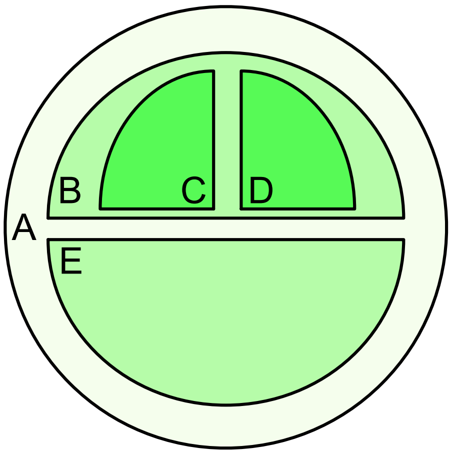
The *Spiral* design is characterized by structures that are laid out in the form of a spiral starting from an origin. Although many two-dimensional forms of spirals exist (e.g., the *Euler spiral*, the *hyperbolic spiral*, or the *logarithmic spiral*) the compact *spiral of Archimedes*, which moves away from the center with a constant rate, is commonly used. The spiral character itself can be used as a data dimension, e.g., to reflect the progress of serial (periodic) data. The design



(a) The Radial Traffic Analyzer system [54] showing the network traffic and flow in a computer network. Four concentric rings are used to display the information of four different dimensions. Reprint from Keim et al. [54].

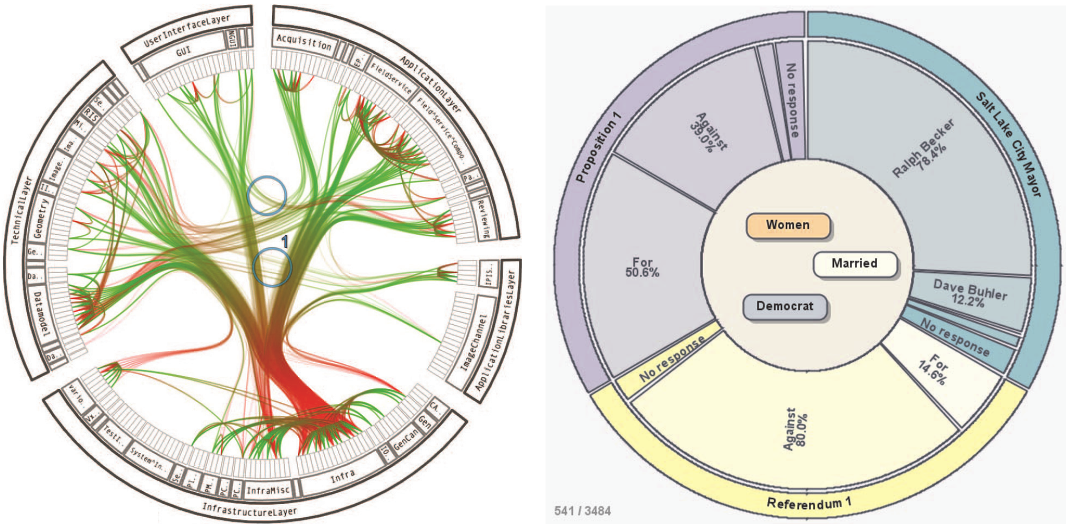


(b) A spiral plot encoding the number of events per month (blue circles) on a time axis (red spiral). Reprint from Draper et al. [24].



(c) A radial Euler diagram depicting the sets A, B, C, D, and E, and their relationships. Reprint from Mohammadi-Aragh and Jankun-Kelly [72].

Figure 2.11: Examples of the three different *Radial Space Filling* design patterns.



(a) Nodes of a hierarchy are placed on the circumference of a circle. Relationships between nodes are indicated by a line connection. Hence, the visualization implements the connected ring design pattern. Image adapted from Danny Holten [40].

(b) The SQirl system uses unconnected rings to visualize the results of opinion polls. Relationships are indicated by colors and positions of the sectors instead of line connections. Hence the system implements the disconnected ring design pattern. Image adapted from Draper and Riesenfeld [25].

Figure 2.12: Examples of the two *Ring Based* design patterns.

pattern is often used for the visualization of time-based data [9, 23]. According to Carlis and Konstan [12], the spiral-shaped layout can help users to detect recurring patterns in the periodicity of data that would not be "evident from other views of the data set". Figure 2.11(b) shows a spiral-shaped plot that maps the progress in time of one year to a spiral progress of 360 degrees.

The *Euler* design pattern relies on circles that are nested into outer circles to reveal hierarchical structures or relationships. This was first conceptualized by Euler in his work on logic diagrams [6], hence the name *Euler diagram*. Figure 2.11(c) shows a radial space filling Euler diagram that visualizes a hierarchical (tree) structure.

Ring Based

As the name indicates, visualizations in this scheme are based on a ring-like placement of entities. The common use is the exploration of relationships among those entities [24]. The rings and their contained entities are mostly placed inside the circumference of a radial structure making them a kind of space-filling visualization although the inner region might not be filled entirely. This scheme is further subdivided into the *Connected Ring* and *Disconnected Ring* design pattern.

The *Connected Ring* pattern is used to visualize relationships among nodes in a data set. This is achieved by placing nodes on the circumference of a circle and drawing edges between relating nodes. Those edges or links are rendered as direct lines or as bent connections, e.g., *B-Splines* [19]. Some systems place histograms or other common visualizations around or inside nodes thus giving the system the possibility of conveying additional information or context. The inner space of the visualization is primarily used for displaying connections. Those connections are predisposed to intersect or overlap, thus reducing legibility. This problem, referred to as *visual clutter*, increases with the number of nodes and connections in a system. One solution to this is to bundle visually-adjacent edges which only branch out near the nodes as proposed by Holten et al. [40] and shown in Figure 2.12(a). The ModelBrowser system [104] adds a great number of interaction possibilities to aid biologists in reasoning about large data models. Such methods can be combined with hierarchical techniques, which further improve scalability and therefore usability of large graphs. Kienreich et al. [58], for example, use hierarchical abstraction, edge routing, and bundling. Another solution is to move and distribute important nodes to the inner space [64] and thereby reduce the number of intersections.

The *Disconnected Ring* design poses a solution for the scalability issue regarding visual clutter discussed in the *Connected Ring* pattern above. The fundamental property of this design pattern is that no links between nodes are explicitly rendered. Instead, nodes expose their relation via attributes such as color, position, shape or labels, or a combination of these. The SQiRL (Simple Query Interface with Radial Layout) system [25] visualizes the results of opinion polls using disconnected nodes placed in sectors of concentric rings. Each ring drawn closer to the center refines the result of the poll or survey. Relations are show by the coloring and positioning of nodes as can be seen in Figure 2.12(b). The interior of the rings can be used to limit the poll's results to a specific set of users, for example, only contributors over the age of 40 that are female.

2D versus 3D Radial Systems

Draper et al. [24] identified that the majority of today's information visualization systems that use a radial layout are based on a two-dimensional design rather a three-dimensional one. Although adding an additional dimension results in more visualization possibilities it also makes navigation and interaction more complex. Draper et al. believe that the existing trend for two-dimensional systems is due to occlusion issues [28] that occur in (but are not limited to) three-dimensional graphics.

Cockburn [17] investigated implications on spatial memory by confronting users with similar tasks on 2D and 3D graphics. Results disagreed with previous work done by Tavanti and Lind [91] that stated that perspective effects make a difference in effectiveness. Thus it remains unclear if three-dimensional visualizations provide any spatial memory benefits.

Strengths and Weaknesses

Radial visualizations offer certain advantages, but also limitations and drawbacks, caused by its radial presentation. Diehl et al. [21] analyzed and summed up the strengths and weaknesses of many different radial visualizations. They compared radial systems against their Cartesian counterparts through an empirical approach. They measured the speed and accuracy of participants when faced with the same task from a pool of generalized user tasks in both layouts. They uncovered a learning effect, which showed that radial visualizations are harder to learn, however not harder to read for trained users. Initially, they expected a stronger learning effect for radial visualizations but had to reject this hypothesis.

Their empirical work concluded that "Cartesian visualizations tend to outperform their radial counterparts especially with respect to answer times". For this reason Diehl et al. [21] endorse Cartesian layouts as the default choice, but acknowledge that radial visualization seem to be more appropriate for focusing on a particular data dimension or "when depicting two dimensions that are not equally important". However, they pointed out that not all design decisions regarding the choice between a Cartesian or radial layout can be covered by their empirical approach. This holds true especially if there exists some natural representation of the data set, which suggests the usage of a certain layout. As such, data sets containing time-oriented or directional data may feel more natural if presented to the user by a radial visualization. As Aigner et al. [1] pointed out, a lot of natural processes are cyclic/periodic and therefore better visualized by an radial visualization layout.

Due to the fact that wind energy production data sets usually contain time-oriented and also directional data, these data types and radial diagrams that were built especially for the purpose of such types of data are discussed below.

2.5 Circular and Cyclic Data

Circular data is data that can be mapped meaningfully to the circumference of a circle, i.e., to angles in the range of $[0,\ 360)$ degrees or $[0,\ 2\pi)$ radians [42]. The circumference of a circle represents a closed space for which a concept of an origin, a start or an end is usually arbitrary or undefined. This issue is further discussed in Section 3.2 on the circular closure of circular data. Hussain et al. [41] pointed out that for this reason, this kind of data implies the need for techniques different from those used for other data types.

The best known data types that fulfill the criteria of being circular are two-dimensional directions, angles, and time. Angles satisfy the criteria as they encode differences of directions. Time is circular due to the human concept of 24 hours of the day representing one rotation of the earth around its own axis or the concept of 12 months as the earth's rotation around the sun.

The term *cyclic data* was used by Ward and Lipchak [99] to describe data sets that contain cyclic or periodic behavior. For instance, the event of the first snowfall of the year recorded in a data set over many years classifies as cyclic data. Radial layouts offer a way of revealing cyclic or periodic information (e.g., trends) of such data.

The next subsections discuss visualization techniques for these two data types concentrating on temporal and directional data. It is followed by popular radial methods used to visualize and analyze such data.

Visualization Techniques for Circular and Cyclic Data

Ward and Lipchak [99] analyzed visualization techniques for circular and cyclic data and their corresponding layout strategies for plots. They identified three design strategies when building systems for the visual analysis of such data types:

- Linear Techniques
- Circular Techniques
- Cyclic Techniques

Linear techniques plot the cyclic data dimension, usually time, linearly on one axis, mostly horizontally, and additional axes are used to encode additional variables. According to Ward and Lipchak [99], such visualizations rely massively on the human vision system to detect patterns. The approach is limited due to the fact that the user has to scan the plot for cycles and then compare cycle by cycle for changes. As a result of this, cycles that are not adjacent are hard or even impossible to compare.

Circular techniques are used to display aggregations of data that have a circular component using the circumference or the inside of a circle. An example is the *circular bar plot*. The circular bar plot maps an aggregate of a variable to a circular category, i.e., data dimensions whose range either is or can be treated meaningfully circular. Thus a circular bar plot is the polar counterpart of a Cartesian bar plot as demonstrated by Figure 2.13. Such visualizations summarize the intra-cycle relationships but cannot reveal any inter-cycle relationships [99]. The big advantage of these techniques is that comparing the begin to the end of a cycle is uncomplicated due to their spatial coherence caused by the circular layout.

Cyclic techniques preserve intra-cycle and inter-cycle relationships of the data. This is commonly achieved by using a spiral based radial visualization. Such a layout allows the user to compare values along a circle, angularly, as well as from a cycle to another cycle, i.e., radially. Although cyclic techniques overcome the obstacle of how to show intra-cycle and inter-cycle relationships simultaneously, Ward and Lipchak [99] concluded that no single technique is useful for all tasks in the field of Visual Analytics.

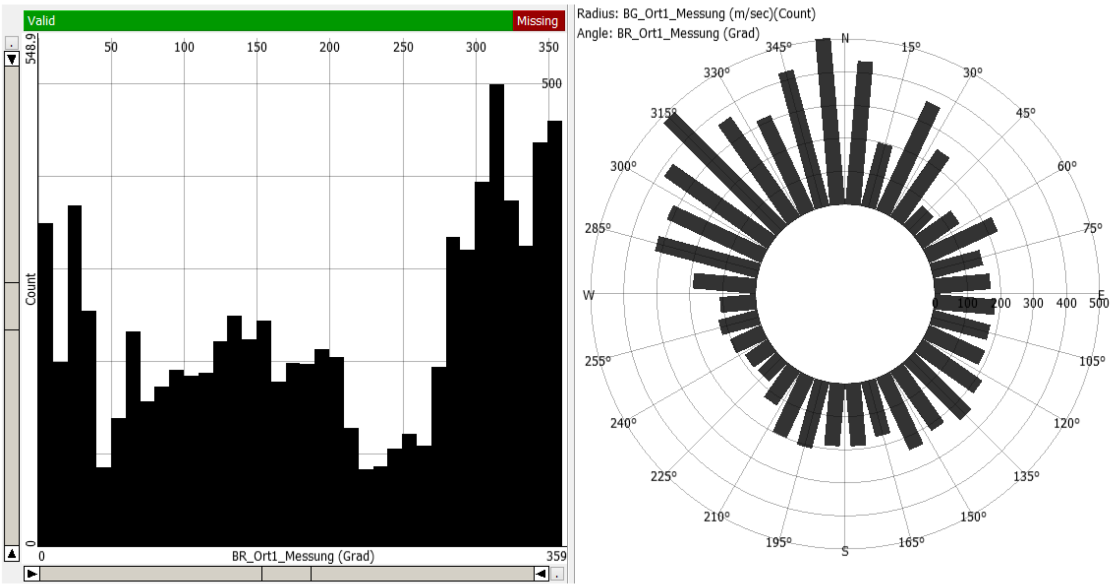


Figure 2.13: A Cartesian bar plot on the left with its polar counterpart, a circular bar plot, on the right. In both visualizations 36 bars are used to show the frequency distribution of the corresponding 36 bins of the circular data.

Radial Methods for Time-Oriented Data

Time-oriented data and data sets with a temporal coherence are ubiquitous in many application domains of visual data analysis. Shneiderman's Task by Data Type Taxonomy [87] identified temporal data as an individual data type which, therefore, needs special handling. Aigner et al. [1] reviewed existing visualization, analyzing and abstraction methods for time-oriented data and emphasized that visualization frameworks should consider time as a special data dimension. They showed that different types and notions of this data dimension exist across different fields. They name and discuss three criteria for designing a time-oriented visualization system:

- Linear time versus cyclic time
- Time points versus time intervals
- Ordered time versus branching time versus time with multiple perspectives

Regarding the first criterion, Aigner et al. [1] pointed out that many natural processes are cyclic, for example, due to reoccurring seasonal climatic changes. They discovered that the linear versus cyclic time characteristic plays a crucial role when it comes down to the expressiveness of a visualization. Due to this fact and the fact that this section focuses on cyclic and circular data dimensions, visualizations that just present data with respect to a very simple linear time axis will not be considered.

Presenting time-series data in a radial visualization can help the user to detect previously unknown periodic patterns or trends [12, 94, 99–101]. To give an example, if the value plotted

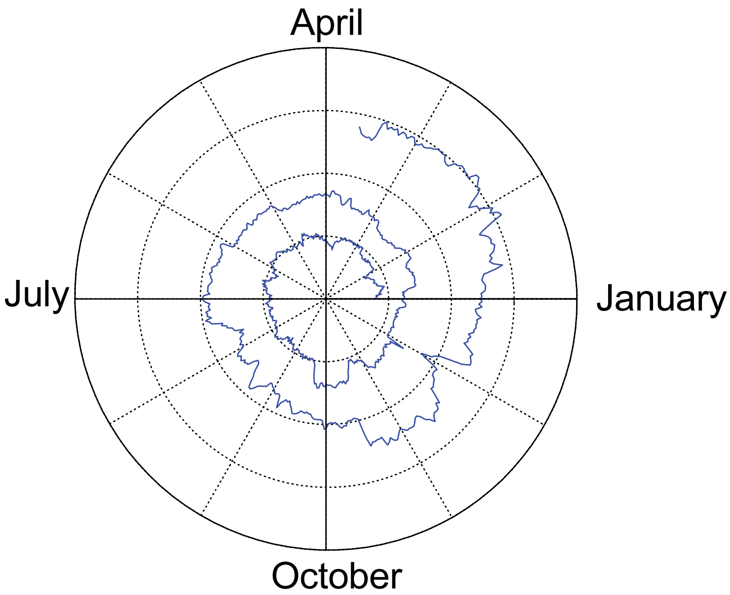


Figure 2.14: A polar plot showing a time series, i.e., the development of prices, of stock market data. Reprint from the book *The Grammar of Graphics* by Wilkinson et al. [101].

on the radial dimension is (monotonically) increasing over time (which is plotted cyclic on the angular dimension by discarding the year information), the final line will appear in the form of a spiral as can be seen in Figure 2.14. Such a polar plot can reveal seasonal patterns that would be difficult to spot in a conventional rectangular time-series plot. Carlis and Konstan [12] ensure a spiral shape by anchoring data to events, i.e., aggregating data over a time interval, on an Archimedes-spiral time axis. Their system is able to highlight one or multiple periodic attributes along the radii of the visualization as shown by Figure 2.15. The mentioned system allows the user to change the temporal range of one lap of the spiral, facilitating exploration if the periodicity is not known a priori. Animating through the length of a lap helps the user to look for lap values where data lines up. Carlis and Konstan pointed out that such a visualization system should be integrated with other data exploration tools, as it is not suitable for all forms of data exploration.

The SpiraClock [23] system is an analog-clock shaped interactive visualization for exploring upcoming events. Inside an analog-clock coordinate-frame a spiral is placed that represents the future starting at the analog clock minute hand. An event is visualized by a sector inside the spiral representing its time interval. As time progresses, the spiral unwinds and sectors drift radially outwards and finally fade out when the event has ended. This continuous unwinding, as illustrated by Figure 2.16, can also be controlled interactively by the user by temporarily dragging the minute hand into the near future. In contrast to previously mentioned methods, this visualization uses a fixed lap cycle of twelve hours. This makes the visualization easy to understand and according to Dragicevic and Huot [23] the spiral display requires only minimal additional explanation to the user.

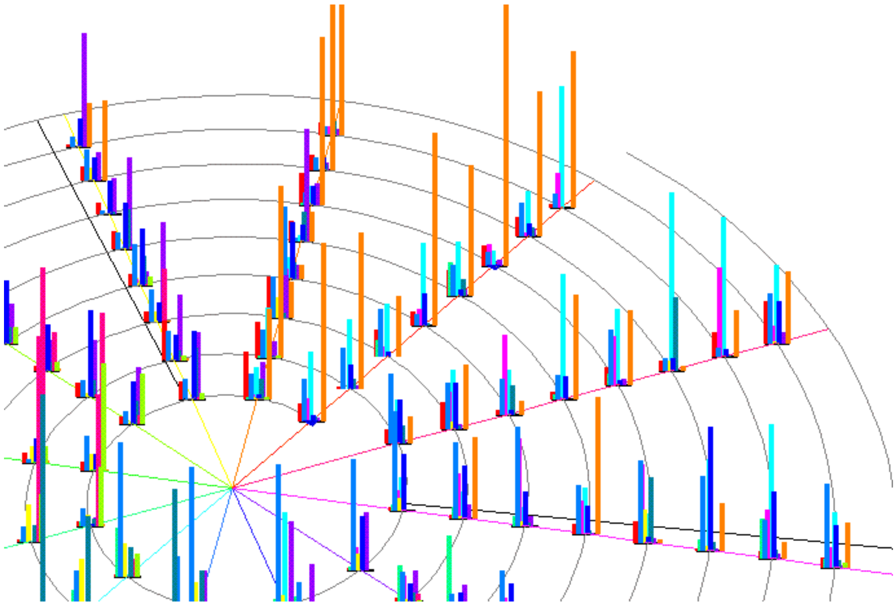


Figure 2.15: An Archimedes spiral is used to depict the time axis. Bars are used to visualize the values of multiple periodic attributes along the spiral. Reprint from Carlis and Konstan [12].

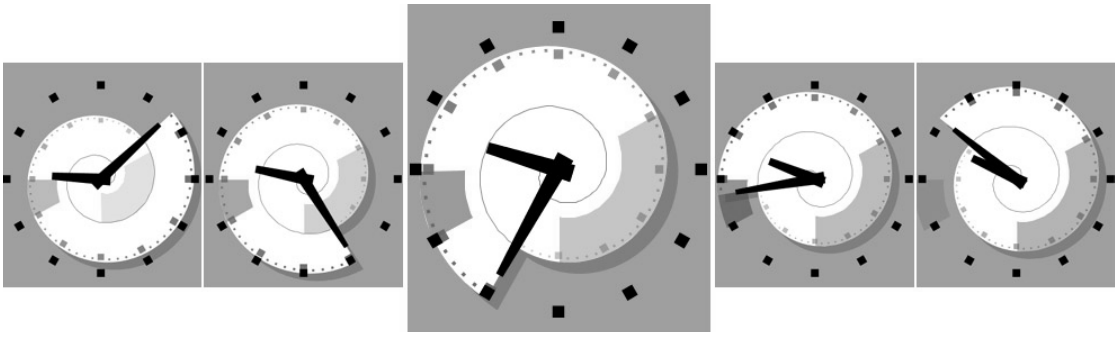


Figure 2.16: The SpiraClock system uses a spiral design to visualize upcoming events. The spiral unwinds outwards as the time shown on the clock interface progresses. Reprint from Dragicevic and Huot [23].

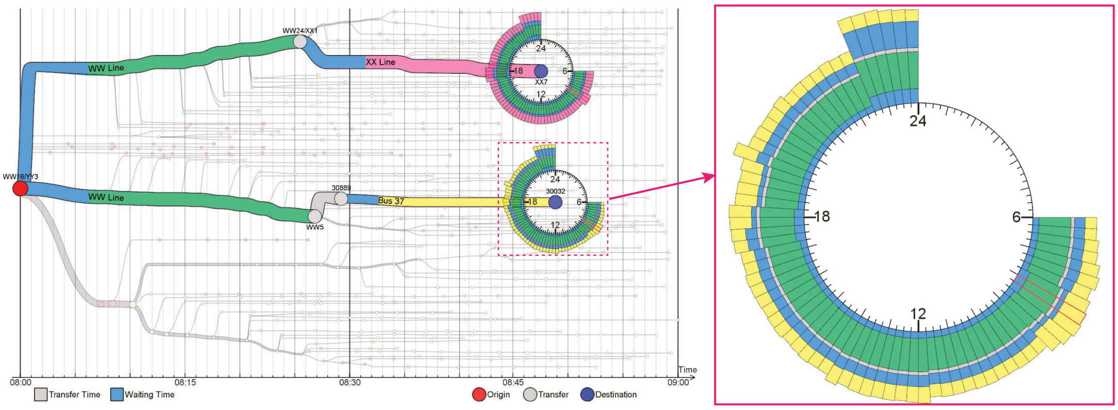


Figure 2.17: The mobility wheel visual cue uses an analog 24-hours watchface to display temporal variation of mobility-related factors (right) in a public transportation network (left). Round-the-clock patterns can thus easily be detected. Reprint from Zeng et al. [105].

Tominski and Schumann [94] further enhanced such spiral-based displays by applying two-tone pseudo coloring [84]. They further added interactive manipulation mechanisms for browsing through time, and easy switching from an overview to a detailed representation of the data.

Zeng et al. [105] designed a special visual cue to help users with the analysis of temporal variations in public transportation networks. Their proposed radial visualization, i.e., the *mobility wheel*, as depicted in Figure 2.17, uses an analog 24-hours watchface as the visualization's foundation. Binned and aggregated mobility-related temporal data is visualized via stacked bars on the outside of the watchface's circumference. This visualization technique enables users to quickly detect round-the-clock patterns of multiple mobility-related factors within a single visualization.

Radial Methods for Directional Data

Directional data naturally classifies as circular data. Thus, radial methods are often used to analyze and present directional data such as data describing air/liquid flows, or dominant directions.

Hussain et al. [42] used wind data from five sensor locations collected by a meteorological data service to analyze climatic changes in Malaysia. In their work on analyzing the data, they concluded that "A useful graphical representation of the circular histogram is the *Rose diagram*, in which the bars of a circular histogram are replaced by sectors" [42, 66]. This type of graphic, also called *wind rose* or *"rosa ventorum"* in Latin, was used by cartographers on navigational charts in the thirteenth century for displaying estimates of oceanic wind strengths and directions [101]. Hence, this visualization shows aggregated circular frequencies of directional data.

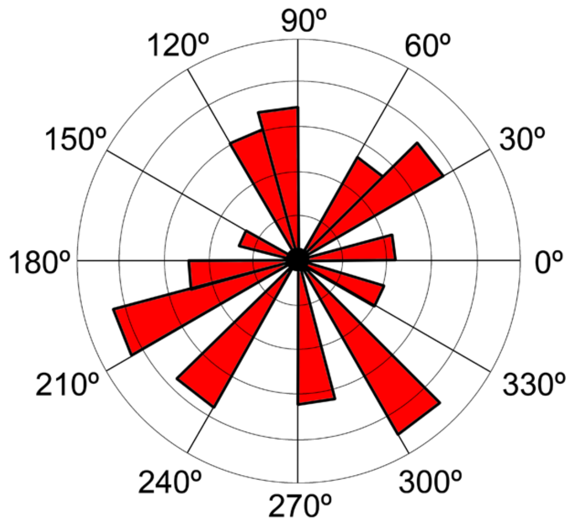


Figure 2.18: A rose diagram showing the frequency distribution of directional data. Image adapted from Hedrick et al. [38].

Analog to the bar chart, categories inside a sector can be presented by stacking. This results in a *stacked rose diagram* as shown in Figure 2.19. Kirchner-Bossi et al. [59] worked on the reconstruction of daily surface wind speeds in time periods when no actual wind data was recorded. Their algorithms were tested against a ground truth of data recorded at six weather stations at wind farms across Spain. Stacked rose diagrams were used to demonstrate the results of their algorithms in a side-by-side display.

Although rose diagrams, as shown in Figure 2.18, give a better visualization on circular data compared to Cartesian plots, a simple rose diagram is not able to reveal correlation over time between different locations. For this purpose, Hussain et al. [41, 42] used *spoke plots* to represent the correlation of dominating wind directions between two distinct locations. The spoke plot consists of two concentric rings where each ring represents a location. As can be seen in Figure 2.20, points lying on the a circle represent the dominating wind direction on a specific day at that location. Data points of the same day are connected with a line resulting in the characteristic *spoke*. This way users are able to easily spot positive, i.e., spokes not crossing the inner circle, or negative correlations, i.e., spokes crossing the inner circle. By this definition, however, the correlation sign depends on the ratio between the radius of the inner and the outer circle. The approach is limited to the visualization of directional correlation, i.e., the wind's direction, but lacks the ability for users to inspect the correlation of a second dimension, e.g., the wind's speed.

Visualizations for directional data that also contain (geo-)spatial information are found in the field of Flow Visualization. A common technique there is to use glyphs, often in the form of arrows, to visualize dominating flow directions and magnitudes. Figure 2.21 shows such a flow visualization using this technique. Although this representation is actually not radial it is

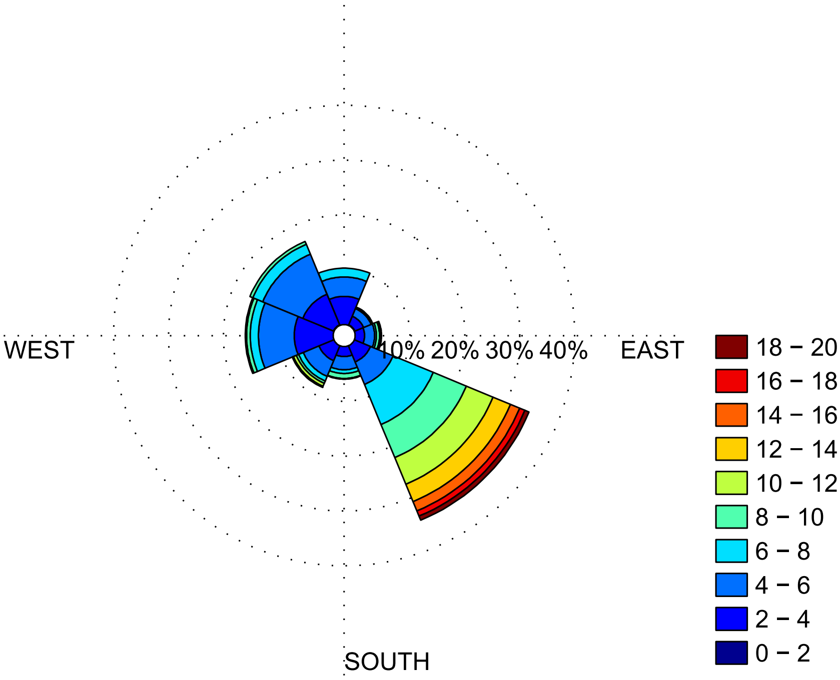


Figure 2.19: Stacked rose diagram showing the frequency distribution of wind directions and speeds. Stacking and coloring is used to additionally visualize the frequencies of wind speeds inside sectors. Reprint from Kirchner-Bossi et al. [59].

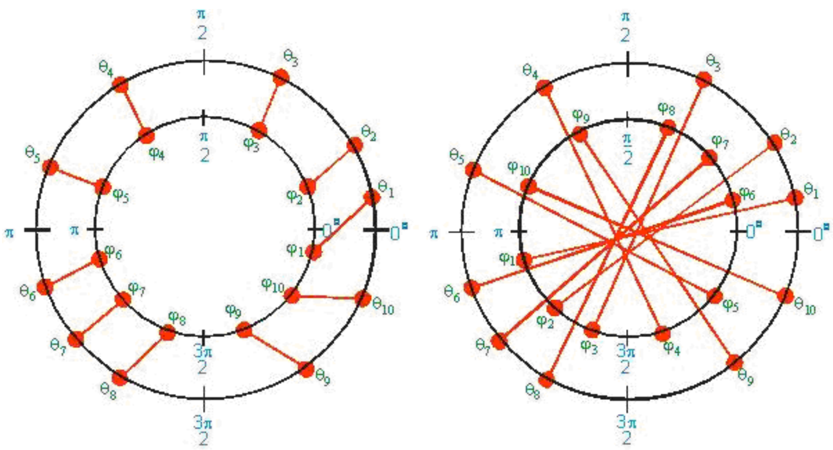


Figure 2.20: Two spoke plots showing strong correlation (left) and weak correlation (right) of dominant wind directions between two sensor locations over time. Reprint from Hussain et al. [42].

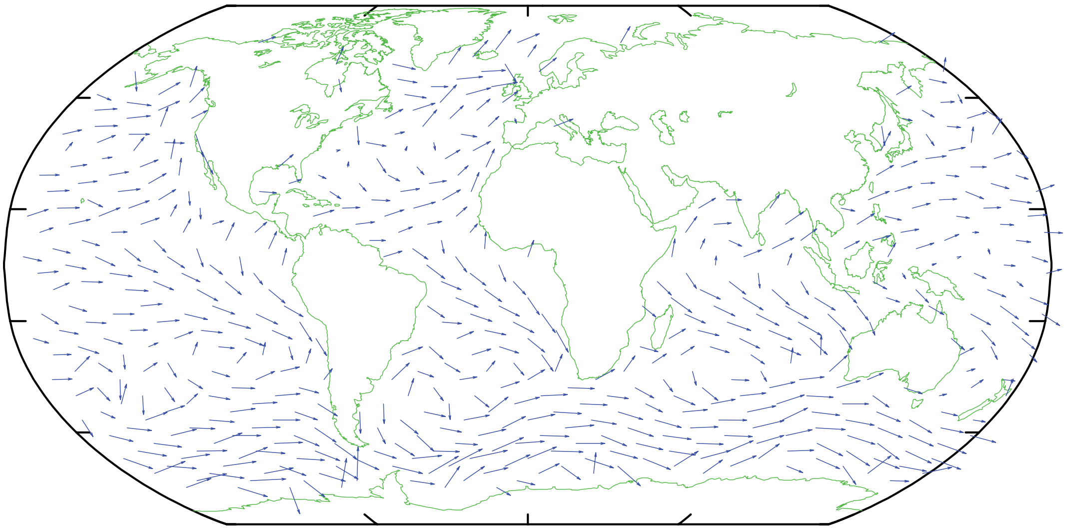


Figure 2.21: A visualization of global oceanic wind directions and speeds using glyphs. Each arrow encodes the wind direction by its orientation and wind strength by its length for a particular region. Reprint from the book *The Grammar of Graphics* by Wilkinson et al. [101].

worth mentioning as it is often used to display directional data within another plot. Hussain et al. [42], for example, developed an alternative wind diagram, which focuses on displaying the dominant wind direction and speed over time allowing the user to detect patterns. The average wind speed is plotted as a line as a function over time. Glyphs are used to show the dominant wind direction and speed of gusts at discrete moments in time. Figure 2.22 shows a recording of 21 days revealing a reoccurring pattern.

Wilkinson et al. [101] used a polar plot to visualize the barometric pressure by wind speed by wind direction. Instead of plotting the actual values for each direction and speed, they smoothed the data over the surface given by wind direction and speed. Finally, they rendered iso-lines of barometric pressures and color-coded them according to their magnitude of pressure, as shown in Figure 2.23.

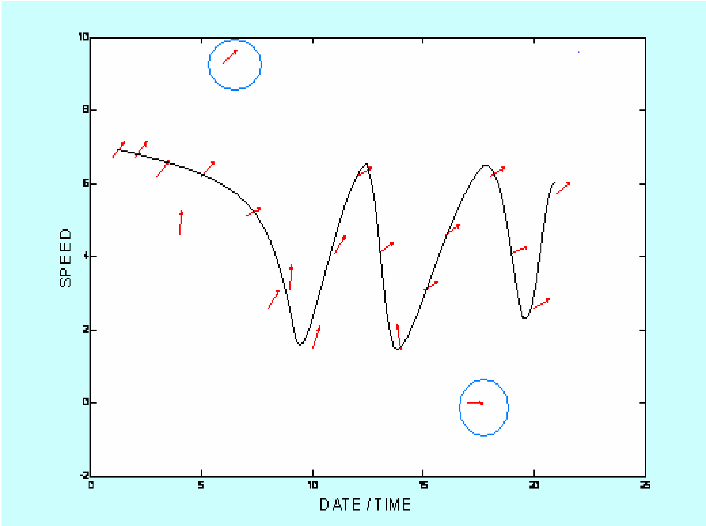


Figure 2.22: An alternative diagrammatic representation for wind directions, speeds, and gusts showing a recording of 21 days. The plotted line shows the average wind speed over time. Red arrows indicate the direction and speed of gusts within a certain time frame. Reprint from Hussain et al. [42].

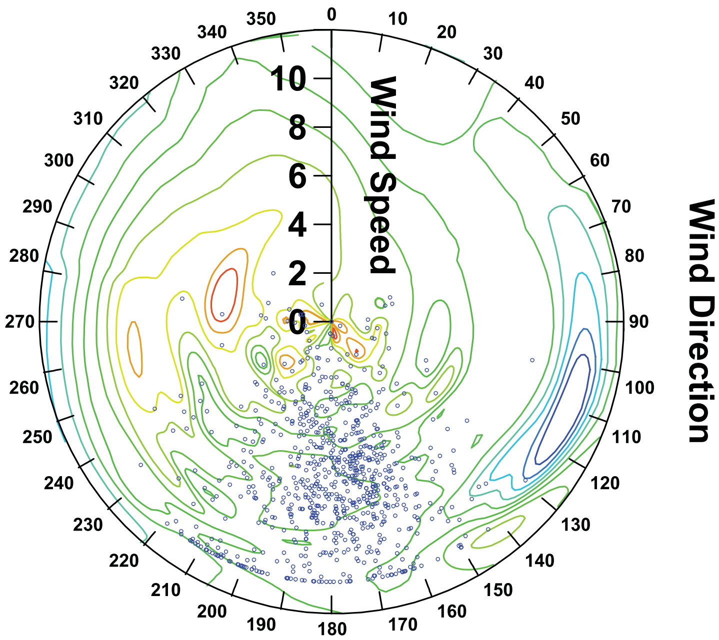


Figure 2.23: A polar plot showing barometric pressure contours (iso lines) by wind speed by direction. Blue rings indicate the highest gusts recorded each hour. Reprint from the book *The Grammar of Graphics* by Wilkinson et al. [101].

CHAPTER 3

Problem Characterization and Abstraction

This chapter focuses on characterizing the existing problem in the domain of analyzing data in the wind energy production sector. An abstraction of this problem to more generic data types is presented and discussed. Due to this data abstraction, a much wider range of application domains is covered by the approaches proposed within this thesis. Finally, common pitfalls of the abstracted problem are presented.

3.1 Problem Characterization

Data analysts in the energy sector are confronted with huge data sets, which incorporate a wide range of data sources and thus contain a multitude of very different data types. These data sets may, for instance, be composed of temporal annotated recordings from different sensors, temporal events, categorical data, or the outputs of computer simulations. In addition, experts in the field of wind energy production are particularly interested in the analysis of weather data, as the weather steers the power output of a wind turbine and thus the total power output of a grid of turbines or a wind park. Understanding the weather and its influences ensures better forecasts, which are an essential part for cost-efficient planning of larger scale energy production. The weather data typically includes recordings as well as forecasts of temperatures, wind directions, wind speeds, and wind gusts for different geographical locations. Experts address the challenges listed in Section 1.1 by performing a visual analysis of this data.

However, using traditional non-radial methods for this visual analysis is problematic if the data consists of directional and/or temporal data. Directional wind data, due to the underlying nature of this data, is intrinsically a circular data type. Furthermore, temporal data such as time stamps may not be only treated as linear data but also as a cyclic data type in order to uncover periodic patterns. Whereas temporal data, when treated as a simple linear time dimension, is usually well supported by common enterprise data analysis software suites, circular or cyclic data types are often mishandled or not supported at all. As circular and cyclic data requires specialized approaches, a multi-modal data analysis with visualization techniques appropriate to each data type and user task is required when analyzing such data. The problem is thus characterized by the fact that non-radial methods are not well suited for the analysis of wind energy production data. The subsequent sections discuss how this problem can be abstracted to more application domains and demonstrate the pitfalls of working with circular data.

3.2 Problem Abstraction

Circular and cyclic data can be found in many domains besides the wind energy sector previously mentioned. Wind data is also utilized in the research fields of meteorology, aerodynamics, and aerospace engineering. In acoustics the measurement of sound propagation from a single sound source involves recording directional data which, in this case, is circular data. Any process with cyclic or periodic behavior produces cyclic data. Application domains, which incorporate cyclic data, are numerous and ubiquitous. For example, the task of optimizing internal combustion engines for the automotive or shipbuilding industry involves the recording and analysis of periodic events as the process of fuel and air combustion drives the rotation of the engine, i.e., cycling of the crankshaft. All these domains must use appropriate solutions, which respect the special properties of circular and cyclic data. These special properties are discussed below.

Natural Coherence in Circular and Cyclic Data

Circular and cyclic data inherits a *natural coherence*, i.e., a real-world relationship or correlation between values, from their respective data sources. This coherence usually originates from a natural process involved in the recording (or simulation and generation) of the data. Such a coherence may be of a *temporal origin* such as data that incorporates the seasons of the year, months of the year, hours of the day, or other periodic events of any kind. For instance, there exists a relationship for successive and opposing months within a year, e.g., the month February is always followed by the month March, and the month August is the “opposing” month of February.

Furthermore, a *spatial coherence* does exist in data that incorporates values such as two-dimensional directions, angles, or other geographical information. For example, the direction representing “North” is spatially opposite of the direction encoding “South”, whereas “East” is opposite of “West” and so on.

However, depending on the concrete type(s) of the underlying data and the chosen visualization technique, this natural spatial or temporal coherence may be hidden, i.e., not encoded visually, and thus obscured in the final visualization. Furthermore, the information may be rendered distorted, leading to wrong assumptions about the visualized data values. Any linear visualiza-

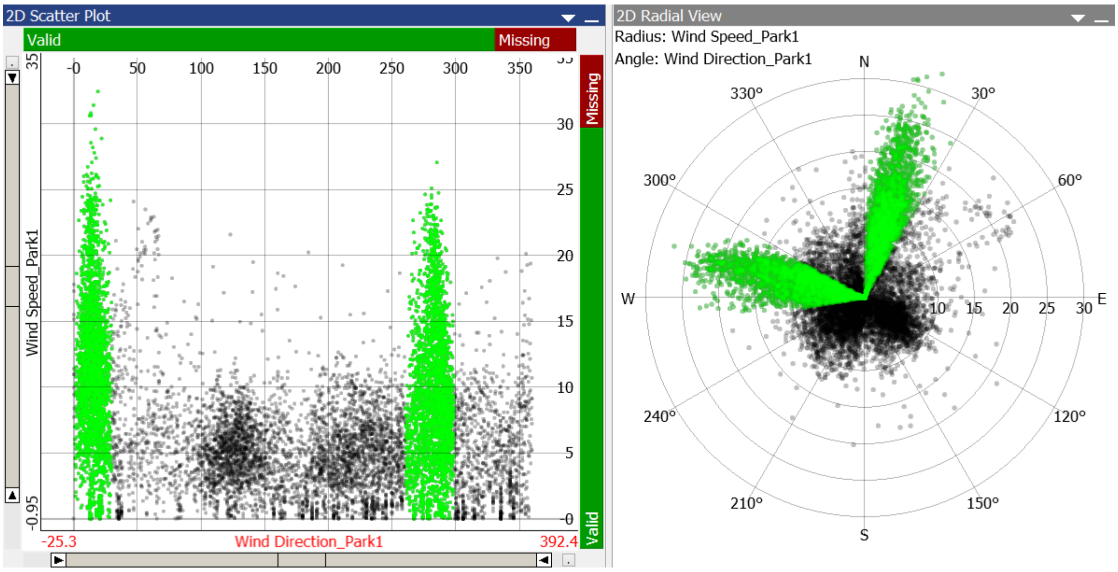


Figure 3.1: An Euclidean two-dimensional scatter plot on the left and a radial scatter plot on the right displaying the exact same circular data. The spatial relationship is not retained in the linear technique on the left. Hence, statements about the directional relation of the selected points (highlighted in green) are hard to make. In the radial design on the right, however, it is much easier to determine the spatial relationship of these points.

tion technique applied to circular or cyclic data, like plotting such data in an Euclidean-space scatter-plot, will mask those previously discussed spatial or temporal relationships. Figure 3.1 depicts how spatial relationship is not retained if the wrong visualization design is used. Hence, statements about the directional relationship of selected points (highlighted in green) are hard to make in the plot on the left. In the radial design on the right however, it is much easier to determine that the most dominant direction is exactly orthogonal to the second most dominant direction.

The loss of this implicitly contained information may prevent data analysts from carrying out an efficient data analysis. For this reason, special treatment of such data types is required to prevent this loss of information. Hence, it is a goal to maintain and display any natural coherence present in the data in the final visualization. This involves many different tasks such as choosing an appropriate interaction and visualization technique suitable for the analysis task and/or the desired presentation.

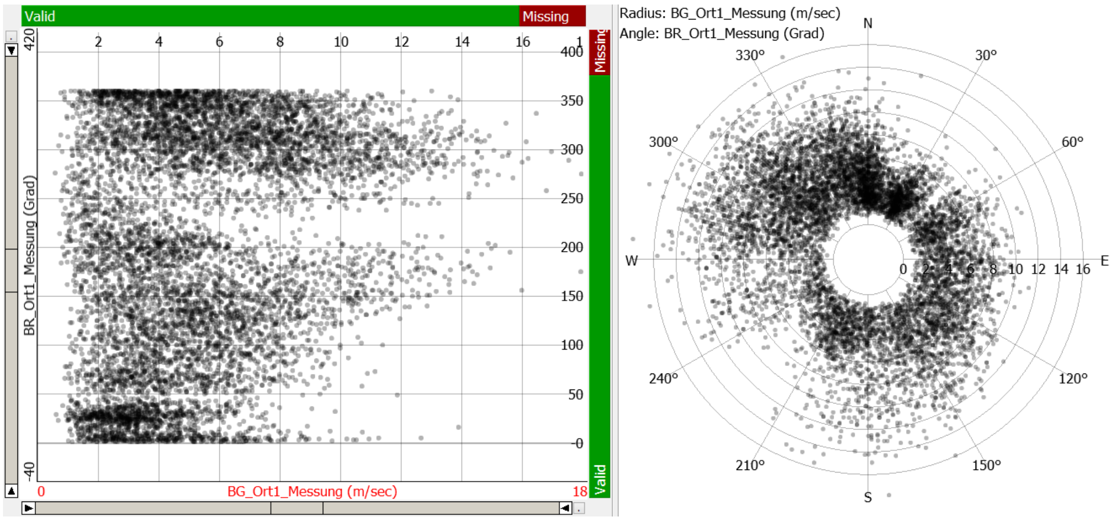


Figure 3.2: Two scatter plots displaying the exact same two data dimensions, once using the Cartesian coordinate system (left), and once the polar coordinate system (right). In the Cartesian plot the circular data dimension, i.e., the one assigned to the y-axis, can only be displayed with a jump in continuity, whereas it is correctly visualized as a continuous data dimension in the polar plot.

Circular Closure of Circular Data

As already discussed in Section 2.5, circular data maps (and correlates respectively) by its definition to the circumference of a circle. As the circumference of a circle represents a closed space, the concept of an origin, a start or an end, is usually arbitrary or undefined [42].

This *circular closure* of the data space requires designers, system builders, and end-users alike to ensure special handling when dealing with circular data. The following subsections discuss the pitfalls and problems that may occur due to the circular closure characteristic of a circular data type. These challenges are addressed in Chapter 4 on the visualization and interaction design of a novel software component. Furthermore, Section 5.3 discusses the concrete implementation of common visual analysis tasks that involve this circular closure property.

3.3 Common Pitfalls

Circular data requires specially designed approaches if the task is to analyze and/or present this circular information. System builders, designers, and data analysts should avoid the typical pitfalls when working with this data type, which are discussed below.

Visualizing Circular Data

Traditional visualization techniques may not be suitable, i.e., they depend on a user focus and task, for this kind of data. For instance, it is due to the fundamental property of circular data that a circular data dimension cannot be displayed in the widely-used Cartesian coordinate system without a jump in continuity. The visualization frame of a Cartesian coordinate system can be set to exactly cover the range of the circular data dimension, i.e., the exact range of values that map to $[0,\ 2\pi)$ radian on the circumference. However, a jump in continuity from the end to the start of the plotted circular dimension does always occur. Using a linear visualization technique for this kind of data makes it hard to observe events, track features, or detect patterns, which are located at (or involve) the jump in continuity. This in turn may prevent data analysts and decision-makers from gaining insight, or even worse, drawing the right conclusions. Hence, it is necessary to avoid this discontinuity when working with circular data.

Most radial diagrams use, although in many different approaches as demonstrated in Section 2.4, the polar coordinate system. This two-dimensional coordinate system supports the display of one circular data dimension, i.e., the angular dimension of a polar plot, without a jump in continuity. Hence making it suitable for rendering data that incorporates circular data. Figure 3.2 demonstrates the circular closure of the angular data dimension in a polar plot and shows a direct comparison with a Cartesian coordinate system plot rendering the same data.

Executing Operations on Circular Data

Algorithms, mathematical operations, or predefined tasks present in many software suites expect a linear data space and hence may produce incorrect, ambiguous, or unexpected results if applied to circular data due to the circular closure characteristic. Even calculating a very simple metric, such as the difference between two angles, for example 350 and 10 degrees, has to account for the circular closure to produce the result of 20 degrees (the short distance on the circumference). Moreover, this result is ambiguous as minus 340 degrees (the long distance on the circumference) might also be a valid/desired result. Therefore, any data analysis algorithm or mathematical operation that is applied to circular data has to adapt for the circular closure property. Data analysts have to be very cautious if the desired algorithm or operation is correctly applicable or has been designed for circular data.

Defining Intervals on Circular Data

Defining value-based intervals on one or more data dimensions is a common task in data analysis. Intervals, i.e., value ranges, are very versatile as they can be used to select or filter data, define categories, or present results of some operations to the user, thus making them a valuable tool. However, due to the previously described circular closure, defining a common one-dimensional interval on a circular data dimension just by a start and end value results in an ambiguous description. For example, the user may define an interval with a start value of $15\,^\circ$

and chooses 350° as its end value. If written in the traditionally-ordered, i.e., monotonically-increasing, way, the interval will look like $[15^\circ, 350^\circ)$. However, with this being the only information, it remains unclear if the user wants to describe all data values ranging from 15 to 350 degrees or from 350 degrees to 15 degrees, thus include the discontinuity at the polar axis in the range.

CHAPTER 4

Visualization and Interaction Design of the 2D Radial View

An important contribution of this thesis is the design of a new *View*, i.e., the *2D Radial View*, which is able to conveniently enrich any existing visual data analysis system. Figure 4.1 shows the *2D Radial View* integrated into the Visplore system. The design proposed in this chapter is driven by the intention of supporting data analysts in their work with wind energy production data and the challenges that lie within this application domain (see Section 1.1). The design's overall goal is to provide users with possibilities to better analyze this special kind of data. However, the presented approach in this chapter is not limited to wind energy production data. It can be applied to more generic data and user tasks, i.e., any application domain that incorporates circular data, as already discussed in Chapter 3.

This chapter is split into three major parts: The first part gives a motivation by an analysis of user tasks and specifying important design requirements that have been determined by this analysis. The second part discusses design choices, which lead to all the different visualizations the final version of the *2D Radial View* offers. These different types of visualizations are illustrated along the discourse. The third part focuses on design decisions regarding all active but also passive interactions, which have been made available to the user by extending an existing visual analysis framework by the proposed *2D Radial View*.

4.1 Motivation

The design of the 2D Radial View is driven by the need for a better data analysis solution requested by a national power grid operator. The goal is to design a new visualization technique, which offers an advantage, e.g., a benefit in visual perception or user acceptance, in the target domain over already existing solutions. Finding the suitable design for this visualization

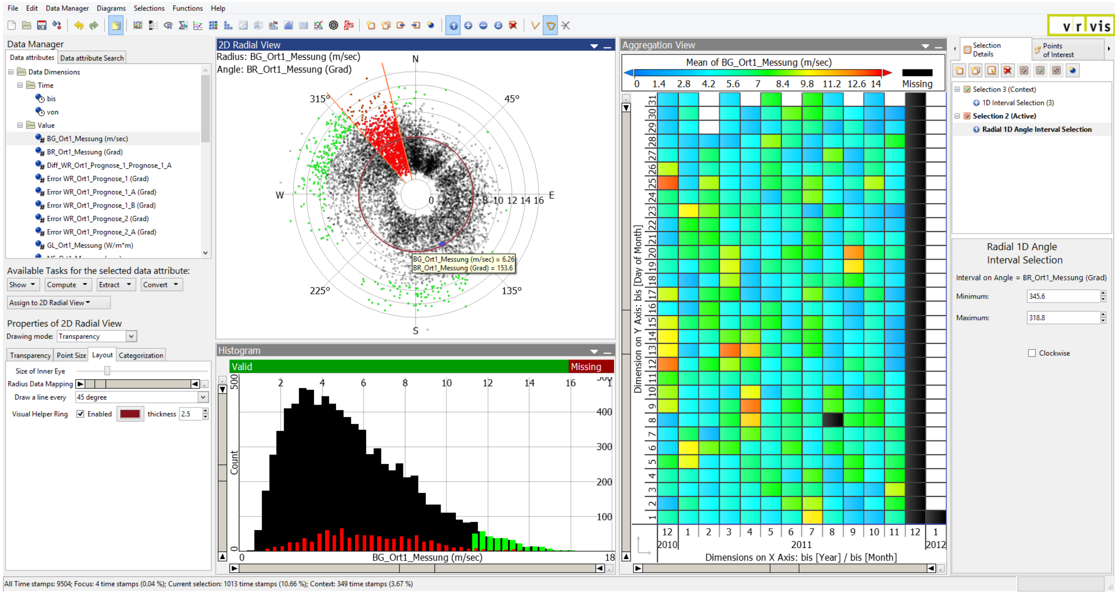


Figure 4.1: The screenshot shows the proposed 2D Radial View integrated into the visual analysis software Visplore. The novel radial view (top left) enriches the existing system by providing new techniques for the visual analysis of circular data.

technique is one of the main contributions of this thesis. As already discussed in Chapter 3 of this thesis, a big challenge in the energy sector, especially when wind energy production data is involved, is the appropriate visualization and thus efficient and correct analysis of data that incorporates circular components. Motivated by this need in the wind energy production sector, a requirements analysis was performed. The resulting design requirements and the shortcomings of solutions that are currently used by data analysts are discussed below.

Design Requirements

Based on communication with the targeted user group, i.e., data analysis experts in the wind energy sector, the following design requirements have been identified:

The design has to support the visual analysis process (see Section 1.3). In particular, it must enable users to discover unknown behavior of complex systems or to prove/disprove hypotheses about their data. Therefore, the design has to provide data interaction techniques. Interactions and data changes must be reflected as fast as possible in the visualization.

Due to the problems discussed in Chapter 3, circular data must be supported as a special data type. Not only the visualization but also the interaction design has to support the analysis of a circularly-closed data space such as presented by the data of wind directions.

Recorded wind directions, i.e., numerical values, relate to directions in the real world. Thus, a visualization design appropriate to this real-world directional relation has to be used.

Users want to visualize a circular data dimension (e.g., the wind direction) with respect to another non-circular data dimension (e.g., the wind speed). Furthermore, relationships among disparate entities must be made visible to the user. The visualization design thus has to provide this functionality. Possible interactions within the 2D Radial View have to adapt to the visualized data space.

Users want to be able to easily compare, i.e., numerically and also visually, (wind) data recorded at two or more different locations. Furthermore, the analysis of recorded data versus prognosticated data for a specific location must be supported by the design. Hence, visualizing and analyzing such correlations has to be included in the design.

Any used visualization and interaction technique must support big data sets. The design must be able to handle up to hundred thousands of data values per dimension. Users want to identify trends and differences even in extremely large data sets, which contain detailed recordings of many successive years. Common scatter plot representations are not sufficient as they lack abstraction and cause visual clutter. Thus, visualization techniques appropriate to such a high number of data points must be used.

Introducing new software into an existing company infrastructure usually encompasses at least testing, licensing, deployment, and special user training. To keep integration costs to a minimum, the new solution has to integrate seamlessly into an existing tool and support the company's existing workflow. As domain experts often use a modular software suite, the novel solution has to be designed as an extension to such a system.

An important task is the presentation of the results of a visual data analysis to decision-makers. This requires visualizations to be expressive as they must transport the insight, i.e., the most essential information, to the viewer. At best, every final visualization is self-explanatory, hence no additional explanation is needed. To achieve this goal, the proposed techniques must not deviate too far from user-familiar techniques.

Shortcomings of Existing Approaches

Already existing approaches do not fulfill the above stated design requirements. They fail some or all of the requirements for the following reasons:

General purpose software suites often use a visualization layout domain experts are not familiar with and where the relation to the real-world application is not maintained. For example, polar plots often render angles of zero degrees on the positive x-axis and handle increasing angle values in a counter-clockwise manner. However, this standard mathematical projection does not match today's representation/mapping of real-world directions where we expect zero degrees to be mapped in the north and 90 degrees to be mapped to the east.

Traditional visualizations are only static, inflexible in their visualization parameters and thus not very versatile. Existing interactive approaches are not integrated into a linked multi-view data analysis framework and thus provide only limited data analysis capabilities. It is also for these reasons that these systems do not support an effective, e.g., direct and interactive, comparison of multiple sets of data dimensions.

Existing solutions are often built upon a complicated workflow, which involves using multiple tools in order to accomplish the designated task. Refinements and updates, which are essential in the visual analysis process, are therefore tedious, slow, and prone to errors.

Many software systems, frameworks, or utilities do not support the circular data type. Hence, they are not suited for the visual analysis of data common to the wind energy production sector.

4.2 Visualization Design

This section discusses design choices regarding the chosen representation for the 2D Radial View. The used visual encoding is presented and explained in the discourse.

Design Scheme and Pattern

In order to help data analysts to efficiently analyze circular data, the 2D Radial View uses a radial layout for all of its different visualizations. As already discussed in Section 2.4, there exits a variety of possible design patterns for radial visualizations. Choosing an appropriate design pattern is crucial as it is the first step in encoding data into a visualization. The viewer, in return, tries to visually decode information from this encoding. Cleveland and McGill [16] point out that a good encoding is vital as it results in a more accurate data decoding. An accurate decoding is reflected in a "better chance to detect and understand properly the patterns and behavior of the data".

The proposed 2D Radial View implements the design scheme of a polar plot and its layout matches the definition of the star design pattern. This design choice has been made on the account of the following observations: Firstly, the circular nature of wind energy production data suggests to use a radial layout for the visualization. As Wilkinson et al. [101] point out, directions and astronomical time are "intrinsically polar" and therefore it is evident to use a polar arrangement for these data types. Such a polar arrangement makes also sense for temporal data because it allows users to examine relations across the circular boundary, i.e., the annual or daily boundary. Secondly, polar plots are a good choice to visualize relationships among disparate entities and thus well-suited for the targeted application domain. Thirdly, communication with domain experts has revealed the desire to implement an already familiar visualization layout, i.e., a layout they are already used to and thus understand. Aggregates of wind directions are often communicated through a Rose diagram, which uses the layout of a polar plot. Thus, the targeted user group is already familiar with the concept of a polar plot.

General View Properties

Although the 2D Radial View implements different visualization techniques, some general properties regarding the layout and organization hold true for all of the different visualizations proposed in this chapter:

All visualizations of the 2D Radial View are based on the layout of a two-dimensional polar plot. By using the polar coordinate system, the layout offers the freedom that every point within the resulting radial display space is addressable by a polar coordinate and thus can be used to present information. By this means, visual elements are not constrained to any predefined structure such as, for example, the path of a spiral.

By being based on a two-dimensional polar plot, the 2D Radial View visualizes relationships of entities among two data dimensions. Therefore, data must be assigned to each of its two polar dimensions, i.e., its angular and its radial dimension. A successful data assignment results in the instant rendering of a visualization. Changes to the data assignment are immediately reflected in a new rendering of the visualization.

The view's angular dimension is designed to represent numerical data that encodes directions. With the focus on the most common task of domain experts, i.e., using directional data as angles of the visualization, this dimension expects numerical data in the range of $[0, 360)$, which is interpreted as an encoding of directions ranging from zero to 360 degrees. This mapping can intentionally not be changed by the user for better ease of use. Furthermore, this mapping is a linear projection making the displayed directional data free of distortions and preserves the natural coherence. Any circular data represented in a different encoding scheme has to be modified to match the view's accepted range prior to assignment. This restriction ensures that no accidental misinterpretation of circular data can occur due to a fuzzy data source assignment performed by the user or system.

The view's radial dimension accepts numerical data containing values of any numerical range. It thus is not bound to a specific purpose making it versatile to many applications. A mapping function defines, which value range of the assigned data is mapped, and in what way, to the view's radial dimension. As the radial dimension is not constrained to the display of a fixed range of numerical values, choosing a suitable initial range to display is non-trivial. The 2D Radial View's approach to solve this is as follows: Upon the assignment of data to the radial dimension, the implementation finds the minimum and maximum values, i.e., the numerical limits, of this data. These detected limits are then used to setup the radial dimension's data mapping to cover the complete data range in the radial visualization. In this way, the user initially is shown a representation of all of the available information in this data dimension. In contrast to the angular data dimension, the user is encouraged to change this mapping, and thus the displayed information, at any time during the visual analysis process. This functionality is exposed to the user through a graphical user-interface element, i.e., a data mapping widget. Changes to the widget are immediately processed by the system and interactively reflected in the corresponding visualization.

Every visualization of the 2D Radial View is designed to always effectively use the maximum radial display space that is available within the rectangular viewport. This property is achieved by setting the diameter of the polar plot's coordinate frame to match the short edge of the viewport rectangle and placing the origin of the coordinate system into its center. Any event that results in a resizing of the viewport triggers an immediate redraw of the visualization. After such a redraw, it again perfectly matches the new size. While this sacrifices some of the available display space, the approach eliminates the need for the user to interact with the view's layout in order to keep all features of the visualization visible.

Center Design

The placement, the role, and the look of the visualization's center play a crucial role in the design of a radial visualization. In a radial visualization, the center implicitly provides the user with some visual context as it gets assigned a specific role [21]. It serves as a visual and/or semantic reference. The 2D Radial View assigns the center the origin of the underlying polar coordinate system and by this serves as the center of all of its radial projections. As mentioned above, this origin is always placed in the exact center of the viewport.

The targeted display-device family, i.e., common office liquid-crystal displays, offers only an evenly-distributed pixel density and by this a constant resolution across the device's entire display surface. As a result of this, any radial design faces the following problem when displayed on such a device: The available resolution for displaying visual features is extremely low in the center but increases rapidly outwards by the factor of the square of the radius. Asides from the available resolution, the same quadratic relation applies to the available absolute space for displaying entities. Due to these circumstances, placing visual features at or close to the center is problematic as they may heavily overlap, suffer from aliasing artifacts, or be not recognized as individual features. Hence, important details near the center in a polar plot may get lost. On the other hand, plots based on polar coordinates can be used to enhance detail in areas where it is important. For example, placing the root node of a hierarchical tree in a polar plot results in a lot of space for showing leaves or terminal nodes [101]. However, the 2D Radial View's application domain is not such a visualization of hierarchies. Therefore, the design of the view's center requires special attention and is discussed below.

The problems mentioned above encompass perception issues as well as the technical challenge of avoiding frequency aliasing. In order to address these problems, different approaches are applicable: The first option is not to show the information that maps very close or to the

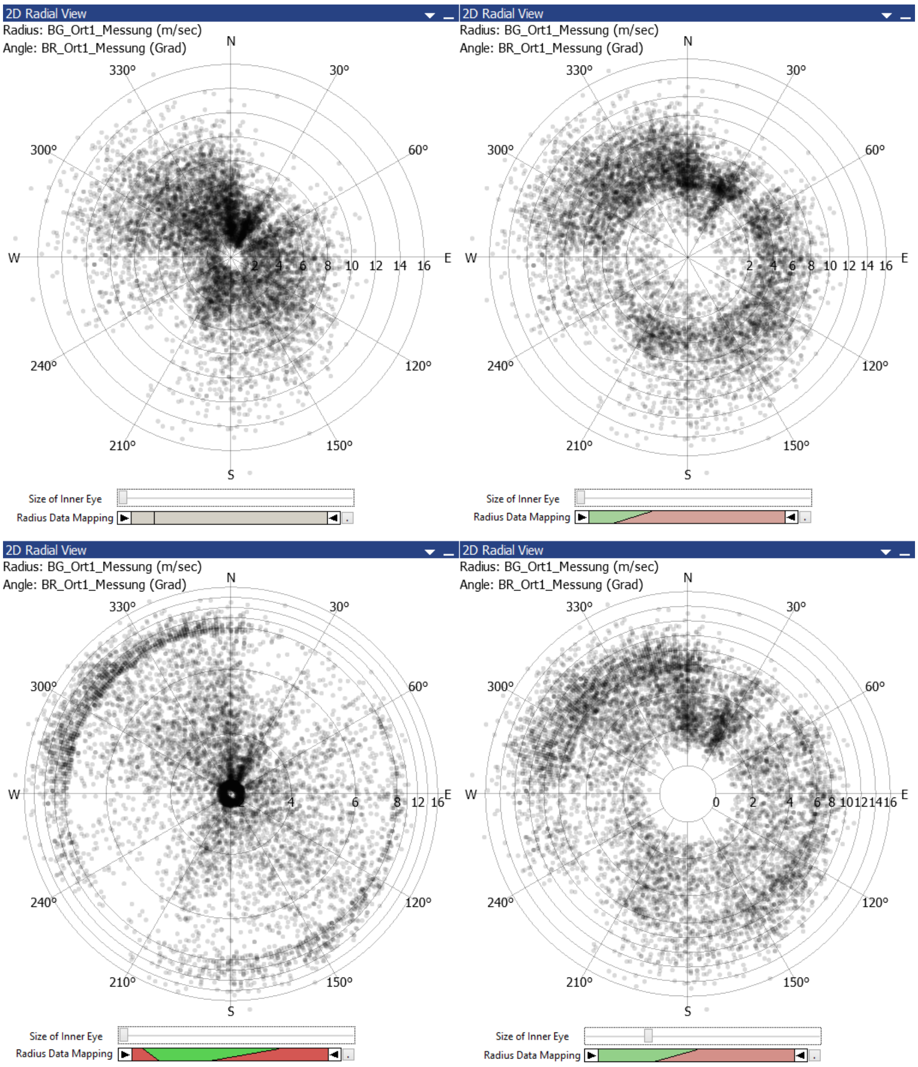


Figure 4.2: Without any radial distortion, points close to the center are hard to distinguish and suffer overdrawing (top left). A data mapping widget (depicted below each plot) enables users to adjust the mapping of the radial data dimension. The resulting distortion of the polar projection enables the user to focus on a particular region of the mapped numerical range (top right and bottom left). Adding a void in the view's center (bottom right) greatly improves the radial visualization as this counteracts the polar plot's non-linear projection properties.

center. Obviously, this option is not supportive, wise, or useful in most applications. The second option is to place only less important information close to the center. This approach may be only applicable in very rare cases where the importance of the information to display is already known a priori. The third option is to distort the radial visualization, e.g., by a non-linear transformation of the radial data dimension, to ensure that less information is displayed in the inner-most region. The fourth option is to represent the center not as a single point but as a circle. This approach avoids the critical space in the center. It equals mathematically adding a constant offset to the radial component of every polar coordinate. With the design's focus on the task of analyzing wind data, the 2D Radial View does not implement the options one and two as the importance of data entities is not known a priori. To give an example, although slow winds or gusts will naturally be rendered close to the center and might be of less importance during some user tasks, they might play an important role in others. For this reason, neither some automatic smart-placement nor a discarding of entities performs practically in the center. Instead, the 2D Radial View supports the third and the forth option, which are both discussed below.

The view provides users the ability to display a void in the center of the visualization as depicted in Figure 4.2 in the lower right. This approach supports the research conducted by Diehl et al. [21], which concluded that the innermost ring or region in a radial layout is hard to memorize and should therefore be avoided or left blank. The concrete size of this void, i.e., the inner circle, can be changed by the user through a user-interface element, i.e., a slider. It has been found useful to set the initial size of the inner circle to nine percent of the radial visualization's radius. In a typical workspace layout with multiple data analysis views displayed across a standard pixel density, i.e., 96 dots per inch, computer screen, this circle size eliminates most of the aliasing artifacts and perception issues while leaving enough space for the rest of the visualization. With an increasing trend of high pixel density and high resolution displays in the mobile but also computer desktop market, no foolproof parameter can be given and manual tuning may be required. For practical reasons, the inner circle size is limited to a maximum of 50 percent of the radial visualization's radius. The lower limit is zero percent, which, depending on the situation, may be useful as it gives the user a classical polar plot of the data without any offset from the center point.

Additionally, the 2D Radial View implements the functionality to apply a non-linear transformation to the mapping of the currently assigned radial data dimension(s). Such a transformation results in a non-linear distortion of this/these radial data dimension(s). The distortion enables the user to focus on a particular region of the mapped numerical range. Figure 4.2 illustrates the effects of applying such transformations to radial visualizations. The feature can be used to counteract the above-mentioned non-linear effect of the polar projection. According to Wilkinson et al. [101], polar plots generally benefit from a non-linear transformation applied to the radial data dimension. This functionality is exposed to the user through a data mapping widget. Figure 4.2 shows this user-interface element with different settings for the transformation.

In conclusion, the proposed 2D Radial View offers two very different approaches to address the layout in the center of a polar plot. These two approaches can be combined freely or applied solely depending on the situation and/or desired presentation.

Visual Helpers and Guides

The 2D Radial view's design includes a series of visual elements, which are dedicated to guiding and assisting the user during the process of visual data analysis. Thus, they improve the overall efficiency of the visualization. These different visual helpers and guides are discussed below.

Radial Data Grid

A common task in visual data analysis is the comparison of disparate data entities across various data dimensions. The 2D Radial View uses the two dimensions radius and angle to position or address, i.e., look up, entities in the visualization. Displaying a "grid" and placing labels helps the user to compare, relate, or remember the positions of such entities. The word grid is used in this thesis although the used radial-equivalent to a two-dimensional Cartesian data grid looks more like a spider web instead of a grid. Drawing lines, which mark values with the same angle, results in radially outreaching spikes, whereas indicating all values with the same radius results in drawing concentric circles.

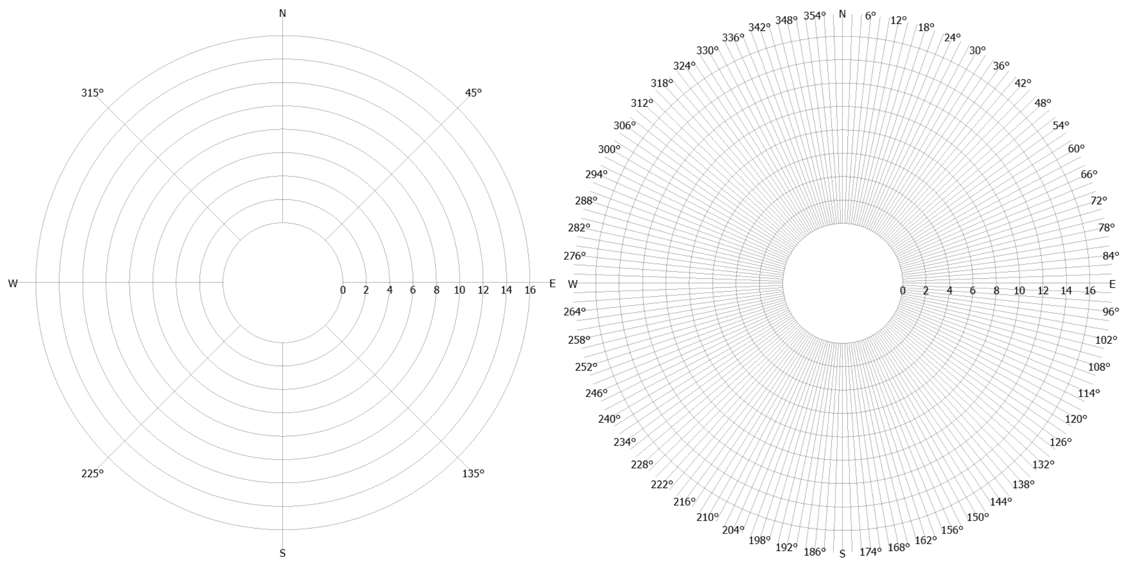


Figure 4.3: The 2D Radial View's data grid with a very sparse (left) and a very dense (right) subdivision of the angular data range. The subdivision results in radial sectors of the same size. Depending on the analysis task or desired look of the final presentation, the user can adjust this granularity.

As the 2D Radial View's angular data dimension is dedicated to present undistorted directions in the fixed range of $[0, 360)$ degrees or $[0,\ 2\pi)$ radians, the displayed range is known and can therefore easily be subdivided into equidistant sections. This results in drawing sectors in the radial visualization. For easy usability, the granularity of this subdivision is adjustable in a number of predefined discrete steps. Depending on the analysis task or desired look of the final presentation, users have the option to choose settings ranging from a very dense, e.g., a line every two degrees, to a very sparse, e.g., a line only every 45 degrees, configuration. Figure 4.3 illustrates a sparse and a dense data grid configuration. The initial grid layout is a subdivision into twelve sectors, hence a radiating line is drawn every 30 degrees.

Although the angular dimension presents a circular closed space, it has been found of importance to always include the line that represents zero degrees or, with respect to visualizing directions, “North” in the angular grid-arrangement. The label for each radial grid line is placed on the line’s outer ending. Thus, labels describing the angle are placed around the radial visualization. To emphasize and clarify the direct relation to real-world directions, the labels at 0, 90, 180, and 270 degrees are marked with “N” for North, “E” for East, “S” for South, and “W” for West.

In contrast to the view’s angular data mapping, the radial data mapping is not locked to a fixed range. Therefore, the actual subdivision of the grid’s rings is carried out by an algorithm. The algorithm divides the range at easy-to-read numerical values while ensuring a division into five to nine segments. This way, the grid interactively adapts to the data displayed in the radial data dimension and always provides the user with enough visual cues. Each ring is labeled with the numerical value it represents. These labels are placed on each ring slightly below the horizontal line which marks the 90 degree angle. This placement has been chosen as it favors the western world’s most common reading direction. By the proposed design, the labels are guaranteed by reading from left to right to show monotonically-increasing values.

The proposed radial data grid can be disabled by the user to reveal an unguided “pure” view on the data. Even in the absence of those visual cues, an effective data analysis may still be possible due to the view’s active and passive interaction features, which are discussed in Section 4.3. Besides the radial data grid, additional features discussed below have been designed to further visually aid and guide the user.

Tooltips

Although the radial data grid provides the user with some approximate information about the displayed entities, more precise information is often needed during data analysis tasks. It is obvious that it is not feasible to label each element when analyzing very large data sets. For this reason, the 2D Radial View gives the user precise feedback about the mouse cursor’s position within the two-dimensional polar coordinate frame or about the entities below the mouse cursor. To display this additional information, a box, i.e., the tooltip, is rendered with some offset from the mouse cursor. In this way, the user is able to investigate the exact numerical value(s) of an entity or the numerical range(s) of a region of interest. Figure 4.4 shows the tooltip mechanism in action. As shown in the figure, each component of the coordinates is preceded by the full label or name of the data assigned to its dimension.

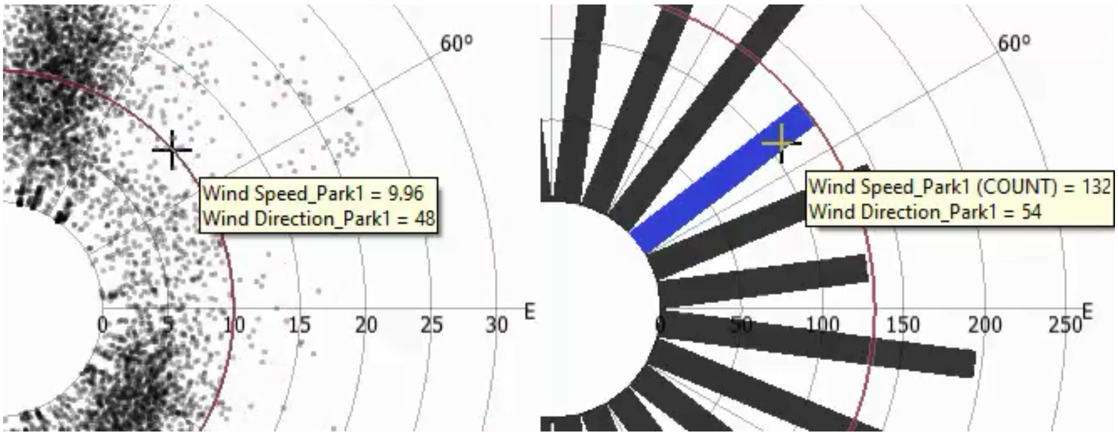


Figure 4.4: The 2D Radial View provides the user with additional information about displayed entities below the mouse cursor via a tooltip (left). Tooltips are also available for entities of aggregated data (right). If there is no entity below the mouse cursor, the tooltip exposes the cursor's exact position within the polar coordinate frame for enhanced navigation.

Visual Helper Ring

Additionally to the proposed radial data grid, the 2D Radial View implements a visual helper ring in order to aid users when comparing or differentiating plotted elements. The helper ring's main purpose is to support the user when comparing the radial magnitudes of values across the plot's assigned angular data as depicted in Figure 4.5. The motivation for implementing the visual helper ring is the fact that radial visualizations are in general not well suited for comparing radial magnitudes if the corresponding visual elements are located at diverging angles. Experiments on visual perception performed by Cleveland and McGill [16] showed that the accuracy decreases as the distance between judged objects increases. To overcome this limitation of a radial visualization system, the 2D Radial View renders a ring from the center, i.e., the outline of a circle, with the drawing radius of the mouse cursor's current position. In this way, the ring's outline intersects always with the mouse pointer.

Whenever the view operates in a mode where it displays aggregates of radial values, the helper ring snaps to the closest displayed aggregate. This approach ensures the easy and correct comparison of such values. The process of snapping is also reflected in the information exposed to the user via the displayed tooltip. The tooltip's behavior changes in the way that it now displays the aggregate's numerical value found by the snapping procedure and not the exact value corresponding to the current mouse pointer position in the polar plot. Although this may appear non-intuitive at first, the visual feedback of the helper ring, i.e., its offset in position to the mouse cursor, clarifies this special behavior.

The helper ring is always rendered fully opaque and superimposed over the rest of the visualization to ensure its persistent visibility. A *halo* around the helper ring adds a feeling of

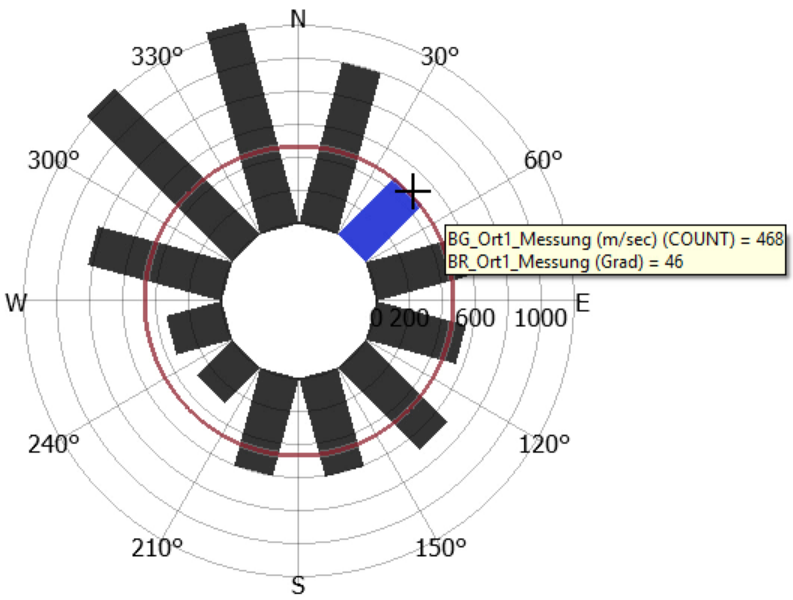


Figure 4.5: A visual helper ring supports the user when comparing radial magnitudes across the plot's angular dimension. A halo effect adds a feeling of depth and ensures perfect visibility.

depth [26, 29] thus making it appear above the rest of the visualization. This visual trick makes the ring better distinguishable from any visual elements below regardless of their color and brightness. The halo effect is achieved by first drawing a semi-transparent white line, which is drawn a little bit thicker than the line of the helper ring, and then the helper ring's actual line on top of it. Figure 4.5 shows the helper ring with its halo effect and depicts how this contribution visually improves the comparison process for radial magnitudes.

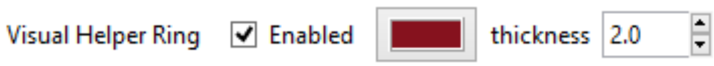


Figure 4.6: The helper ring's appearance, i.e., the thickness of the line or its color, is adjustable via graphical user-interface elements. This enables users to match the look with the color scheme they are used to work with every day or within another application. The helper ring feature can be turned off for situations where it has been found distracting.

An early version of the view displayed the helper ring whenever the mouse cursor entered the view's window. Users pointed out that this additional visual cue is in some situations during a data analysis distracting and/or undesirable. Additionally, users requested to be able to change the line thickness and color at runtime to match the color scheme they are used to work with or within another application. Based on this first field-test feedback, the 2D Radial View implements these features. Figure 4.6 shows the graphical user interface that enables the user to enable/disable the helper ring and change its appearance.

It has been found much easier for users to visually compare elements that are located at approximately the same angle but at different radii. For this reason, to avoid visual overload, and to avoid further occlusions, the 2D Radial View does not implement the angular-equivalent of the helper ring, i.e., a line originating from the center of the plot which intersects with the current mouse position. It suffices to display the lines provided by the radial data grid in order to guide the user in the angular data dimension.

Data Layers

The 2D Radial View uses four distinct layers for visualizing different aspects of its assigned data. These layers are: A layer for all assigned data, a layer for currently selected data, a layer for context (data of a previously created selection), and finally a layer for temporally highlighted data by hovering over their visual representation (see Section 4.3). Each layer can be drawn independently. This results in a high flexibility for composing intermediate or final visualizations. More importantly, by implementing this separation, the user gains control for adjusting the visualization attributes of individual layers such as color, opacity, size, and so on. Another advantage is that, based on the analyst's workflow, certain data layers can be disabled, i.e., excluded from drawing, in order to focus on a special aspect in a view.

Radial Scatter Plot

Based on the feedback from data analysts in the field of wind energy production, a scatter plot poses a preferred presentation choice for quickly analyzing data dispersion, correlation, and for identifying outliers. Scatter plots are often used to gain a first impression of some data and to start a deeper data analysis. The 2D Radial View implements a radial equivalent to a two-dimensional Cartesian scatter plot. Values of the assigned data are visualized by drawing dots on the plot's polar plane. Individual dots are drawn for the data of all enabled data layers. Visualization attributes of the dots are adjustable for each data layer via graphical user interface elements. The controllable attributes are discussed below.

Transparency and Size

The design features an adjustable size for plotting the dots. The size attribute may be adjusted by the analyst to make a data layer appear more important than another, make dots more or less overlap, or just to customize the plot to the preferred look for a final presentation. The proposed design uses transparency to reduce occlusion of overlapping dots. Dots are per default drawn semi-transparently and blended together additively. By employing this technique, dense regions appear darker than sparse regions whenever a large number of values is plotted. Hence, by using transparency and blending when plotting values, this visualization approach becomes applicable to big data. Figure 4.7 illustrates the benefit if using transparency for a larger set of data.

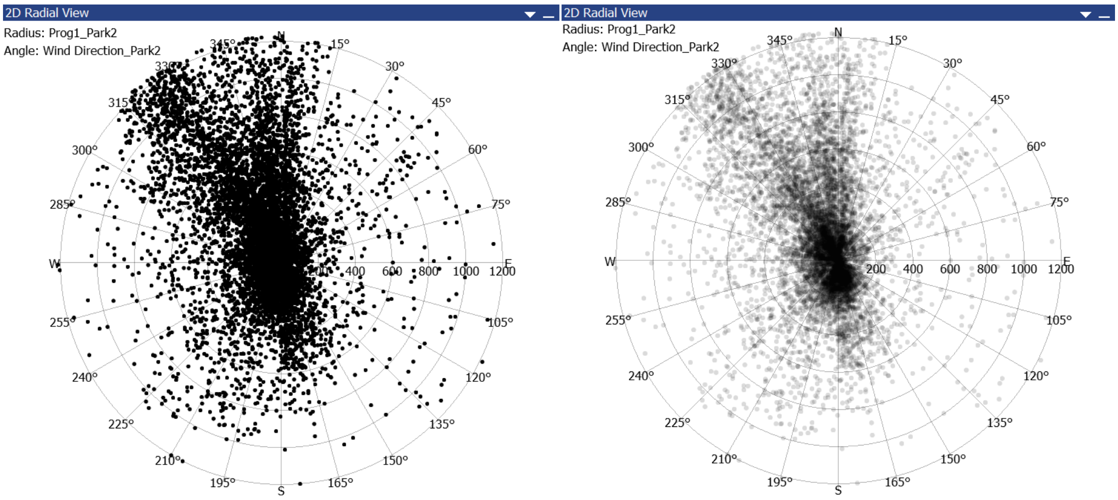


Figure 4.7: The 2D Radial View is used for a scatter plot of a larger data set. If no transparency is used (left), overlapping dots completely mask dots at the same position. Changing the transparency and size attribute of dots greatly improves such a visualization as dense regions appear darker than sparse regions (right). Hence, data analysts gain more insight into the visualized data.

Color and Intensity

The 2D Radial View's scatter plot design utilizes color in two different ways depending on the number of data dimensions assigned to the view's two visualization dimensions. If exactly one data dimension is assigned to each visualization dimension, four distinct colors are used to visualize the affiliation of each dot to its data layer. For the application specific task of comparing wind data from two or more geographical locations, assigning multiple data dimension to each visualization dimension is supported. If multiple pairs of data dimensions are assigned to the view's visualization dimensions, a dot for each data dimension pair and each data layer is drawn. In this situation, coloring is used to differentiate between the dots of different data dimension pairs. A color legend above the visualization communicates this information to the user. As the color attribute cannot be used anymore for the visualization of data layer affiliation, color intensity (i.e., color saturation) is used instead. Figure 4.8 illustrates the usage of color for comparing recorded wind data of two different locations. The dots for each set are drawn in serial. This approach may result in the occlusion of dots of a previously drawn set. To overcome this limitation, the drawing order of the colored sets can be changed by the user.

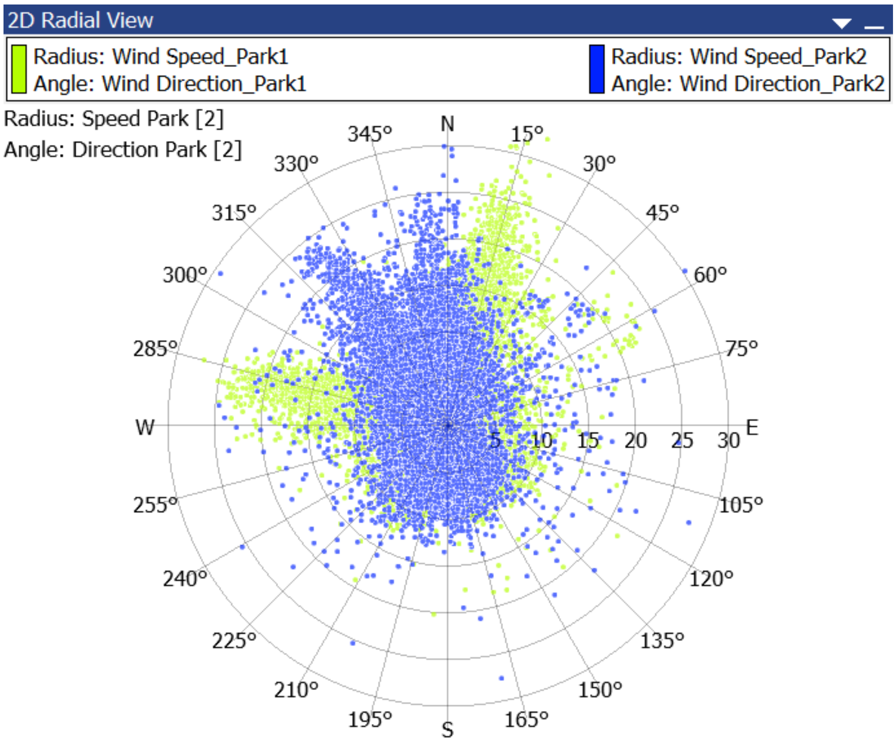


Figure 4.8: The 2D Radial View's scatter plot supports plotting dots for the values of multiple pairs of data dimensions in a single visualization. Coloring is used for visualizing data affiliation. A color legend above the visualization communicates this information to the user.

Radial Line Plot

The visualization technique *Radial Line Plot* is motivated by the star plot design pattern, which represents data to the user by drawing a closed shape. A shape has the advantage that it can be compared (by its size and form) very fast and easily to another shape. Shapes therefore often serve as the inputs of a visual classification process which uses characteristic shape profiles as classifiers.

The Radial Line Plot's approach uses equidistant angular data binning. For the data in each bin an aggregated value of the assigned radial data values is calculated and visualized by a bent line at the position of the aggregated value. The line endings of adjacent bins are connected by a line in order to form a closed shape. However, empty bins produce no aggregated values thus no lines and cannot be connected meaningfully with adjacent lines. As a result, the final shape is not closed if there exists one or more empty bins. The key benefit of plotting aggregates is the better visual scalability compared to scatter plots. For example, a line plot displaying the aggregates of a thousand points results in the same overdrawing as a line plot displaying the aggregates of one million points.

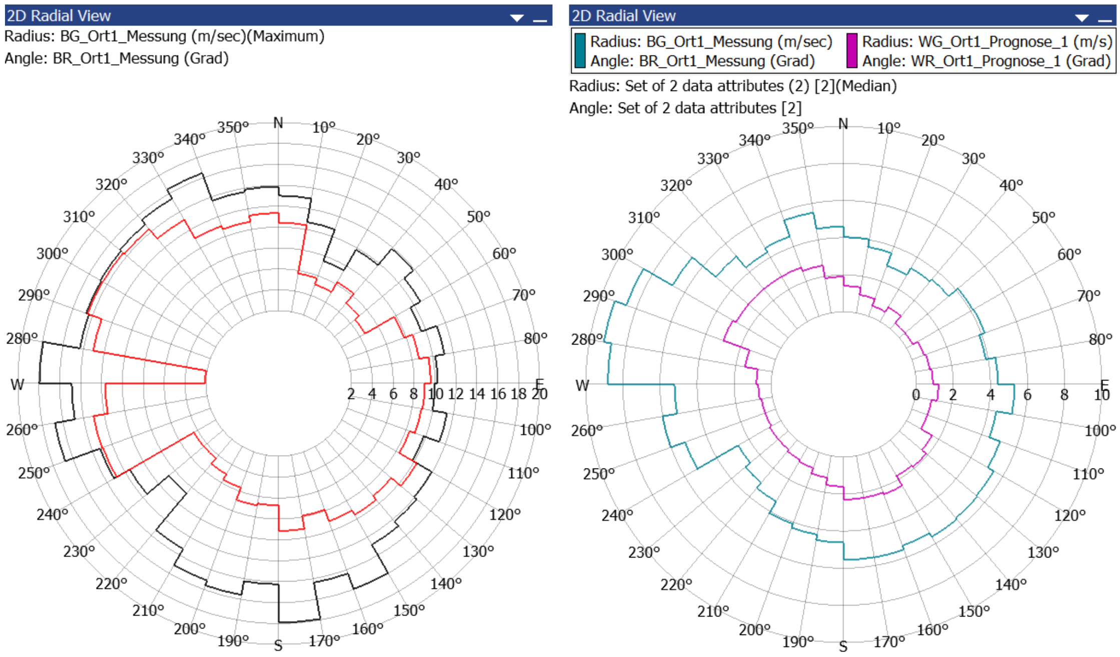


Figure 4.9: The Radial Line Plot empowers analysts to visually analyze different aggregates of data with a circular component. By using connected lines as visualization primitives, shapes are formed. Color is used to either indicate the corresponding data layer (left) or data pair (right). The proposed design enables analysts to compare shapes of different data in a single visualization.

The 2D Radial View supports the calculation of the following aggregates for its angular data bins: the minimum, the maximum, the count, the median, and the average. The default number of bins is 24, which results in bins spanning 15 degrees. The user is encouraged to alter the number of bins in the range from twelve (a bin for every 30 degrees) up to 144 (a bin every 2.5 degrees). By drawing a line for every aggregated value, this visualization technique is well suited for visualizing large numbers of records.

Color and Intensity

The proposed design utilizes color and color intensity in the same way as the *Radial Scatter Plot*. The coloring of the shape depends on the number of data dimensions assigned to the view's two visualization dimensions. If exactly one data dimension is assigned to each visualization dimension, four distinct colors are used to visualize the affiliation of each shape to its data layer. If multiple pairs of data dimensions are assigned to the view's visualization dimensions, a shape for each data dimension pair and each data layer is drawn. Color is used to differentiate between the shapes of different data dimension pairs. A color legend above the visualization communicates this information to the user. As the color attribute cannot be used anymore to indicate data layer affiliation, color intensity (i.e., color saturation) is used for this purpose instead. The proposed design empowers the user to compare shapes of different data in a single visualization instead of performing manual side-by-side comparisons. Figure 4.9 depicts how this design supports analysts in comparing wind data from two different locations in a single visualization.

Radial Bar Plot

The visualization technique *Radial Bar Plot* is motivated by the Cartesian bar chart technique, which is suited for comparing magnitudes. As Cartesian techniques are not well suited for circular data, the 2D Radial View proposes a radial layout, which uses radiating bars as its visualization primitives.

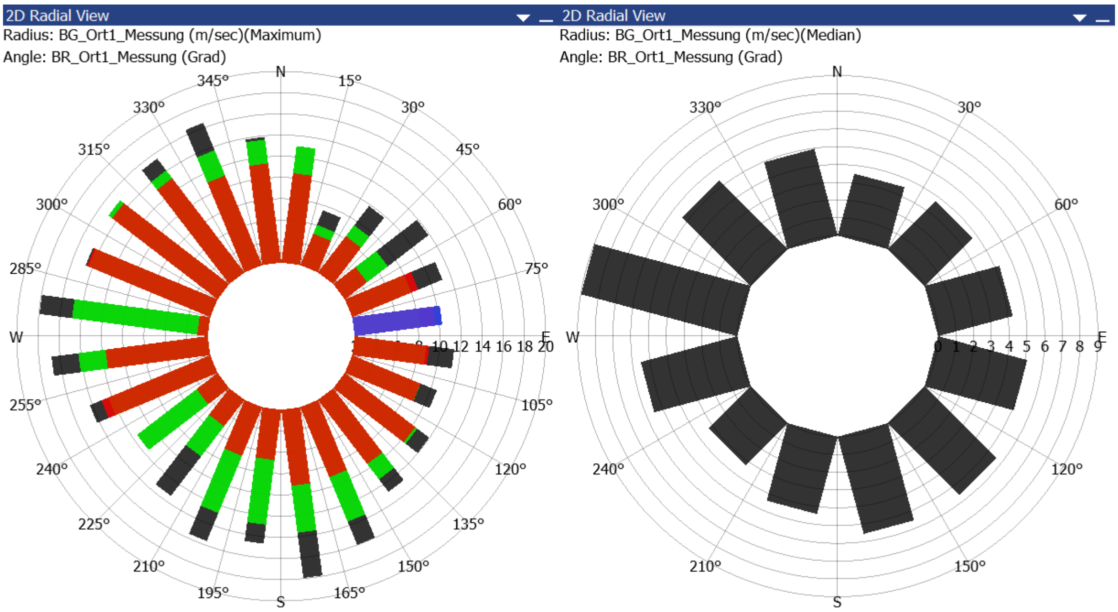


Figure 4.10: The Radial Bar Plot visualizes aggregates of angularly binned data with rectangular bars radiating outwards from the inner circle baseline. The width of a bar depends on the number of angular bins and the circumference/radius of the inner circle. On the left, the maximum of each bin and data layer is visualized by a colored bar. On the right, the median of all data entries per bin is visualized. Due to a reduced number of bins, i.e., only twelve, and a bigger inner circle radius, the bars are drawn much wider.

The geometric radial equivalent of the bar chart is the rose diagram. However, this visualization technique is problematic for comparing magnitudes as polar plots face an uneven distribution of display space. With increasing distance from the center, the available two-dimensional display space grows quadratically. Hence, a wedge with double the radius of another wedge (with the same angle) has four times its area. This makes such structures to be perceived as being much more important than their difference in magnitude indicates. Wilkinson et al. stresses this fact by stating: "Attractive as it may be, the polar bar chart confounds area with radius" [101]. Additionally, visually comparing very similar angular magnitudes happens to be difficult. Draper et al. [24] point out "when wedges in a pie chart are almost the same size, it is difficult to determine visually which wedge is largest" and conclude that bar charts are therefore the better choice for this task. In a series of experiments on graphical perception conducted by Cleveland and McGill [16], it was shown that "Position judgments are the most accurate, length judgments are second, angle and slope judgments are third, and area judgments are last.". To address these issues, the 2D Radial View uses rectangular bars radiating from the inner circle in the view's center. Thus, the area increases linear with the bar's height.

The visualization technique uses equidistant angular data binning. For the data in each bin, an aggregated value of the assigned radial data values is calculated and visualized by a bar which ends at the position of the aggregated value. The width of a bar depends on the number of bins and the circumference/radius of the inner circle. Thus, this approach has the disadvantage of requiring an inner circle as the bar's width converges towards zero with a decreasing inner circle radius. Figure 4.10 depicts this circumstance. By drawing a bar for every aggregated value, the proposed visualization technique is well suited for large numbers of data records.

Visualization Errors

As the bars are drawn rectangular instead of bent, a number of visualization errors are introduced: Firstly, the bar's baseline is rendered as a straight line instead of an arc. With a decreasing number of bins, this error increases. Thus, the lower limit for the number of bins is set to twelve (a bin for every 30 degrees), which has proven to be an acceptable lower limit for analyzing wind energy production data. Secondly, the bar's end is rendered in the same width as the baseline circle segment although the corresponding arc presenting the angular bin at this position is much longer. This error increases with the bar's height. Thirdly, the magnitude visualized by the bar's end is only correct in the center of this end. The two corner points do not reflect the correct magnitude on the underlying data grid (when enabled). This error increases with decreasing distance from the polar center. All these visualization errors mentioned above are visible in Figure 4.10 on the right. However, as the view's primary purpose is conveying global structure, with numbers only as details on demand, users preferred these drawbacks over the aforementioned distortion artifacts.

Color and Transparency

The Radial Bar Plot uses coloring to indicate data affiliation to the four data layers. Transparency and blending is supported to reveal the underlying data grid and visualizations of the data layers below. Figure 4.10 illustrates the coloring of the different data layer visualizations.

Radial Box Plot

Although line and bar charts scale much better than scatter plots, they hide all information about the distribution of the values within each bin. Data analysts in the field of wind energy production, however, are interested in gaining insight into the distribution and deviation of numerical data related to recorded or prognosticated wind or gust directions. Box plots are a widely used technique for visualizing multiple important statistical values of distribution and deviation. Thus, the 2D Radial View employs a design which is based on drawing multiple boxes that are integrated into the view's common radial layout. The resulting visualization technique, i.e., the *Radial Box Plot*, is in principle an evolved technique of the previously discussed Radial Bar Plot. Instead of using multiple bars, which radiate outwards from a common baseline, multiple rectangular box plots are drawn (see Figure 4.11). By using rectangular shapes instead of radial wedges as visualization primitives, a consistent design with the Radial Bar Plot is maintained. However, this design choice therefore results in the same advantages and disadvantages as already discussed for the Radial Bar Plot.

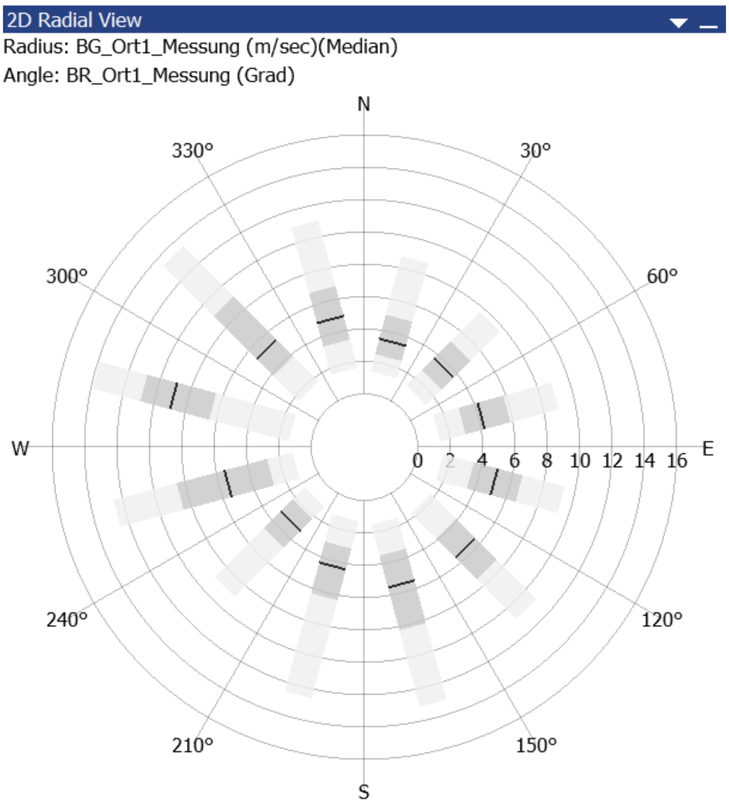


Figure 4.11: The Radial Box Plot enables data analysts to directly observe the dispersion of circular data. The approach uses equidistant binning of the assigned angular data and calculates multiple statistical aggregates (i.e., the median and various percentiles) per bin. The aggregates are then used for forming boxes and drawing the median as a black line.

The approach uses equidistant binning of the assigned angular data. In contrast to the Radial Bar Plot, which calculates a single aggregated value for each bin, multiple statistical aggregates are calculated, which are then used for forming boxes. As Mühlbacher and Piringer [69] point out, there exist a variety of options for the visualization of univariate distributions. The 2D Radial View implements their proposed variant for box plots, which visualizes the median as a black line, the inter-quartile range as a light gray box, and the ranges from the 0.05 to the 0.25 percentiles and from the 0.75 to the 0.95 percentiles by dark gray boxes. Figure 4.11 illustrates this technique, which enables data analysts to directly observe the dispersion of data in each radial bin.

As each box represents multiple statistical values (i.e., the median and various percentiles), the visualization is able to convey more information in a single image than the Radial Bar Plot. The Radial Box Plot's design does on purpose not support assigning data of multiple data dimensions to its visualization dimensions. As already multiple values of each data layer are displayed per bin, visualizing even more values makes the final image cluttered and thus hard to read. The task of comparing aggregates of multiple data-dimension pairs in angular bins in a single image is covered by the design of the Radial Line Plot.

Color and Transparency

The Radial Box Plot uses coloring to indicate data affiliation to the four data layers. Transparency and blending is used to reveal the underlying data grid and the visualizations of box plots of the data layers below.

Radial Divergence Plot

The *Radial Divergence Plot* uses a visualization technique, which has been specially designed to visually reveal the entity-wise divergence of two angle/radius data dimension sets. The proposed technique is motivated by the task of analyzing weather forecasts by comparing the actual recorded values with their predicted values.

To visualize the divergence of two entities that each have its source in one of the two data dimension sets, the approach draws a line, which connects the two relating entities (see Figure 4.12). Whenever two entities perfectly coincide, i.e., their divergence is zero or below the visible threshold, no line connection is visible. This way, identifying diverging entities becomes an easy and straightforward task for the user. The approach also enables data analysts to identify patterns in the entity-wise divergence, which may be caused by systematic errors (e.g., shifts by a steady value) of prognosis models. Clusters of lines, on the other hand, may indicate errors, which are local, i.e., occur only in a very specific data range. However, choosing the right visualization technique to indicate this divergence is essential. The important design decisions are discussed below.

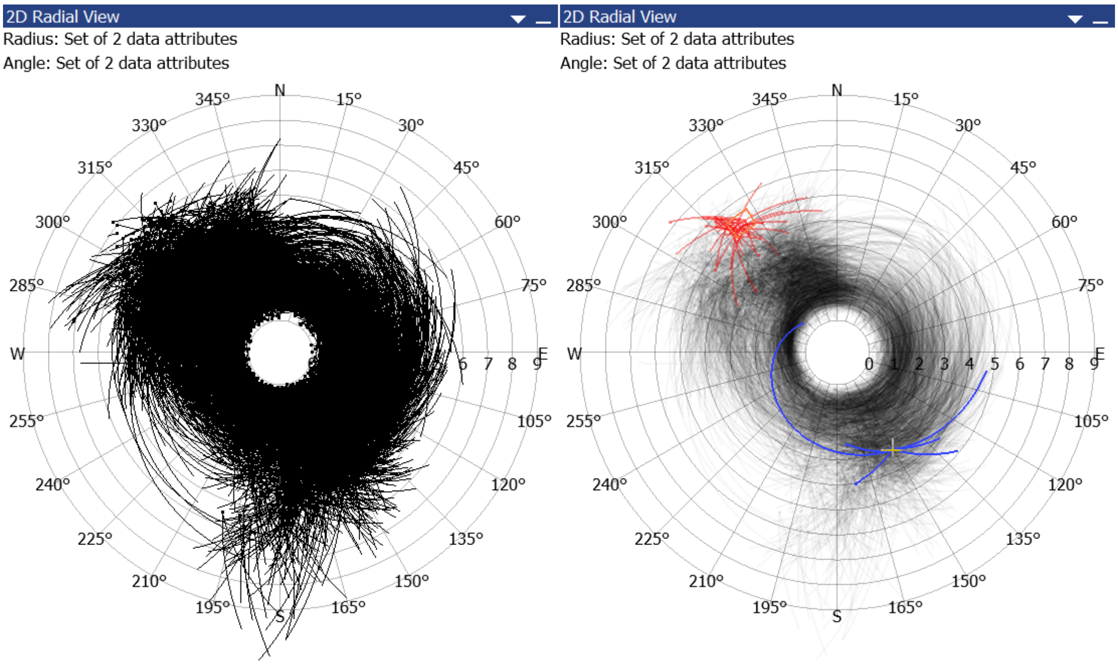


Figure 4.12: The Radial Divergence Plot is used to visualize a large number of records by drawing a large number of arcs. Without transparency (left), massive overdrawing issues occur resulting in a loss of available information. By employing transparency, denser and sparser regions are revealed (right). The color and amount of transparency is adjustable for the arcs of each individual data layer. In a default setting, all entities may be drawn transparently, whereas highlighted entities (red and blue) may be drawn opaquely and in different color for increased visibility.

Connection Drawing

As a first approach, a straight line was used to visualize connections. Although this provided a first impression of the potential of this visualization technique, the result was not satisfying. Even with a small number of data entities, a huge number of lines was crossing the view's void center (if enabled) and lines were intersecting other lines at very steep angles. This resulted in a lot of visual clutter, which made interpreting the final image very difficult.

In order to eliminate these problems, the 2D Radial View draws a bent line for every connection and ensures that lines never intersect the view's void center. The approach draws each line by linearly interpolating in each of the start and end point's polar coordinate components. This interpolation in polar space results in nicely bent and thus natural looking connection lines.

As the view's angular dimension is a circular data space, in many cases a simple linear interpolation of the radial values will produce an arc which covers more than 180 degrees (i.e., the major arc) in the polar plot because circularity is ignored by the interpolation procedure.

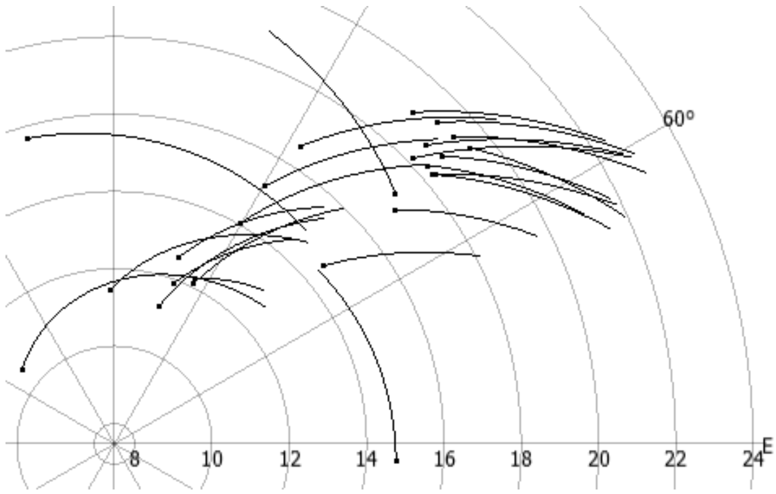


Figure 4.13: In order to be able to distinguish between the start and ending of a connection, the Radial Divergence Plot draws a small filled square at the end of each arc. By drawing a small square instead of a complex geometric object, e.g., an arrow's head, the approach avoids the introduction of additional visual clutter.

In these cases, a shorter line connection is always available (i.e., the minor arc), which results in less overdrawing, and is thus desirable in cases where the direction of the divergence is not known or irrelevant. As this is the common case, the 2D Radial View always identifies and draws the minor arc between two entities.

However, a simple line does not provide the user with any information about its start or ending. In order to be able to distinguish between the start and ending of a connection, i.e., to identify its direction, the 2D Radial View draws a small filled square at the end of each line, as illustrated in Figure 4.13. This approach has been found to provide the user with the same visual cue as when drawing a bent arrow but introduces less visual clutter. Furthermore, drawing a simple geometric figure such as a square is very fast and efficient on modern graphics hardware, which becomes important when rendering large numbers of entities.

Color and Transparency

The Radial Divergence Plot uses coloring to indicate data affiliation to the four data layers. With an increasing number of visualized arcs, overdrawing issues limit the available visual information and the visualization becomes cluttered. In order to analyze big data, transparency is used to overcome this problem. By blending semi-transparent arcs, dense regions become darker and sparse regions lighter. This approach makes the visualization suitable for analyzing large data sets, as demonstrated in Figure 4.12.

4.3 Interaction Design

This section discusses the design choices, which lead to all the available interactions with the 2D Radial View. The proposed view provides different *passive* and *active* interactions. These interactions are explained in the discourse below.

Passive Interactions

A *passive interaction* is an interaction that influences the current view but does not originate from the view itself but from another component within the system. The component may be any other view, an external program, or a computation that changes data or information about data, which is currently assigned to the view. As the component can be any other view, it might also be another instance of the "same" view, e.g., one 2D Radial View instance influences data which is also mapped to another 2D Radial View instance.

The proposed 2D Radial View supports passive interactions within the visual analysis software framework by providing the user with instant visual feedback. Data selections within the system, i.e., regardless of their origin, are immediately reflected in all affected visualizations. Any data changes, e.g., adding or removing values, changes to the format or to the value(s), are also reflected. No manual user interaction is ever required to invalidate and redraw a visualization inside the proposed 2D Radial View. Thus, *visual consistency* is ensured among all views within the system.

Interaction Modes

The 2D Radial View implements different active user interactions. If the context admits for more than just one user interaction, the user's intention for interacting with the view must be known a priori. Many software suites use combinations of mouse buttons and/or keyboard shortcuts to setup this intention. Such actions are hard to remember if the user is working with multiple software products and also hard to look up if they are hidden in menus. Icons which can be accessed from a toolbar area within the application, however, are easier to reach than menus and their sub-menu entries. Their position and purpose can easily be remembered by the user.

The 2D Radial View's design proposes the use of large and easy to reach icons located in a toolbar to enable users to switch between the different active interaction modes. The available modes are discussed in the discourse below.

Data Selection

The 2D Radial View enables the user to select data through a *browsing mechanism* as discussed in Section 2.2. The user is encouraged to perform a direct selection of a data subset inside the displayed visualization. The 2D Radial View's approach is based on performing selections through visually defining ranges, i.e., intervals of membership. The task of selecting individual data entities is best carried out in other views, for example, in a *detail view*, which displays a list of individual entities. Such views are available to the user by the visual analysis software environment.

By the 2D Radial View being designed as a two-dimensional data plot, it is possible to visually define a selection based on intervals restricting either a single data dimension or both dimensions at once via a logical AND combination. Hence, in total three different selections are attainable: A selection based solely on the angular dimension, a selection based solely on the radial dimension, and a selection based on both dimension via combination. However, only two selections are implemented based on feedback received by domain experts. They explicitly pointed out that a one-dimensional selection on the radial dimension is hardly ever needed when carrying out their tasks and can easily be performed in other (linked) linear views. The selections, which incorporate a circular data dimension with its circular closure property (see Section 3.2), on the other hand, were required to be carried out more efficiently.

Although the 2D Radial View supports many different visualization techniques, the data selection has been designed to be independent of the employed technique. Instead, it is always defined as continuous intervals on the angle and/or radius dimensions. This approach offers two advantages over implementing different interactions for each visualization technique: Firstly, it is easy to understand as it operates identically in all of the view's visualization techniques. Secondly, the properties, i.e., data ranges, of a data selection can be visualized independently of the selected visualization technique.

In a modern visual analysis software framework, the information about a data selection is stored centrally and it is decoupled from the generating view. Thus, data selections, which are created by a 2D Radial View, become available to the complete system. Furthermore, selections, which are applicable to a view, are automatically updated and reflected in that view. The proposed interactions for performing data selections in a radial layout follow the interaction principles that are implemented in already existing views of the used system. This drastically reduces the initial time for learning the task of performing data selection(s). The individual selections and their interaction mechanisms are discussed in more detail below.

Angular Selection

The angular selection enables the user to perform a data selection based on the 2D Radial View's angular dimension. The selection is described by an interval in this circular data space. As such, the selection is defined by two distinct values in this data space - a start and an end value. When designing the required user interaction for this task, a *click, drag, and release* mouse solution has been chosen as users are already familiar with this interaction mechanism. Such a solution is present in many common operating systems, e.g., when selecting multiple files in a file browser. However, a two step approach, e.g., performing two separate mouse clicks, might offer a higher precision at the cost of not conforming with the interaction mechanisms of the system.

Clicking somewhere inside the 2D Radial View initiates the angular selection procedure with the angular value of the mouse cursor's position. Dragging the mouse to the desired angular end value and releasing it ends the procedure. Visual feedback is provided during this action by drawing lines for the two limits of the selection interval (see Figure 4.14). These lines, radiating from the center in the direction of the start and end angle value, are independent of the radial dimension's data and mapping, and thus always visible to the user.

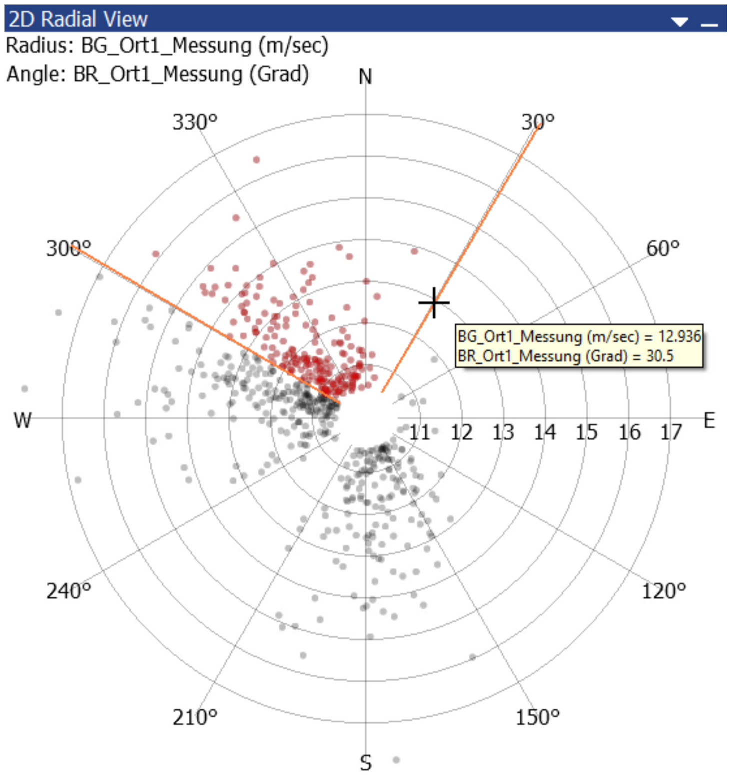


Figure 4.14: The 2D Radial View enables users to perform interactive selections based on angular values of the assigned circular data. For the start and end value of the selection a line radiating from the center is drawn to guide the user during this operation and to visualize the angular selection.

Two-Dimensional Selection

The 2D Radial View's two-dimensional selection enables users to select entities based on the view's data assigned to its radial and angular data dimension. Compared to a one-dimensional selection, the user must choose four values to fully define the two-dimensional region. This results in a greater design space for this interaction. For instance, three clicks may be used to define the selection in a step-by-step approach. However, to conform with the visual analysis system the interaction is again implemented with a *click, drag, and release* mechanism similar to the angular selection discussed above. In a Cartesian plot, such as a Cartesian scatter plot, a two-dimensional selection is often carried out by defining a rectangle in the displayed Cartesian space. The four interval values, i.e., the start and end values in each dimension, limit this rectangle. However, due to the polar projection, a rectangle in Cartesian space becomes a bent region in polar space. To reflect this property, the limits of the radius dimension's interval thus are drawn as bent lines. Figure 4.15 illustrates the 2D Radial View's approach to visualize a two-dimensional selection.

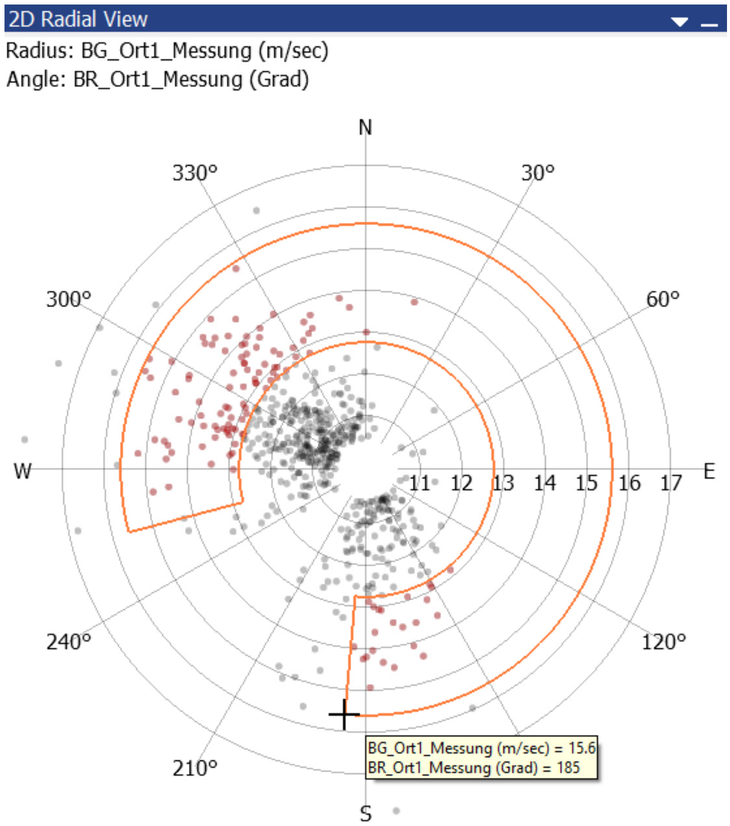


Figure 4.15: The 2D Radial View enables users to perform an interactive two-dimensional selection. The limits of the selection are drawn to guide the user during this operation and to visualize the selection. The limits of the radius dimension's interval are drawn as bent lines due to the polar projection.

A limitation of this approach is that the user can only construct a two-dimensional selection within the currently displayed data range. Furthermore, as the data mapping can be changed by the user, one or both radius interval limits of a selection may not get rendered if their value is located outside of the currently displayed data range.

Selection Modifications

It is a central task in the visual analysis process to modify data selections. In the proposed design, selections can be refined through two different actions: Firstly, the 2D Radial View enables the user to alter an existing selection by dragging and dropping its selection border handles, i.e., the line representation of an interval limit, via mouse input. However, this action is only applicable to handles that are currently visible in the visualization, i.e., reside in the range of the defined data mapping. Furthermore, the precision of this operation is limited by the resolution of the used display space. Another limitation of this visual approach is that a selection can only be shrunk or extended to the limits of the visible range of the view's data mapping. To address these issues, the 2D Radial View implements the option for performing a direct numerical alteration of the selection interval limits. To realize this, graphical user interface elements serve the user as a numerical feedback about the properties of a selection and can be used as input fields for direct numerical alteration.

The 2D Radial View implements browsing interactively through data with an existing selection acting as the brush. Whenever the mouse cursor enters the region of an existing selection, the cursor's icon changes to inform the user that a brushing action is available via clicking and dragging. This interaction enables the user to shift a selection to a new position, i.e., change its interval limits without changing its initial form. During data brushing, the selection's position is shifted in either one or both dimensions depending on the underlying selection type and the user's mouse path. Due to the implemented view linking technique, all changes to a selection are interactively reflected in all views. Although the brush's form is limited to the two basic forms that can be created by an angular or two-dimensional selection, it has shown to be a powerful feature, which provides analysts enough flexibility for performing common tasks.

Circular Categorization

Wind energy domain experts are especially interested in data aggregated for different wind directions. The proposed Radial Line Plot, Radial Bar Plot, and Radial Box Plot provide such aggregated data by performing evenly binning based on the circular data component. However, experts pointed out that this fixed and evenly-distributed binning is not flexible enough for their work of classifying and extracting information of their wind energy production data. A common domain task is to classify the data based on some dominating wind directions and extract the sum of the total power output in each class. This task provided the motivation for implementing a one-dimensional categorization for circular data without constraining it to some predefined layout.

To accomplish a free layout, i.e., a categorization solely defined by the user, the 2D Radial View proposes the following approach: The user is encouraged to visually define and alter *categorization boundaries*, which in turn automatically define the intervals for the categories. Defining boundaries instead of intervals has two advantages: Firstly, the data range is always fully categorized without any undefined ranges. Secondly, establishing the preferred categorization is simple and straightforward. Exactly two boundaries are needed to define an initial partitioning into two categories due to the circular closed value space of circular data. Adding an additional value boundary effectively splits an existing interval into two and thus replaces the affected category with the two newly created ones.

The first time the circular categorization feature is enabled by the user, the 2D Radial View initializes with a partition into three categories. A mechanism for partitioning data into only two categories is already provided via the technique of data selections. The initial setup is not laid-out by some data-aware algorithm. Instead, it serves as a starting point for the user to interactively form the desired categorization.

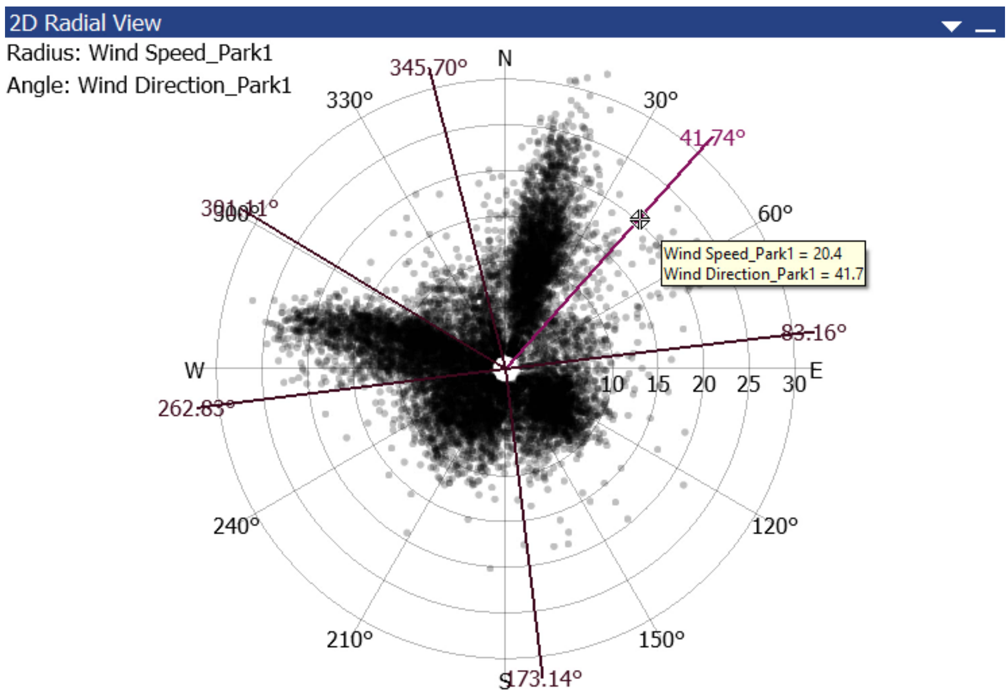


Figure 4.16: The 2D Radial View's interactive circular categorization feature enables users to classify circular data based on visual information. Boundary handles defining categories (i.e., the violet lines) can be added by clicking into an existing category. Each handle can be repositioned by dragging it inside the radial visualization and dropping it at the desired value.

The categorization boundaries are represented by lines radiating from the view's center. The lines are rendered on top of the visualization. At their outer line endings their numerical value is reflected to the user via a text label similar to the labels of the radial data grid. Categorization boundaries can be altered by dragging and dropping their visual handles, i.e., their line representation, with the computer mouse. While dragging a handle, it is highlighted by rendering it with a thicker line width and in a brighter color. A new boundary is introduced into the system by clicking into any existing category. The proposed approach requires only minimal user input in order to form a circular categorization based on visual information. Figure 4.16 depicts this interactive circular categorization feature.

In addition to being able to visually define circular categories, the 2D Radial View also provides the user with exact numerical feedback about and the option to fine-tune the values defining the categorization. This functionality is provided via graphical user interface elements as illustrated in Figure 4.17. Changes to the categorization are immediately reflected in the system by automatically updating the categorical information for the data and by design all involved visualization(s).

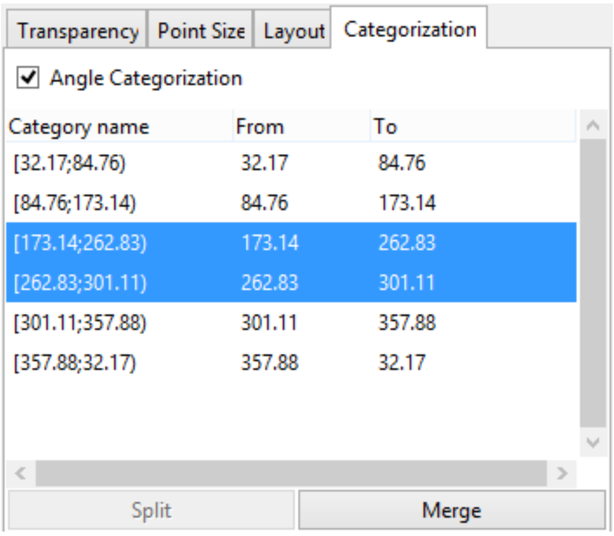


Figure 4.17: Graphical user interface elements show the user exact numerical information about the circular categorization. The values defining a category can be easily changed by entering new values. If a single category is selected in the list, it can be split into two new ones. If two adjacent categories are selected, as shown in the figure, they can be merged.

The result of the circular categorization process is information about categorical membership of data based on the defined intervals. This information can be used within the software suite in various scenarios such as selecting or filtering data based on this membership. A common task is aggregating data for each category and visualizing the resulting information for further investigations. Figure 4.18 illustrates such a scenario and the resulting interplay between different views.

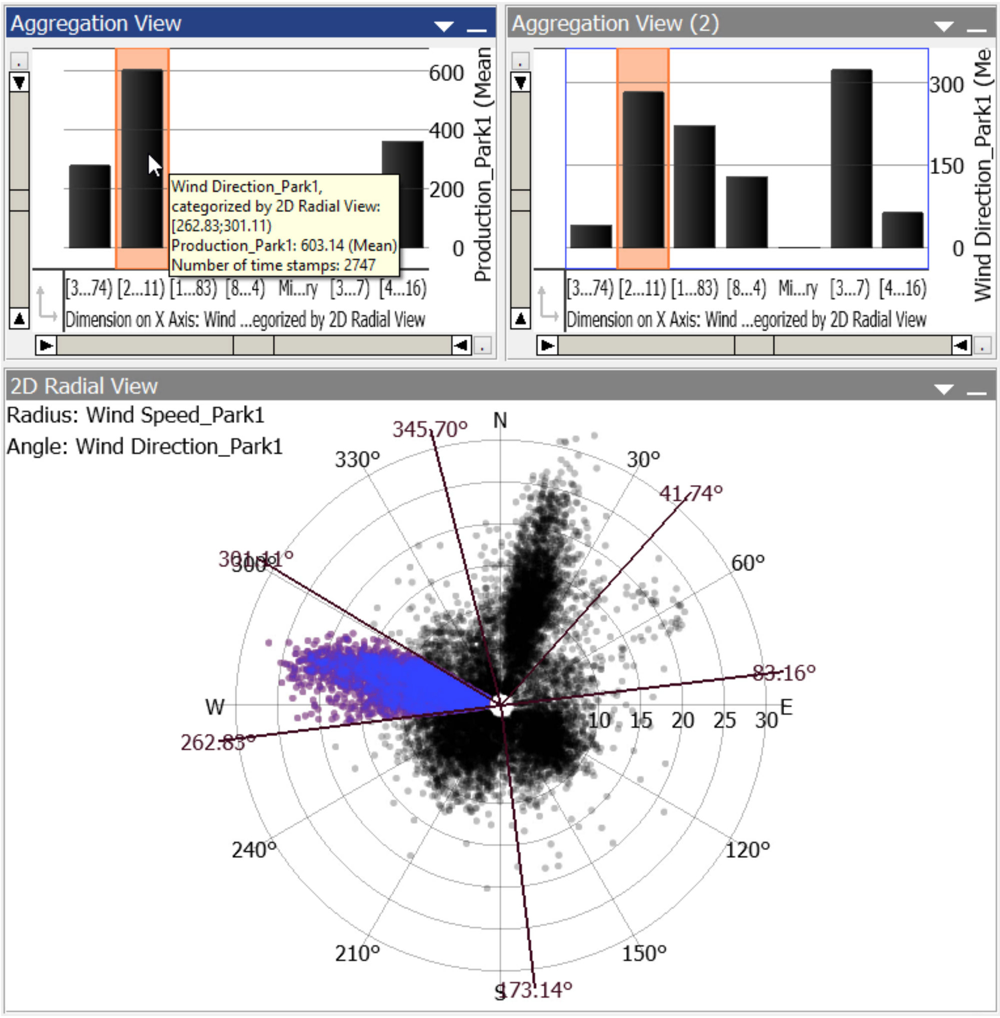


Figure 4.18: The 2D Radial View's circular categorization feature is used to classify entities into six distinct categories (bottom). The resulting categorical information is then used as the input for generating and visualizing aggregated data (top). Data records that lack angular information are presented by a bar labeled *Missing entries*. As the views are linked, hovering over a bar in a top view highlights the contributing entities inside the 2D Radial View.

Peek Brush

There exist many situations during the process of visual analysis where analysts want to get insight how a particular displayed graphical entity, e.g., a bar in a bar chart, relates to its data. A common way is to select the entity and thereby observe its related values in another view. However, this requires the user to change or create a new selection. This action may interrupt the user's workflow. To overcome this problem, the visual analysis software framework provides the concept of *peek brush* [8]. Peek brush enables the user to highlight data relating to the entity below the mouse cursor by just hovering over it at any time. Hence, no other task is interrupted and no interaction mode has to be switched to perform this action.

What data gets highlighted by the mouse-hovering event depends on the selected visualization mode of the 2D Radial View. In a scatter plot, data close to the mouse cursor is highlighted. In the bar, line, and box plot mode, the data corresponding to the angular bin of the bar, line, or respectively box is highlighted. Figure 4.19 illustrates the user peek-brushing on a bar in a Radial Bar Plot, thus observing its relation to another data dimension in a histogram view. Due to view linking, this feature is also available as a passive interaction, e.g., hovering over an entity in another view highlights matching data in instances of the 2D Radial View.

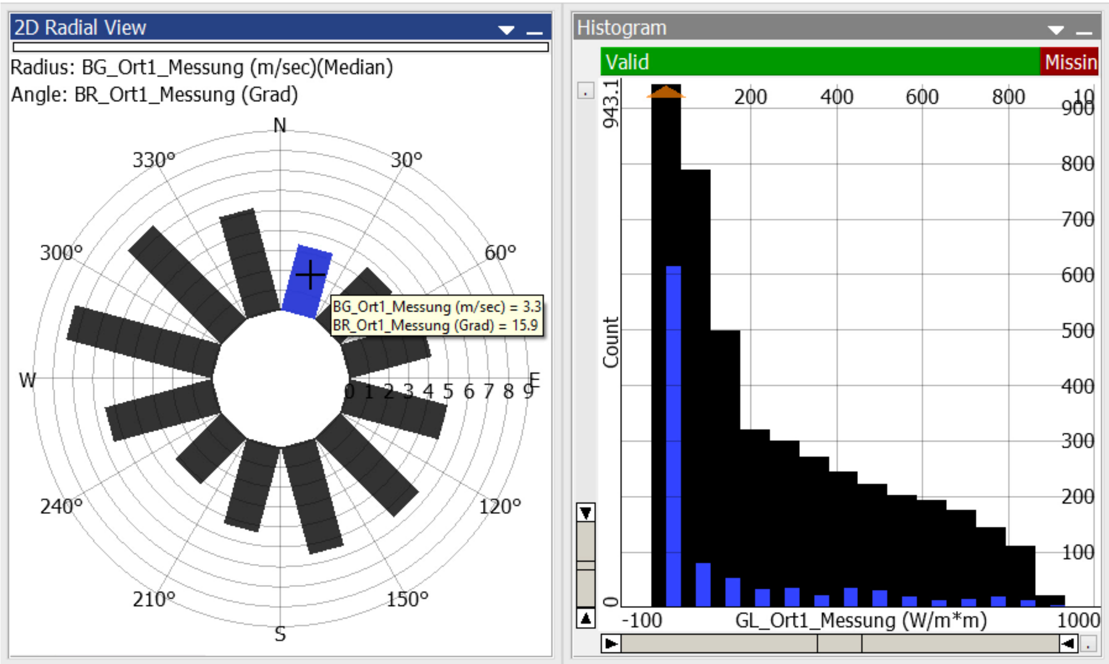


Figure 4.19: The concept of *peek brush* enables data analysts to easily attain insight into how a particular displayed entity relates to its data. Data relating to the entity below the mouse cursor gets highlighted when hovered upon. Due to view linking, this features is available across many views. In this example, hovering over a bar in the Radial Bar Plot (left) reveals its relation to another data dimension in a histogram view (right) by highlighting this data in blue.

CHAPTER 5

Implementation and System Integration

The 2D Radial View with its different visualization and interaction techniques (see Chapter 4) has been implemented as an extension, i.e., as a plugin, for the software framework *Visplore* [30]. Visplore is a visual analytics software (see Section 2.3) developed at the *VRVis Research Center* [31], a computer science research company located in Vienna, Austria.

Just like the majority of Visplore's source code, all software components of the 2D Radial View are written in the programming language *C++*. The open graphics library *OpenGL* [36] is used for the final rendering at interactive refresh rates of all visualizations. Graphical user interface components make use of the *GIMP Toolkit GTK+* [92]. As all these technologies provide cross-platform compatibility, the software can be used on a variety of operating systems such as *Microsoft Windows* or *GNU/Linux*.

The following sections will give information about the technical background of the Visplore software framework and discuss important methods used to implement, integrate, and thereby realize the previously introduced 2D Radial View.

5.1 Extending the System

Visplore can easily be extended functionally, as it is a modern visual analytics framework that fulfills the requirements discussed in Section 2.3. New data importers and exporters, new computations and algorithms, as well as new views on the data can easily be added to the existing system. This functionality is provided by the software concept of *plugins*. Upon the program's startup, all available plugins and their corresponding dependencies are loaded. Depending on which plugins have been found on the local filesystem, i.e., plugins that have explicitly been deployed to the end-user, the system offers a different set of features to the user.

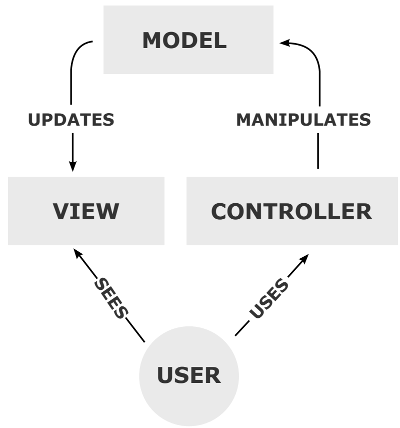


Figure 5.1: An illustration of the possible interactions between the three components of the Model-View-Controller architecture and its user.

Considering that data analysts may not need every available plugin to solve their desired tasks, the 2D Radial View has been implemented as a plugin for the software framework Visplore. The individual software components of this plugin are discussed below.

Architecture of the 2D Radial View Plugin

The 2D Radial View plugin consists of three major software parts. The partition into exactly three components is inspired by the component splitting proposed by the *Model-View-Controller (MVC) architectural pattern* [32]. In order to increase flexibility and reuse, this implemented software pattern decouples the internal representation of information, i.e., the *Model*, from its representation, i.e., the *View*, and the ways a user can control or manipulate the model, i.e., the *Controller*. Figure 5.1 illustrates the interactions between these components with respect to the user. This implemented software architecture poses several advantages when building an interactive visualization system such as the 2D Radial View. For example, due to the separation of concerns, different visualizations can be created based on the same information instead of having to recreate, e.g., calculate or extract, the information multiple times.

The components of the 2D Radial View plugin comply with the naming conventions used within the Visplore framework and thus are named *View*, *Backend*, and *Frontend*. Figure 5.2 gives an overview of these software components and the overall integration of the plugin into the Visplore system. As the component named "View" matches the "Model" entity in the terminology of the Model-View-Controller pattern, each component is discussed in more detail in the following:

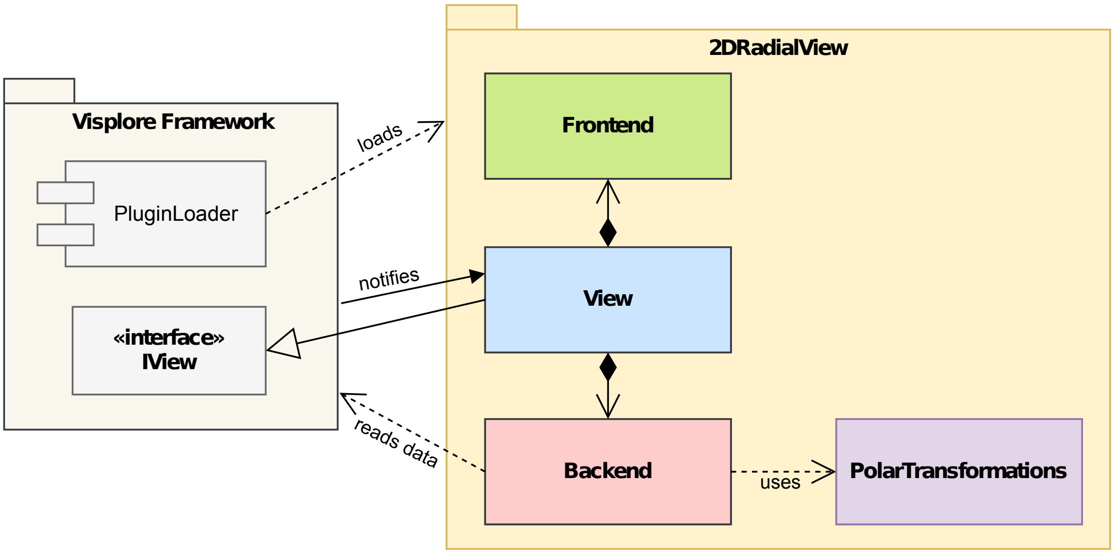


Figure 5.2: An overview of the components of the 2D Radial View plugin and its interaction with the Visplore Framework. The framework loads the 2D Radial View as a plugin upon startup and sends notifications to the View, which consists of a Frontend and a Backend. The Backend class reads data from the framework and uses the PolarTransformations class to map and render the visualization. The Frontend is responsible for handling user interactions.

The *View* is the central part of the 2D Radial View as it connects with, and reacts to notifications it receives from the Visplore system. The View is responsible for maintaining the current states and settings of the Frontend and the Backend. Whenever an event occurs that invalidates the Frontend's or Backend's state, an update is invoked on this/these component(s). This behavior matches the *Model* entity shown in Figure 5.1.

The *Backend* is responsible for generating the visualization, i.e., a view on the data in MVC terminology. The Backend uses information from the View to map data, and to compose and render the actual visualization. Thus, the Backend is also responsible for calculating aggregations or other intermediate results, which are essential during this visualization step. The Backend receives different kinds of notifications from the View whenever the visualization or parts of it need to be updated and reacts accordingly.

The *Frontend* acts as the controller, which accepts user input and forwards it to the View. Its tasks include providing different graphical user interface widgets that allow the user to manipulate settings, as well as monitoring and forwarding any mouse movements and events that take place inside the visualization.

Encapsulation of Coordinate System Transformations

As the 2D Radial View is designed as an advanced polar plot, its underlying conceptional coordinate system is the polar coordinate system. However, all rendering commands and mouse inputs operate within Cartesian coordinates. In order to minimize code duplication, I bundled all the functionality for the needed coordinate system transformations in the *PolarTransformations* helper class. As illustrated in Figure 5.2, its methods are used by the backend of the 2D Radial View. By organizing this functionality in a separate class, it can easily be reused and provides a central point for platform specific code optimizations or other improvements in the future.

5.2 Performance and Scalability Mechanisms

An important part of this thesis was the creation of a high-performance and highly-scalable implementation of the designed 2D Radial View. Explorative tasks, which play a key role in the explorative data analysis process, require frequent changes of the view on the data. These frequent changes may be triggered by a continuous user interaction such as the exploration of data ranges via a graphical user-interface slider. [78] The (partial) result of such a *dynamic query* upon the data has to be generated and displayed in under 100 milliseconds [86] to ensure a smooth and - more importantly - efficient workflow. Efficient in this context means reducing the risk of losing, i.e., skipping, interesting results during any continuous interaction [78].

As Piringer et al. [78] point out, the challenges are to keep the application responsive at all times while providing sufficient visual feedback. Further, the application must scale to very large data sets and multiple views. In order to address these challenges, the 2D Radial View implements a multi-threading architecture and layering system [78], which is especially designed for the interactive visual exploration of big data. The different thereby applied performance and scalability mechanisms are discussed below.

Multi-threading and Early Thread Termination

As the graphical user interface should never lock up due to the computation and rendering of a visualization, the 2D Radial View implements the *active object design pattern* [63]. The design pattern decouples (by the use of threading) the invocation from the execution of actions thus introducing asynchronous concurrency.

Following this pattern, each instance of a 2D Radial View respectively consists of a main thread and a single dedicated visualization thread. The 2D Radial View's main thread serves as an event loop, which is responsible for receiving and handling user requests or external events, whereas computation-time costly operations are delegated and executed in the separate thread. Figure 5.3 illustrates this implemented multi-threading architecture and its inter-thread and system communication.

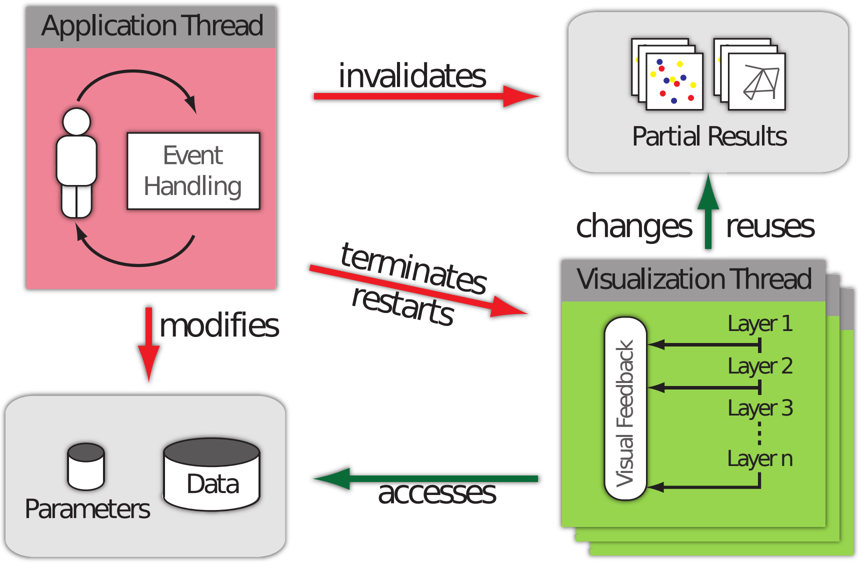


Figure 5.3: A schematic of a multi-threading architecture for an interactive visual data exploration following the active object design pattern. The event handling loop can accept new user input, while the image generation task is executed in the separate visualization thread. Reprint from Piringer et al. [78]

A central thread state object is used to store the desired state change from the main thread thus allowing the visualization thread to perform inexpensive queries on this attribute. The visualization thread queries from time to time if it is expected to continue with the ongoing execution or if it should terminate early (and thus restart) due to an event, which invalidated any further results of the computation. These checks (mainly) take place during the mapping and rendering of individual data entries. This querying happens at a high frequency as this checking and possible early aborting directly influences the view's lowest achievable response time. As the visualization thread may be terminated before the final image is displayed, this software paradigm is referred to as *Early Thread Termination* [78].

As explained in Section 4.2, the 2D Radial View uses semantic layers [78] for composing a final visualization. Separating the final image into passes that can be processed independently is a key concept of the implemented architecture. This way, some visual feedback can be presented to the user early. The layers are rendered incrementally back to front, i.e., using the painter's algorithm, in order to achieve the correct transparency effect when blending them together. A finished layer is blended and displayed as soon as it becomes available in the visualization thread. This presenting of partial visual results during a continuous user interaction is part of the Early Thread Termination paradigm.

Layer Caching

The chosen processing order of layers reflects the likelihood of change from top to bottom during the typical visual analysis process. The grid, when enabled and shown, is the lowest of all layers followed by the layer for all data items, context, selection, and superfocus. These layers are followed by the user interface components for the currently active interaction mode. Consequently, many situations in the visual analysis process require only the topmost layer or layers to be updated in order to result in a new valid visualization. For that reason, the 2D Radial View applies the principle of *layer caching* to further enhance performance, i.e., by reducing rendering times.

The grid and the four main visualization layers are cached in textures, which store *snapshots* of the OpenGL viewport during the rendering process. I decided not to cache lightweight visualization layers such as selection boundary indications as these can be rendered very fast anyway. The taking of snapshots occurs once a complete rendering of the visualization can finish without any interruption, i.e., an early thread termination event. If any layer of the visualization is invalidated by an event, a re-rendering of the 2D Radial View is requested by the main thread as the final visualization as such is not valid anymore. Depending on the layer that was invalidated, snapshots up to the layer below the one that was invalidated are reused to represent the information below this layer. Figure 5.3 illustrates how the main thread invalidates partial results and how the visualization thread reuses them. This mechanism speeds up the rendering process as a viewport-filling texture can be rendered much faster than mapping and rendering the respective layer.

Layer Preview Rendering

In addition to the previously discussed early thread termination and layer caching mechanisms, the 2D Radial View implements the rendering of *previews*. A preview is an approximate or incomplete visualization of a layer, which therefore can be rendered very fast. The approximation is usually a rendering of a small subset, i.e., a small fixed number of randomly sampled data entries, of the data for a specific visualization layer. Additionally, the visualization quality may be reduced, e.g., drawing boxes instead of points, to speed up the needed rendering time even further.

Previews are rendered in situations where the data base for the visualization is expected to change more frequently than the estimated time it would take to render and display the visualization of all of the involved data entries. Such situations include, for example, the user changing the size of the view, or the user changing the mapping for an axis of the visualization. Thanks to this preview rendering, the 2D Radial View gives the user valuable feedback regarding how a final visualization might look during such redraw-intensive interactions. Stolper et al. [90] named such a workflow *progressive visual analytics*, as it “enables an analyst to inspect partial results of an algorithm as they become available and interact with the algorithm to prioritize subspaces of interest”.

Previews, when rendered, are always displayed to the user only as an intermediate step before the complete and hence much more time consuming rendering is started if the thread has not been requested to stop in the meantime by an early thread termination event. Figure 5.4 shows a generated preview during, and the final image after a redraw-intensive operation.

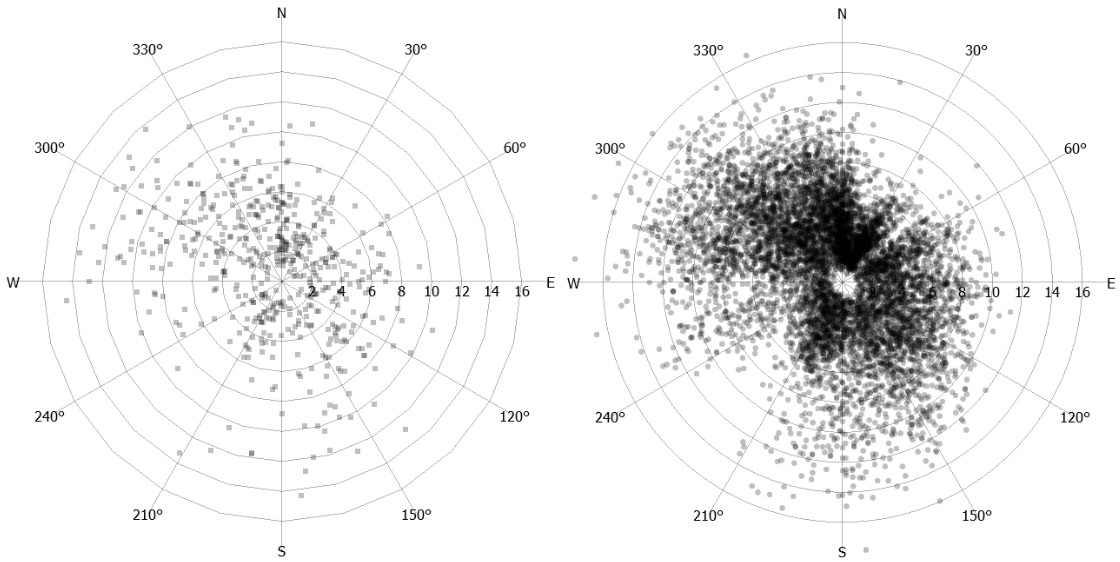


Figure 5.4: During redraw-intensive interactions, the 2D Radial View provides the user with previews of the final visualization by rendering temporary approximations of expensive visualization layers. The left side shows such a preview where a reduced number of entities is rendered and the precision of the visualization's geometry (e.g., tessellation of the data grid) is reduced to maintain interactive frame rates. The subsequent final image is shown on the right.

5.3 Implementing Circular Data Characteristics

Another important element of this thesis was the implementation of the circular data characteristics, which are an essential part of the 2D Radial View's interaction design, into an existing software framework. Many tasks that the new 2D Radial View plugin has to handle turned out to work correctly by applying just small modifications to existing concepts within the framework, whereas other tasks turned out to be much more challenging to implement. Hence, two very important and representative tasks with their concrete software solutions are discussed below.

Selections for Circular Data

Due to the fact that the used software framework did not provide any built-in support for a circular data type, it also did not offer a way for storing and handling selections on such data. A selection for this kind of data needs to be aware of the discontinuity at the polar axis for angular coordinates. Hence, it was necessary for me to extend the framework by developing and implementing a new type of selection that correctly handles this special data type characteristic.

The used software system already offered a mechanism to select data based on intervals on linear data types. This mechanism uses a semantic description, e.g., an interval, as a discriminator for a selection upon data values. Such a semantic description of a selection is called a *selection component*. A selection component can combine two or more selection discriminators by the use of binary operations to form more powerful semantics. As a selection component is a semantic description, it can be visualized to and interacted by the user as shown in Section 4.3. When the system evaluates a selection component, it results in a binary decision for each value upon if it is part of the selection or not.

In order to achieve an interval selection for circular data, I developed a new selection component. The component holds the semantic description for a *circular interval* on a single circular data dimension. This semantic description consists of the following three values: the lower bound of the interval, the upper bound of the interval, and the information if the interval describes a clockwise or counter-clockwise selection in the circular data space. These three values define a circular interval, which is easy and comprehensible for the user. Depending on this description, the algorithm decides if one or two linear intervals are needed to internally store the particular selection. If the circular selection interval contains the linear discontinuity the selection component uses two intervals, i.e., one interval to cover each side of the discontinuity, which are then logically combined with a binary OR-operation. This means that all data values are selected that either are contained in the first interval or in the second interval. In the case that the discontinuity is not part of the selection, the selection component uses only the first linear interval to store the complete interval and leaves the second interval empty. By implementing the algorithm in this way, I was able to use existing well-established mechanisms to model circular intervals.

As the 2D Radial View is a polar plot and thus two-dimensional, also a two-dimensional interval selection is necessary for an effective data analysis, as discussed in Section 4.3. The two-dimensional version uses the presented principles from above, but is extended by the information of the linear interval of the second, i.e., the radial, data dimension. The selection component adds this interval to the internal discriminator logic with a binary AND-operator. This results in the following logic: A data entry (e) that is in the radial interval (ri) and in either the first (ai_1) or second interval (ai_2) of the the angular dimension is a selected entry ($e_{selected}$). Equation 5.1 annotates this decision logic.

$$e_{selected} = (e \in ai_1 \vee e \in ai_2) \wedge e \in ri \tag{5.1}$$

Categorizations for Circular Data

As previously discussed in Section 4.3, the 2D Radial View enables the user to perform one-dimensional categorizations on the view's currently assigned angular data dimensions. This is executed by setting category boundaries at certain angular values and thereby defining intervals for each category. The used software framework did already offer a way to create, store, and manage such data categorizations, which were based on such value-based boundaries. However, the existing implementation was not capable of handling an interval that includes the discontinuity located at the polar axis. Independently of the number of categories, there may exist exactly one interval that includes the discontinuity, i.e., an interval that includes the jump in linearity from 2π to 0. The only exception to this is a category boundary that is placed exactly at the value 0. This results in exactly one interval that starts at 0 and exactly one interval that ends at 2π. Hence, I extended the existing categorization mechanism to support circular categorizations. Similar to the selection described above, this newly implemented type of categorization correctly handles the circular closure characteristic of a circular data dimension by detecting a jump and handling all software-internal intervals correctly.

CHAPTER 6

Evaluation

This chapter evaluates the proposed design by performing a case study that is driven by real-world scenarios and data. The study thereby demonstrates the design's value for data analysts in the field of wind energy production. As proposed by Sedlmair et al. [85], the solution is evaluated with real users. The case study is followed by a scalability and performance analysis to evaluate the implementation's applicability to large numbers of records.

6.1 Case Study

The following case study uses the proposed 2D Radial View together with the visual analysis framework it has been integrated into. The study is based on two large data sets. The goal of the performed visual analysis is to reveal patterns and/or interesting relations within these data sets.

The Data Sets

The two data sets originate from a national energy company, which is an energy producer and transporter. In recent years, the company has increased its focus on renewable energy sources, especially the production of wind energy, and on overcoming the challenges that lie within this field. The company's wind energy department contributed to this thesis by providing two wind energy production data sets for performing a real-world case study.

The first set has been recorded over a time span of twelve months and the second set over a time span of nine months. The sets thereby cover large parts of the years 2010, 2011, and 2012. The data is organized by date and time, which means that each value of every data dimension is assigned to a specific date and time of the day, i.e., its *time stamp*. The first data set contains 9504 time stamps with intervals of one hour and 31 data dimensions. The second data set contains 13154 time stamps with intervals of 30 minutes and 18 data dimensions.

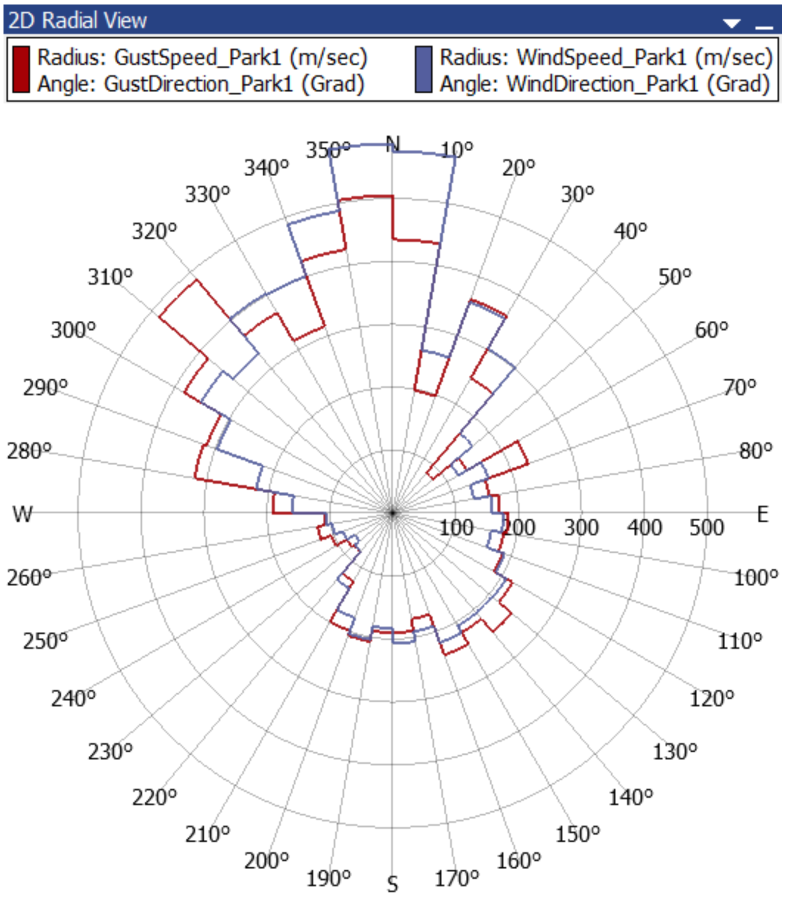


Figure 6.1: A 2D Radial View showing the angular distribution of winds (blue) and gusts (red) in a single visualization. The two generated line strips show that the angular distribution of winds and gusts are very similar. The plot reveals the two dominating directions present in the wind park, i.e., approximately zero and 150 degrees.

The data dimensions hold meteorological values such as air temperatures, relative air humidities, air pressures, wind and gust speeds, wind and gust directions, precipitation levels, or sun insolation levels for different geographical locations of wind parks. Both data sets contain values of the combined or individual total power output of these wind parks. The actual names of wind parks and geographical locations have been anonymized.

The data sets do not contain only values that have been recorded by sensors but also weather forecast data for these time stamps. These values have been generated by a prognosis model and algorithm based on recorded values. The algorithm for these generated values has not been provided and thus is not known to the analyst in this case study. By having forecast values along with their ground truth available, domain experts can analyze the quality, i.e., accuracy, of these forecasts.

Identifying Dominating Wind and Gust Directions

An important task for wind park operators is to identify dominating wind or gust directions for geographical locations inside wind farms. The *2D Radial View* is especially useful for such a task as it is able to display the circular closure of directional/circular data. To get a first overview, the data analyst assigns the wind and the gust direction data to the plot's angular dimension and the matching wind and gust speed data to the plot's radial dimension. The analyst uses the Radial Line Plot representation, as shown in Figure 6.1, where a quantitative distribution of direction values is revealed when the bin aggregation is set to the count of records. In contrast to numerical approaches or a Cartesian histogram, the complete and circular closed distribution of all directions is presented to the analyst. The plot shows that the quantitative distributions of the recorded wind and gust directions are very similar.

After adjusting the number of angular bins to a finer level of 36, the analyst concludes that there are two dominating directions. The identified dominating directions are pointing in almost the opposite direction, i.e., zero and 150 degrees. This insight strengthens the analyst's hypothesis that winds in this geographical region mostly flow in a preferred directional channel.

Understanding Total Power Output Influences

Estimating and controlling the total power output of a wind park is an important task for grid and network operators as the power grid must be maintained in a stable state although the network's load varies heavily due to changing levels of demand. The challenge is to satisfy the customers, which expect a constant supply of energy in a constantly changing environment. The demand and thus network load changes over the time of the day, the day of the week, the weather, the season of the year, and many other influencing factors. Understanding this influences enables system planers and operators to better balance the network under such varying network loads.

In the performed case study, the analyst's first approach is checking the hypothesis that the power output depends on temporal factors such as the day and night cycle or the season of the year. The gathered data, in combination with the software framework, enables the analyst to quickly verify such assumptions by using different specialized views and tools. Plotting the production per hour of the day in a bar chart, as show in Figure 6.2, reveals a systematic variance in the wind park's output with respect to the hour of the day. However, computing correlation measures between the time of day and the production shows that there is no simple linear or monotonic correlation between the two variables (Pearson = 0.016, Spearman = -0.002).

To investigate dependencies on other variables than temporal factors, the data analyst chooses a more unbiased approach for performing the regression analysis (i.e., finding one or more data dimensions that model the total power output of a wind park). The expert uses the techniques presented by Mühlbacher and Piringer [69] for building and validating regression models. The

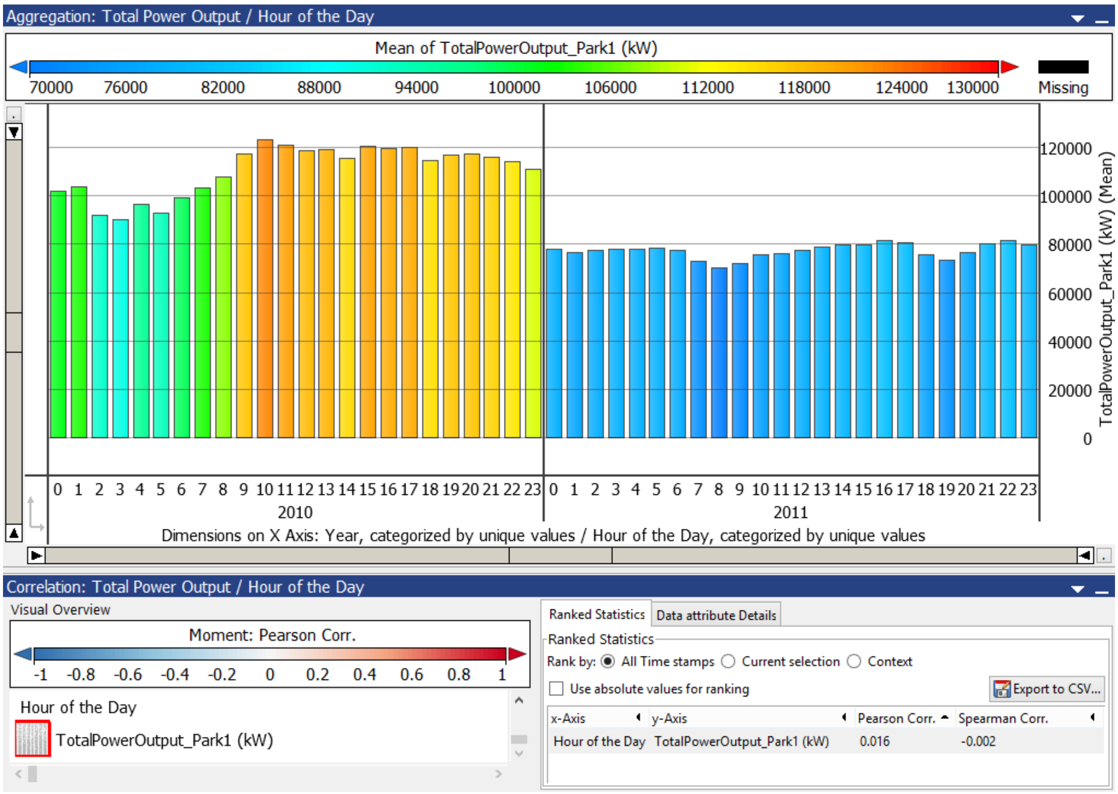


Figure 6.2: A bar plot (top) showing the mean production of a wind park (i.e., the power output) with respect to the hour of the day of the available recordings of the years 2010 and 2011. As the recordings of 2010 cover only the last 13 weeks of the year, the aggregated production means should not be compared with the ones of the year 2011. The Pearson correlation coefficient of 0.016 (bottom right) indicates that there is no linear correlation and a Spearman's rank correlation coefficient of -0.002 indicates that there is no monotonic correlation between the hour of the day and the total power output data dimension.

required views and features for this approach are available in the used visual analysis software framework. The user assigns the wind park's production as the target variable and the meteorological time series as inputs of the analysis. As depicted in Figure 6.3, the numerical indication (right) as well as the distribution of the target across each input dimension as box plots (left) is presented to the user. Due to the ranking, the coloring, and the numerical hints, the analyst uncovers that the gust speed presents the strongest influence. The gust speed is surprisingly higher ranked than the wind speed. For this wind park's power output, high gust speeds are more important than high wind speeds. This is the analyst's first gained insight.

The ranking in Figure 6.3 is based on R^2, i.e., the *coefficient of determination* of the piecewise linear regression of each input with the target. Details on this *goodness-of-fit* measure are discussed by Mühlbacher and Piringer [69]. As the R^2 value of 0.775 is high but not close to 1.0,

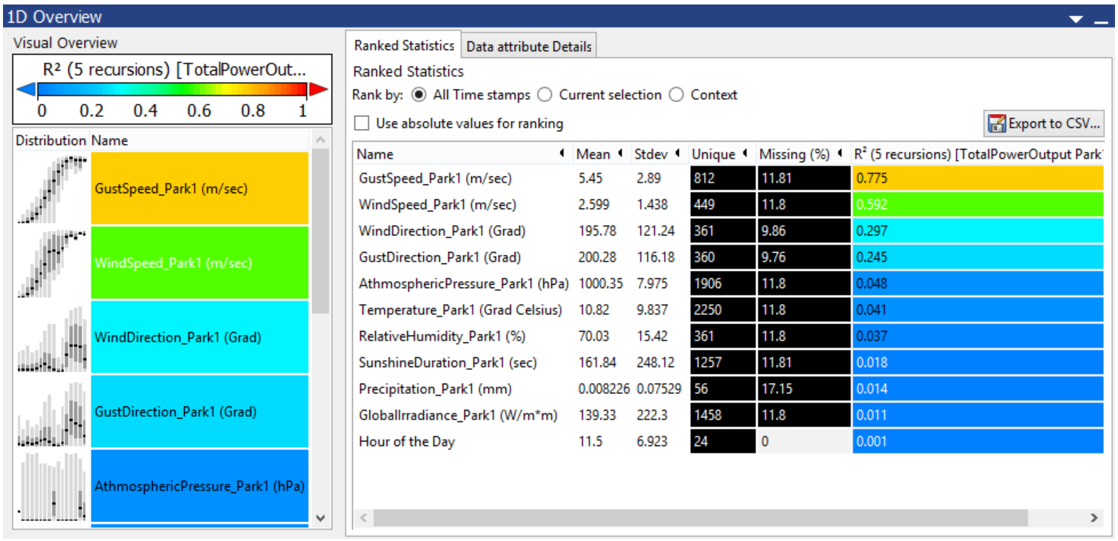


Figure 6.3: The visualization shows the correlation of different data dimensions with the total power output of a wind park. The data dimensions are ranked and colored based on their statistical correlation to the total power output data. The ranking reveals to the user that in the selected wind park gusts are a stronger influence on the total power output than average winds.

the expert concludes that other, more complex influences must exist. The displayed distributions of the wind and gust speed reveal that a range of gust or wind speeds produce an almost identical power output, especially in the upper and lower speed ranges. The analyst decides to perform a detailed analysis of *slow* (2.00 to 3.99 m/s), *medium* (4.00 to 5.99 m/s) and *high* (6.00 to 7.99 m/s) gust speeds with the goal of revealing these concealed influences.

For the task of selecting all data entries within the mentioned ranges of gust speeds, the expert chooses the selection feature within the 2D Radial View. He constructs a radial two-dimensional selection that covers the desired range of gust speeds across all possible angles (see Figure 6.4). When brushing through data, the partition-based ranking updates immediately based on the selected data. This allows the user to analyze the remaining variance. The hour of the day and the global irradiance are revealed as the strongest influences when studying slow gust speeds. The user is curious how the hour of the day influences the total production within this limited range of gust speeds. The expert opens a second 2D Radial View and gets creative by calculating new data that maps the hour of the day to the range of zero to 359 degrees. He assigns the newly created data together with the total power production to the 2D Radial View. Setting the 2D Radial View to the bar plotting mode with 24 bins results in a 24-hour clock visualization of aggregated power production values as shown in Figure 6.4. Such a visualization can be read and understood very easily and solves the problem of the circular closure of the time-of-the-day dimension. The final plot reveals that there is almost no power production caused by slow gusts during the day, but significant more output during the night and morning hours. In line with this discovered relationship, the global irradiance has an inverse correlation

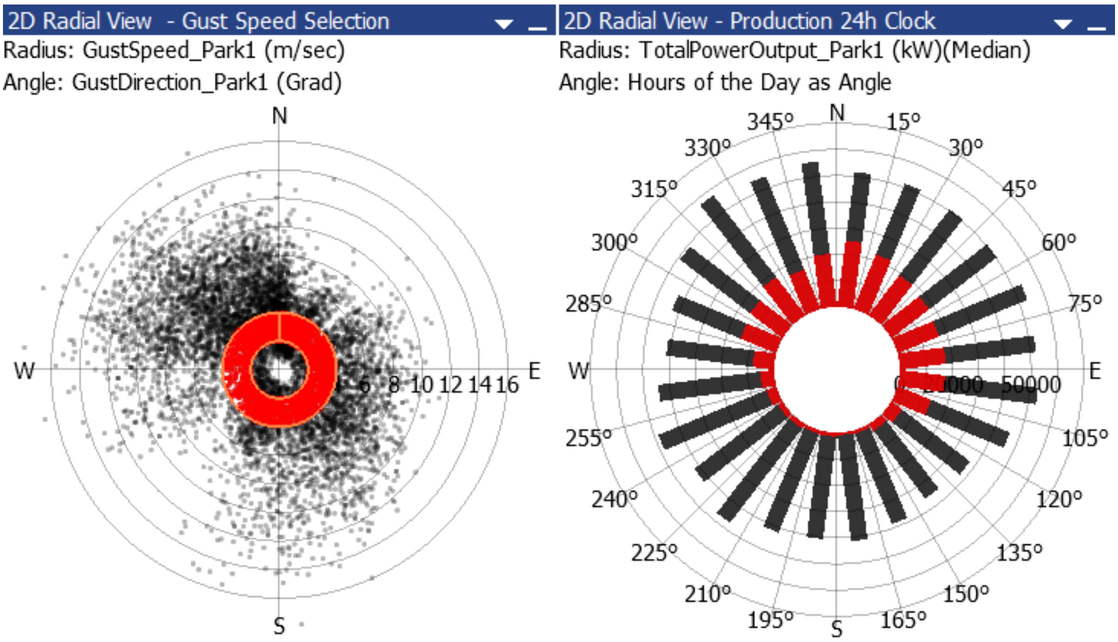


Figure 6.4: A two-dimensional radial scatter plot of gust speeds (left) is used to select slow (i.e., 2.00 to 3.99 m/s) gusts. A linked radial bar plot (right) shows the temporal influence of the hour of the day on the power output of a wind park for the selected data (red). The plot mimics a 24-hour clock and reveals that most power is produced during the night (north) and hardly any during the day (south).

with the total power output for slow gust speeds as there is almost no irradiance during the night. This is an interesting finding, which the expert decides to discuss with colleagues based on the generated visualization.

When the expert selects the gusts of medium speed, the results look very similar to the ones of slower speeds. More electrical power at the same wind speed is produced again during the night when the global irradiance is very low. The linked regression analysis view confirms this finding by listing the hour of the day and the global irradiance as the two greatest influences for the current selection.

When analyzing high speed gusts with the above-mentioned methods, however, the expert uncovers another interesting correlation. The data at hand suggests that, particularly for fast gust speeds, the direction seems to be highly relevant for the total power production (see top of Figure 6.5). To investigate this in detail, the analyst uses a radial bar plot as shown at the bottom of Figure 6.5. The expert uses the visual helper ring with its snapping feature to compare the total power production's median values of different directions. At the same speeds gusts of different directions produce very different power outputs. In the direction of 190 degrees the

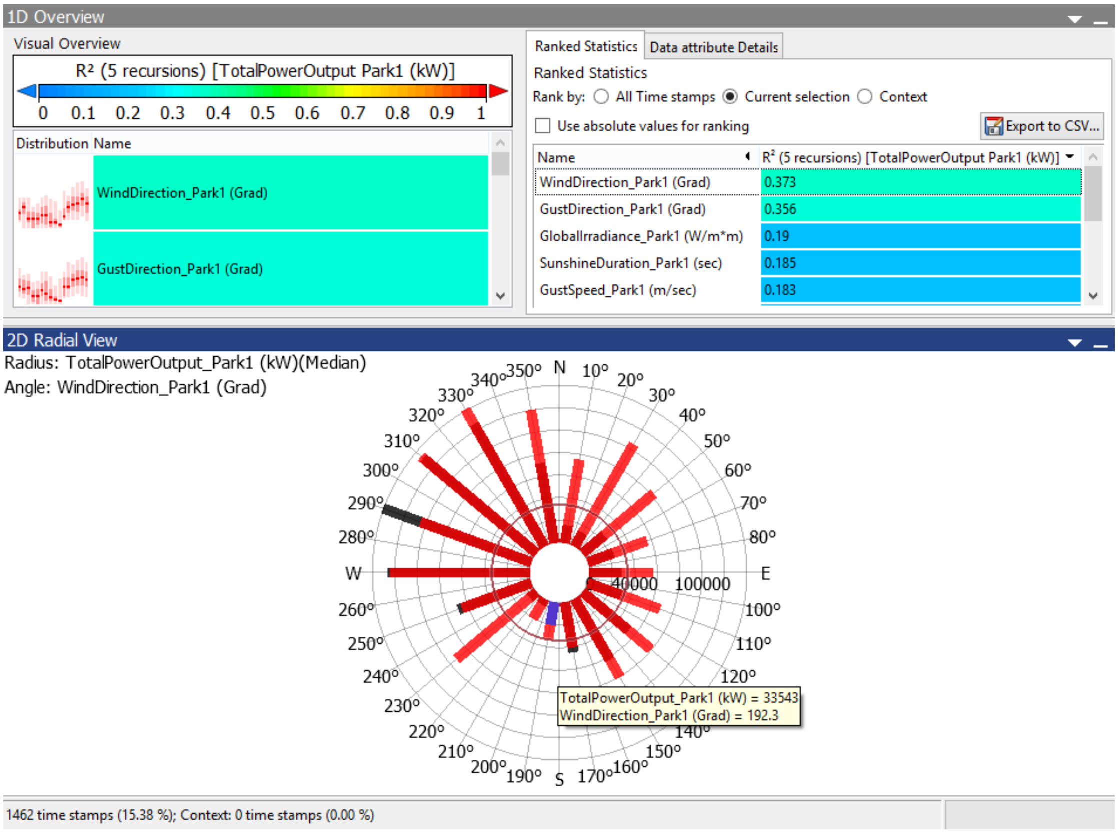


Figure 6.5: When analyzing high gust speeds (via a data selection), the regression analysis view (top) shows an influence of the wind's and gust's direction for the total power output of a wind park. The 2D Radial View (bottom) is used to compare the median power production of different directions for the selected data (red bars). The visualization reveals that the same gust speeds produce very different power outputs depending on the direction of winds.

median power output is only 33 megawatts, whereas the median in the direction of 330 degrees is 141 megawatts (i.e., an increase of 327 %). The expert points out that this finding shall be discussed with colleagues as well.

Evaluating Weather Prognosis Data

In the field of wind energy production, weather prognosis models and their generated data are used to calculate temporal trends and to generate power output forecasts. These forecasts are used for planning (e.g., load balancing the network) and identifying problems (e.g., power shortages) ahead of time. The underlying weather models have to be as good as possible to produce accurate weather forecasts, which are used for planning the power production for the upcoming day(s). As weather forecasts are often acquired by network operators through third parties, it is

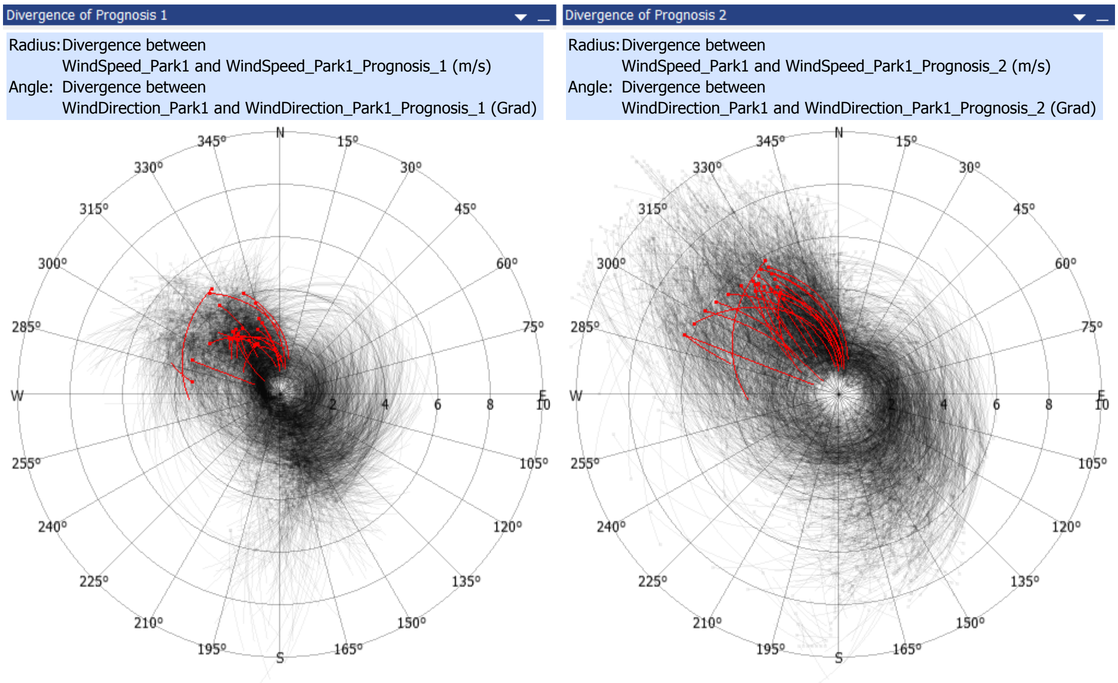


Figure 6.6: The Radial Divergence Plot is used by the analyst to visualize the differences between a wind's speed and direction prognosis and its actual measured data values (black arcs). Using two views side by side allows the user to easily compare *Prognosis 1* (left) against *Prognosis 2* (right). Selecting the 24 hourly intra-day time-stamps of a single day (red arcs) enables the user to perform a more detailed inspection and comparison.

an important task to select the best forecast(s) from a pool of different providers. The following case presents a visual evaluation and comparison of the prognosis data obtained from two different sources by looking at their correlation with the ground truth, i.e., the actual measurements.

For this evaluation task, the expert uses the proposed Radial Divergence Plot to analyze the point-wise differences between the predicted data and their corresponding actual measurements (see Figure 6.6). As the provided data sets contain the forecasts of different providers, the analyst is interested in understanding their strengths, weaknesses, and differences. By exploring the results of two forecasts side by side (e.g., *Prognosis 1* versus *Prognosis 2*), such differences can be visually analyzed by the expert. Long arcs, which visualize large errors (i.e., large drifts from the actual measured values), are mostly found radiating outwards in both plots. This indicates that the models have trouble at estimating the wind's speed. The chosen prognosis models produce similar looking results as both are overall able to predict the wind directions with little but the wind speeds with much greater error (shown by the black, semi-transparent arcs). For a more detailed analysis, and out of curiosity, the expert selects in a separate calender view the 24 time stamps of an individual day. The corresponding data is in turn highlighted in red in both plots as illustrated in Figure 6.6. By selecting or sweeping through individual days in this

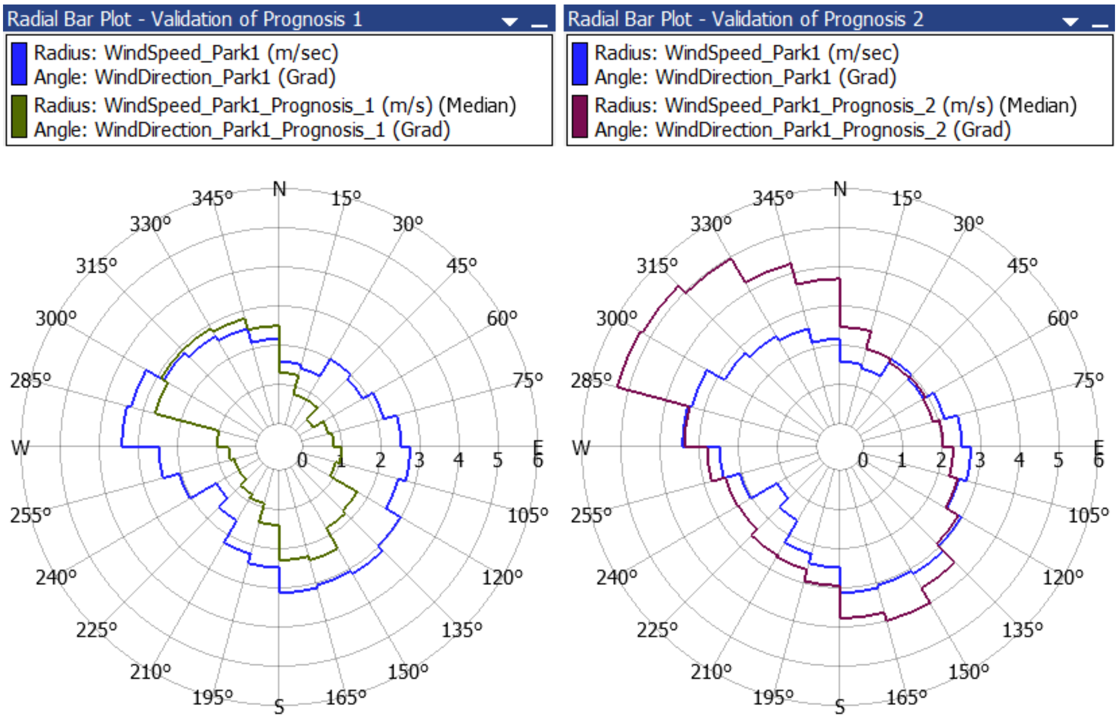


Figure 6.7: Two Radial Line Plots are used to visually compare *Prognosis 1* (left) against *Prognosis 2* (right) of wind speeds. The speed's median per direction of the actual measured values (blue) is plotted together with the prognosis' median (green and purple) in a single visualization. This approach helps users to spot differences in the angular as well as the radial dimension while comparing the prognosis models side by side.

way, the analyst studies how stable the wind predictions were within the short time frame of a single day. After inspecting multiple individual days, the expert concludes that the predictions are stable but contain a strong bias in the radial dimension.

To gain a better understanding of the prognosis' bias, the analyst uses the Radial Line Plot and assigns the prognosis and ground truth data to its dimensions. By changing the settings to display the median of the wind speed per direction, the aggregations become visible and easily comparable as they are embedded in a single visualization (see Figure 6.7). With an ideal prognosis model, the resulting median lines would overlap perfectly. However, as shown by the two visualizations, one model underestimates the wind speeds, whereas the other one tends to overestimate. *Prognosis 1* does a good job in estimating the speeds in the directions with higher recorded speeds but performs worse in the other directions. *Prognosis 2*, on the other hand, estimates the speeds for a wide range of directions correctly. However, *Prognosis 2* is too optimistic about the speeds in the directions ranging from 285 to 360 degrees. Looking solely at the plotted silhouettes (colored line strips), the expert supposes that both forecast models may have a hidden systematic angular bias of approximately fifteen degrees.

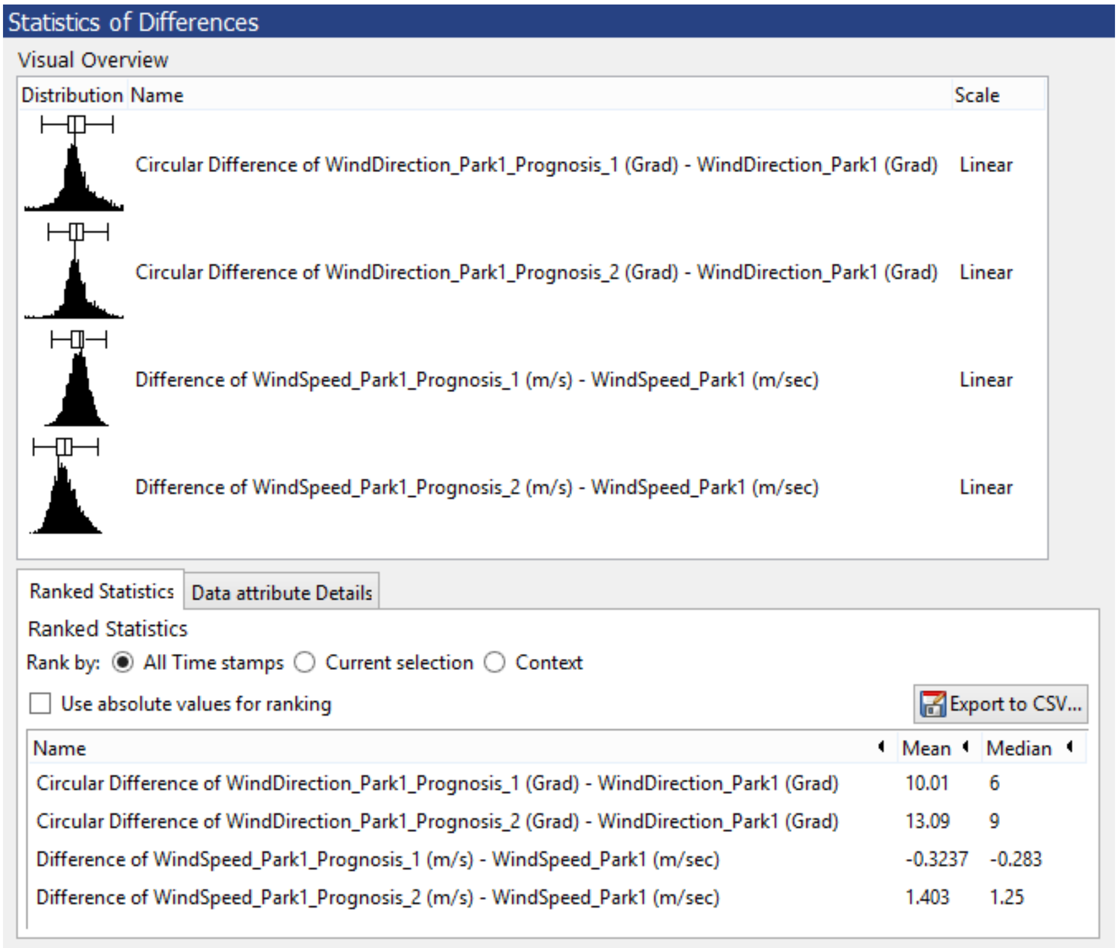


Figure 6.8: Box plots and histograms (top) visualize the distribution of the circular differences between the actual measured wind directions and its predicted values. The statistical median and mean values (bottom right) help the user to reason about the quality of the two prognosis models.

To investigate this more deeply, the expert calculates the circular point-wise differences between the prognosis data and the actual measured data. Plotting the histograms and box plots for these computed differences, shown in Figure 6.8, gives the expert a visual cue about the prognosis' bias towards a specific direction, i.e., the distribution's skewness. The statistical median and mean values of the calculated circular differences of wind directions verify the data analyst's hypothesis of a systematic angular drift. *Prognosis 1* shows a drift of about positive six to ten degrees, whereas *Prognosis 2* shows a drift of about positive nine to thirteen degrees. The median error of estimating the wind speed of *Prognosis 1* is only 0.283 meters per second or 1.02 kilometers per hour. The expert points out that this is an excellent value for a weather prognosis model. *Prognosis 2*, on the other hand, produces four times larger errors when estimating wind speeds. *Prognosis 1* is the expert's favored prognosis model in this comparison as wind speeds are among the strongest influences for the total power output of a wind park.

With the gained insights, the domain expert is now able to talk to prognosis model programmers to investigate the reasons for his findings. By using the above described workflow and incorporating insights like the described ones, the domain expert can improve the forecasts in the field of wind energy production.

6.2 Scalability and Performance

The system used for developing and testing the *2D Radial View* is a laptop computer powered by an *INTEL Core i5 4210U* mobile CPU with three megabytes of cache operating at 1.7 up to 2.7 gigahertz. The two core, four thread CPU has access to eight gigabytes of *DDR3* main memory running at 800 megahertz. The used graphics card is the dedicated mid-range consumer mobile GPU *NVIDIA GeForce GTX 850M* with four gigabytes of graphics memory. All tests are performed running the 64-bit version of Microsoft's *Windows 8.1* operating system.

The data sets used in the case study contain approximately 10000 data points in every data dimension. The raw data files are each about two megabytes in size if stored in a comma separated value (CSV) text format, encoded in the *American Standard Code for Information Interchange* (ASCII) character encoding. Even using multiple simultaneously opened 2D Radial Views and triggering very redraw intensive operations (e.g., altering selections) for these views has no noticeable impact (i.e., degradation) on the system's performance. Interactive update and image refresh rates (i.e., greater than 15 frames per second) are maintained at all times at the common desktop screen resolution of 1920x1080 pixels. No stalling of the system is observable during or between successive user operations.

To test the implemented visualization and interaction techniques' applicability to large numbers of records, a much bigger *performance data set* was created by extrapolating the data to cover a much larger time span. The extrapolated data contains hourly data records starting from the year 2011 up to the year 2022. The data set holds approximately 105000 data points in every data dimension, which is more than ten times the amount of the data used in the presented real-world case study. The implemented *2D Radial View* is capable of loading and visualizing such large amounts of data without any problems. As a performance test, four 2D Radial Views are stressed by performing brushing or highlighting operations through the data, which trigger a redraw of all four views simultaneously. The tests show that there is no noticeable drop of the system's performance and that interactions are carried out in the expected way. However, when using binning, which requires to calculate aggregates based on up to 105000 values, a small delay between an interaction and plotting the final visualization becomes visible to the trained eye. In these situations, interactions are not negatively affected by this small delay as all calculations are decoupled from the interaction and visualization methods.

The visualization which suffers the biggest performance hit is the *Radial Divergence Plot*. When using large data sets (i.e., more than 10000 data points), the time it takes to update a visualization increases up to 250 milliseconds, causing significant lags when carrying out active interactions. Drawing a large number of bent lines results in a very high number of time-expensive *OpenGL* draw calls. The root cause for this issue is the way bent lines are implemented (i.e., drawing a sequence of straight lines) due to restricting the software to work with a legacy version of the *Open Graphics Library* (i.e., a OpenGL version less than 2.0). Higher versions of the OpenGL industry standard offer programmable stages of the graphics pipeline, which allow software developers to decrease the number of draw calls drastically and thus increase the software's performance.

6.3 Conclusion

In the presented case study, the expert was able to quickly identify dominating wind and gust directions using solely the *2D Radial View*. By combining the view with advanced features of the visual analysis framework, he gained insight over the most important total power output influences and performed a successful detailed analysis of different gust speeds. The performed study uncovered an unexpected correlation and thus triggered further investigations. As demonstrated by the case study, the proposed view design helps analysts to identify significant forecast errors in circular data such as wind or gust data. The within described workflow aids domain experts in their task of adjusting their prognosis model's parameters and reevaluating their results.

To conclude, the performed case study revealed that the proposed design aids analysts in real-world use cases in the field of wind energy production. It proved that the users adopted the *2D Radial View* with its many different visualization and interaction techniques into their company's workflows. Furthermore, it demonstrated a successful seamless integration into an existing visual analysis software framework. The proposed software design has shown to be applicable to very large data sets on consumer hardware thus making it a valuable addition to a visual analysis framework.

CHAPTER 7

Design Reflection

This chapter explains the design process used for the 2D Radial Plot. The chapter reflects upon the expert's feedback and explains the lessons learned from this feedback. It finishes by discussing the user's adoption of the developed software.

7.1 Design Process

The design process of this thesis was kicked off by a meeting with domain experts in the field of wind energy production. While talking to these experts about their tasks, they were encouraged to try to verbalize a specific need, i.e., one real-world problem they face. In some respects, the task clarity, i.e., "how precisely a task is defined" [85], was a bit fuzzy in the beginning of the process. However, data was available thus making the task suitable for an explorative design process. The experts agreed to an explorative approach spanning over the period of approximately one year.

The process was organized in an iterative manner: Step one was talking to domain experts. Step two was creating mockups and/or implementing features in a prototype. Step three was conducting a presentation and feedback session that marked the beginning of the next iteration. Such a collaboration between visualization researchers and domain experts with a clear distinction of roles is mandatory according to the practical guidance for performing design studies by Sedlmair et al. [85].

7.2 Expert Feedback

During feedback sessions with domain experts, proposed interactions and visualizations were refined based on new findings, or rejected based on their usefulness rating. The experts and the visualization researchers thus refined their common understanding of the task(s), and how they can be addressed by data visualization. New visualizations and interaction ideas were proposed and scheduled for a follow up presentation and discussion. The insights gathered during these valuable expert feedback sessions shall be discussed below.

Domain experts rejected designs that appeared to them as overly exotic or uncommon. Some techniques, which are common in the field of information visualization, have shown to be uncommon in their domain specific processes. The experts rejected, for example, the idea of an angular heatmap visualization, which encodes additional data information in the color attribute. Another rejected technique was the usage of stacked bars to encode additional data dimensions in the radial bar diagram. As the experts were not familiar with such techniques, they found them to be either too complicated to use or to explain when presenting findings to colleagues and/or management.

In the beginning of the design process, designs were agreed upon when discussed, but rejected later when presented with a mockup or a prototype implementation. The lesson learned was to have mockups (e.g., printed on paper) ready for presentation and further clarification. Mockups and prototypes have been found to be a great way to clarify ideas and uncover misunderstandings early in the design process.

During the process the experts clearly rejected the initial idea of encoding many data dimensions in one very generic and powerful visualization. However, they welcomed the feature of linking different views together. They stated that they prefer specialized views on the data and use these views as part of their existing set of analysis tools. As stated by Sedlmair et al. [85], it "[...] does *not* require a novel algorithm or technique contribution. Instead, a proposed visualization design is often a well justified combination of existing techniques".

The Radial Divergence Plot, presented in Section 4.2, has been reported to be especially useful for gaining an overview of the total dynamic of circular data. However, the users pointed out that it is used mainly as a passive view on the data and is hardly ever interacted with. The lessons learned were: Firstly, not every visualization technique is required to provide active interaction techniques even when embedded in a software system that usually provides such functionality. Secondly, designers should try to clarify upfront if any active user interaction shall be made available to the user to avoid unnecessary implementation efforts.

I would like to point out that the experts did not only reject and evaluate the researcher's proposals but also endorsed and proposed their own ideas. For example, the visualization and interaction technique for creating a circular categorization, as shown in Section 4.3, has been proposed as they identified a use case and need for this task. Furthermore, they expressed their wish for a better way to visually compare radial magnitudes in a polar plot, which resulted in the design of the visual helper ring as shown in Section 4.2.

The above mentioned expert's feedback acted as key input in the process and heavily directed its outcome. I realized that existing domain processes and established company workflows steer the design process and must not be ignored or undervalued. The ultimate goal is to acquire the user's acceptance of the proposed design and thus make them use the built software artifact in their daily work.

7.3 User Adoption

After approximately one year of research and development, the proposed *2D Radial View* has been accepted for productive usage in a national power production and transportation company. At the time of this writing, it has been reported that the software is in use by experts whenever performing tasks on large sets of circular data. As pointed out by Kerren et al. [57], the "adoption by the target users is valuable evidence that the system has met its goals". Furthermore, the software is also occasionally used by the researchers and data analysts of a computer science research company.

CHAPTER 8

Discussion and Future Work

The research in this thesis resulted in one possible design and implementation of radial diagrams for the visual analysis of wind energy production data. This chapter discusses limitations of the proposed design and implemented software solution and gives an outlook onto future work.

8.1 Angular Data Formats

The *2D Radial View's* angular data dimension is currently limited to the data range of circular degrees (i.e., numerical values ranging from zero to 360). There is no implicit support for other circular or cyclic formats such as commonly used temporal values (e.g., hours of the day or months of the year). Formats of other value ranges can easily be converted to the required format by applying a computation that transfers the source values into the angular data range. However, the problem of this approach is that the user has to perform this step manually and the displayed labels, tooltips, and other cues do not match the source's original data range. Future work should include a (semi-)automatic detection of the assigned data format and apply the correct labeling. If the labeling can not be derived from the source it should be possible to configure it. However, no manual data formatting steps should be required when executing common use cases.

8.2 Visual Cluttering

Visualizing a very large number of data in a limited visualization space without applying any kind of data aggregation or abstraction results in visual cluttering. The produced visual clutter may hide important information and/or mislead the data analyst. The implemented *Radial Scatter Plot* as well as the *Radial Divergence Plot* face the problem of visual cluttering when visualizing large numbers of records. Both visualizations employ a blending technique (i.e., blending

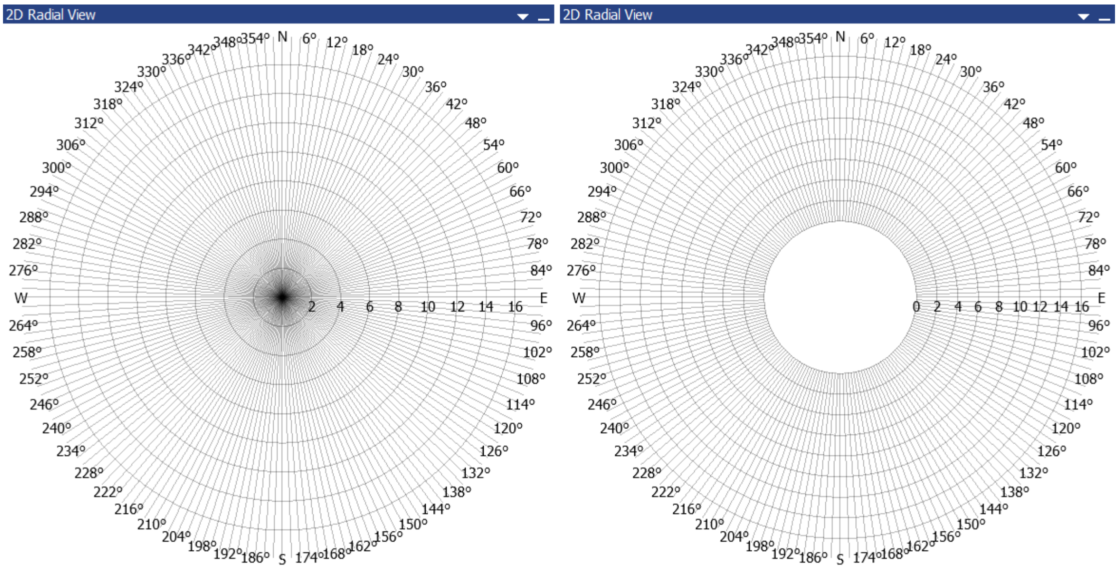


Figure 8.1: Drawing a large number of data grid lines radiating outwards from the center produces aliasing artifacts in the view's low resolution center (left). Shifting the resolution-critical parts of the visualization to a higher resolution area in the polar plot avoids these artifacts but wastes display space (right). This approach might be an unacceptable tradeoff for users as there do exist anti-aliasing solutions to the underlying problem.

of semi-transparent entities) to relax this visual issue in use cases with high overdraw. Future research should incorporate advanced decluttering solutions such as clustering techniques. Ellis and Dix [27] provide an in-depth taxonomy of clutter reduction for information visualization as guidance. From my perspective, especially for the Radial Divergence Plot decluttering techniques for graph layouts (e.g., edge bundling [40]) should be considered. Natalia and Gennady Andrienko [3, 4] proposed methods to generalize and aggregate trajectories of movement data. Their solutions could be applied to the arcs of the Radial Divergence Plot when the arcs are interpreted as spatial and temporal trajectories.

8.3 Frequency Aliasing

Drawing a polar data grid with a high number of radial subdivisions results in aliasing artifacts as shown in Figure 8.1. These spatial-aliasing artifacts in the plot's center are caused by a too low vertical and horizontal sampling resolution with respect to the frequency of the lines in the drawing region. The point-sampled lines form a characteristic Moiré pattern, which is unpleasant to the user's eye. This visual issue is addressed in this thesis by allowing the user to define a radial offset for the center of the visualization as discussed in Section 4.2. However, shifting the critical parts of the visualization to a higher resolution area in the polar plot does not solve the underlying aliasing problem. To achieve a higher image quality, common spatial anti-aliasing techniques such as *supersampling anti-aliasing* (SSAA) [15, 37] or *multi sample anti-aliasing* (MSAA) [2], or more complex post-processing methods such as *enhanced subpixel morphological anti-aliasing* (SMAA) [49] or *fast approximate anti-aliasing* (FXAA) [65] could be applied.

8.4 Polar versus Cartesian Selections

The implemented data selection methods, which operate in the two-dimensional polar coordinate system, can currently not be modified with interaction handles in other Cartesian plots within the system, and vice versa. Selections must be adjusted in the source system or numerically in a separate widget. When working with different plot types, this limitation leads to an interruption of the user's workflow. Due to the circular closure of the angular dimension, unifying the two distinct selection types presents a non-trivial problem. It would be interesting to study how automatic or semi-automatic (i.e., user triggered) conversions could solve this issue and enhance the user's experience when working in such a mixed plot setup.

8.5 Advanced Coloring

All the proposed radial visualization techniques mainly use color as a visual attribute to distinguish between different (sub-)sets of data (e.g., to distinguish the current selection from all data entries). Other visualizations in the software framework, however, make use of advanced coloring techniques to transport other, or more detailed information to the user's eye. Usually color transfer functions are used to visualize another data dimension within the same layout. Although a heatmap visualization was explicitly rejected by the users (see Section 7.2), there exist many other interesting applications of color to research. For instance, the *Radial Divergence Plot* could benefit by coloring all arcs with respect to their arc's length or other attributes (e.g., air temperatures) to analyze correlations and patterns. However, when cluttering occurs the problem of correctly blending the overlapping colors together must be solved. Future work could investigate in what use cases such advanced coloring is feasible to be combined or exchanged with the current coloring technique.

8.6 Sandboxes versus Workflows

The used visual analysis software framework serves as a sandbox environment for analyzing and solving a wide range of often very different problems. As such, it offers the user many different visualizations, interactions, options, and computations. A specific domain (e.g., energy production) with its (known) use cases might employ only a small subset of all these features. Executing a use case, which is common to the domain, currently requires the user to search through all of these features and to execute many steps manually. Many of these steps could be automated if the use case is known at the start. Additionally, options and decisions could be preset to useful defaults for this domain and/or scenario. In recent years, visual analysis software frameworks addressed these issues by designing *task-tailored dashboards* [68, 70, 77] prior to deploying the software to the users. The dashboards restrict flexibility to "allow users to address tasks without extensive training of the visualization software" [70]. Future work could integrate a stripped-down (i.e., tailored) version of the 2D Radial View into such task-tailored dashboards and evaluate if the workflows associated with the experts' tasks are improved by these changes.

CHAPTER 9

Conclusion

This thesis presented the design of visualization and interaction techniques to improve the visual analysis of wind energy production data. The research of the data and the shortcomings of existing systems have clearly shown a need for interactive radial diagrams as directional data such as recorded winds and gusts is often poorly visualized. The work characterized and abstracted the problems of visually analyzing wind data to the issues of handling circular data. The performed design study resulted in the proposal and implementation of visualizations and interactions that respect the circular closure and natural coherence of this special kind of data. The design process was guided by expert feedback and real-world data. It produced valuable insights of the tasks and workflows performed by data analysts. Useful features such as controlling the offset of the diagram's center, the visual helper ring, and the circular categorization emerged from such feedback. The results demonstrated the importance of early and often user feedback in this interdisciplinary field of study. The strong collaboration with visual analysis experts and an end-user (i.e., the data analyst) ensured a viable product with a high user acceptance.

It has been demonstrated by the case study that the proposed visualization and interaction techniques can be effectively used in practice in the field of wind energy production. However, thanks to its generic design, the proposed design is applicable to many other fields that deal with circular data. The thesis successfully demonstrated how to implement novel techniques into an existing software framework. The shared implementation details provided suggestions for handling common issues of big data when building responsive and interactive visualizations.

In conclusion, this work showed the design, implementation, and evaluation process for creating valuable radial diagrams for the visual analysis of wind energy production data. The generated results provided data analysts with better tools and software designers with extensive know-how to tackle the challenges of today's data driven world.

APPENDIX A

Coordinate Systems and Transformations

In the mathematical field of geometry, a *coordinate system* is defined as a system that specifies points using *coordinates*. Coordinates are a set of variables or list of numbers, noted as a n-tuple, needed to fix a geometric entity in a space, e.g., the Euclidean space. Depending on the problem, it is common practice to use a certain coordinate systems in which the problem can be described and/or solved most efficiently.

Today's graphics hardware, and specifically the libraries that talk to this hardware (e.g., OpenGL [36]), are operated by providing them geometric information and operations as input. Geometric information is usually handed over in *Cartesian coordinates* because display devices are designed and built to be efficiently addressable via this type of coordinates.

Two-dimensional radial shapes and structures that are visualized in the scope of this thesis are often addressed by or calculated with *polar coordinates* as they present to them a more natural data space. As graphics libraries either lack the support for non-cartesian coordinate systems or require the user to implement an interpretation for the supplied data, it is necessary to understand the Cartesian coordinate system as well as the polar coordinate system and how to convert coordinates from one system into the other.

A.1 Cartesian Coordinate System

The Cartesian coordinate system uses Cartesian coordinates to denote signed distances on perpendicular axes from an *origin*. The coordinates describe points in space uniquely, as the distances on all perpendicular axes use the same unit of length. Cartesian coordinates are usually donated by the 2-tupel of Equation A.1 in a two-dimensional space constructed by the axes x

and y and the 3-tupel of Equation A.2 in a three-dimensional space constructed by the axes x, y, and z. Both tupels are called a *point* and are often written in the form of a vector. The Cartesian coordinate system is the most widely used system for describing two- and three-dimensional entities and their relations.

$$(x, y) \tag{A.1}$$

$$(x, y, z) \tag{A.2}$$

A.2 Polar Coordinate System

The polar coordinate system is a two-dimensional coordinate system which describes a point on a plane by a distance from a fixed point, called the *pole*, and an angle from a fixed direction. The distance from the pole is usually referred to as the *radius* or *radial coordinate* and denoted by the letter r. The angle is called *angular coordinate*, *polar angle*, or *azimuth* and denoted by the Greek letter φ.

Following this definitions, a point in a polar coordinate system is described by its *polar coordinates* encoded with the 2-tupel of Equation A.3.

$$(r, \varphi) \tag{A.3}$$

A.3 Coordinate System Transformations

In order to convert points from one coordinate system to another one, a mathematical transformation from a source to target system as well as the inverse operation, from target to source, is needed. The two, for this thesis relevant, transformations are performed as follows:

Given a two-dimensional position in polar coordinates defined by radius r and angle φ, applying the following transformation transfers the position into the Cartesian coordinate system:

$$x = r * \cos \varphi \tag{A.4}$$

and

$$y = r * \sin \varphi \tag{A.5}$$

Given a two-dimensional position in Cartesian coordinates defined by an x and y value, applying the following transformation transfers the position into the polar coordinate system:

$$r = \sqrt{x^2 + y^2} \tag{A.6}$$

A case distinction is needed for the calculation of the angle φ:

$$\varphi = \begin{cases} \arctan\left(\frac{y}{x}\right), & x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi, & x < 0\,, y \geq 0 \\ \arctan\left(\frac{y}{x}\right) - \pi, & x < 0\,, y < 0 \\ +\frac{\pi}{2}, & x = 0\,, y > 0 \\ -\frac{\pi}{2}, & x = 0\,, y < 0 \\ \text{undefined} & x = 0\,, y = 0 \end{cases} \tag{A.7}$$

As can be seen in Equation A.7, if x and y both equal zero the angle can not be computed. This definition is due to the fact that otherwise an infinite number of points in polar coordinates, i.e., with radius $r = 0$ and any arbitrary angle φ, would map to the origin of the Cartesian coordinate system. Therefore, these transformations only fulfill the requirements for bijective projections if the set of Cartesian input coordinates is limited to the one shown in Formula A.8.

$$\{(x, y) \mid x, y \in \mathbb{R}_{\neq 0}\} \tag{A.8}$$

Many programming languages offer a special function or method that takes care of the distinction of cases and returns the correct value of φ for arbitrary values of x and y. In the programming language *C++*, which was used for this thesis, this feature is provided by the $atan2()$ function, which is available through the *cmath* header file as part of C++'s *numerics* library.

Figure A.1 illustrates the geometric relationship between polar and Cartesian coordinates by indicating the rectangle that can be constructed when visualizing r, φ, x and y.

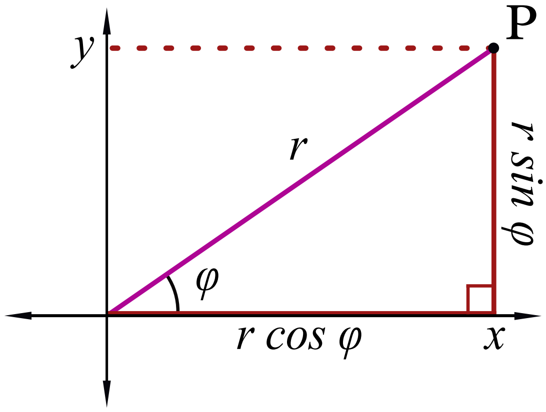


Figure A.1: The plot illustrates the geometrical relationship of point P between its polar (r, φ) and its Cartesian (x, y) coordinates. The image highlights the trigonometrical properties of the coordinate system transformation.

Bibliography

[1] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):47–60, 2008.

[2] K. Akeley. Reality engine graphics. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, pages 109–116, New York, NY, USA, 1993. ACM.

[3] G. Andrienko and N. Andrienko. Spatio-temporal aggregation for visual analysis of movements. *2008 IEEE Symposium on Visual Analytics Science and Technology*, pages 51–58, October 2008.

[4] N. Andrienko and G. Andrienko. Spatial generalization and aggregation of massive movement data. *IEEE Transactions on Visualization and Computer Graphics*, 17(2):205–19, March 2011.

[5] American Wind Energy Association. Wind 101: the basics of wind energy. `http://www.awea.org/Resources/Content.aspx?ItemNumber=900&navItemNumber=740`, 2013. [Online; accessed 6-January-2015].

[6] M. E. Baron. A Note on the Historical Development of Logic Diagrams: Leibniz, Euler and Venn. *The Mathematical Gazette*, 53(384):113–125, 1969.

[7] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 1998.

[8] W. Berger and H. Piringer. Peek brush: A high-speed lightweight ad-hoc selection for multiple coordinated views. In *2010 14th International Conference Information Visualisation*, pages 140–145, July 2010.

[9] E. Bertini, P. Hertzog, and D. Lalanne. Spiralview: Towards security policies assessment through visual correlation of network resources with evolution of alarms. In *2007 IEEE Symposium on Visual Analytics Science and Technology*, pages 139–146, Oct 2007.

[10] J. Blyth. On the application of wind power to the generation and storage of electricity. *Proceedings of The Philosophical Society of Glasgow*, pages 1–2, May 1888.

[11] A. Buja, J. A. McDonald, J. Michalak, and W. Stuetzle. Interactive data visualization using focusing and linking. In *Proceeding Visualization '91*, pages 156–163, Oct 1991.

[12] J. V. Carlis and J. A. Konstan. Interactive visualization of serial periodic data. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, UIST '98, pages 29–38, New York, NY, USA, 1998. ACM.

[13] M. J. Carpano. Automatic display of hierarchized graphs for computer-aided decision analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, 10(11):705–715, Nov 1980.

[14] S. Carpendale and A. Agarawala. PhylloTrees: Harnessing Nature's Phyllotactic Patterns for Tree Layout. *IEEE Symposium on Information Visualization*, pages p3–p3, 2004.

[15] E. Catmull. A hidden-surface algorithm with anti-aliasing. In *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '78, pages 6–11, New York, NY, USA, 1978. ACM.

[16] W. S. Cleveland and R. McGill. An experiment in graphical perception. *International Journal of Man-Machine Studies*, 25(5):491 – 500, 1986.

[17] A. Cockburn. Revisiting 2d vs 3d implications on spatial memory. In *Proceedings of the Fifth Conference on Australasian User Interface - Volume 28*, AUIC '04, pages 25–31, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[18] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, 41(1):2:1–2:31, January 2009.

[19] C. De Boor. A practical guide to splines. *Mathematics of Computation*, 1978.

[20] M. C. Ferreira de Oliveira and H. Levkowitz. From visual data exploration to visual data mining: a survey. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):378–394, July 2003.

[21] S. Diehl, F. Beck, and M. Burch. Uncovering strengths and weaknesses of radial visualizations—an empirical approach. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):935–942, 2010.

[22] A. J. Dore, M. Vieno, N. Fournier, K. J. Weston, and M. A. Sutton. Development of a new wind-rose for the British Isles using radiosonde data, and application to an atmospheric transport model. *Quarterly Journal of the Royal Meteorological Society*, 132(621):2769–2784, October 2006.

[23] P. Dragicevic and S. Huot. Spiraclock: A continuous and non-intrusive display for upcoming events. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '02, pages 604–605, New York, NY, USA, 2002. ACM.

[24] G. M. Draper, Y. Livnat, and R. F. Riesenfeld. A survey of radial methods for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(5):759–776, Sept 2009.

[25] G. M. Draper and R. F. Riesenfeld. Who votes for what? a visual query language for opinion data. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1197–1204, Nov 2008.

[26] G. Elber. Line illustrations $\in$ computer graphics. *The Visual Computer*, 11(6):290–296, Jun 1995.

[27] G. Ellis and A. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, Nov 2007.

[28] N. Elmqvist and P. Tsigas. A taxonomy of 3d occlusion management for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):1095–1109, Sept 2008.

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

[30] VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH. Visplore. `http://www.vrvis.at/projects/visplore`, 2015. [Online; accessed 24-March-2015].

[31] VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH. VRVis Forschungs GmbH. `http://www.vrvis.at/about`, 2015. [Online; accessed 24-March-2015].

[32] E. Gamma. *Design patterns: elements of reusable object-oriented software*. Pearson Education India, 1995.

[33] P. Gjinoll. Wind Energy Potential of Kosovo. Master's thesis, Vienna University of Technology, 2010.

[34] K. Grogg. Harvesting the wind: the physics of wind turbines. *Physics and Astronomy Comps Papers*, page 7, 2005.

[35] M. H. Gross, T. C. Sprenger, and J. Finger. Visualizing information on a sphere. In *Information Visualization, 1997. Proceedings., IEEE Symposium on*, pages 11–16, Oct 1997.

[36] Khronos Group. OpenGL - The Industry's Foundation for High Performance Graphics. `https://www.opengl.org`, 2015. [Online; accessed 24-March-2015].

[37] P. Haeberli and K. Akeley. The accumulation buffer: Hardware support for high-quality rendering. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '90, pages 309–318, New York, NY, USA, 1990. ACM.

[38] B. P. Hedrick, L. E. Zanno, D. G. Wolfe, and P. Dodson. The slothful claw: Osteology and taphonomy of nothronychus mckinleyi and n. graffami (dinosauria: Theropoda) and anatomical considerations for derived therizinosaurids. *PLOS ONE*, 10(6):1–49, 06 2015.

[39] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley. DNA visual and analytic data mining. In *Proceedings. Visualization '97 (Cat. No. 97CB36155)*, pages 437–441, Oct 1997.

[40] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, Sept 2006.

[41] F. Hussain, A. G. Hussin, and Y. Z. Zubairi. On development of spoke plot for circular variables. *Chiang Mai Journal of Science*, 37(3):369–376, 2010.

[42] F. Hussain, Y. Z. Zubairi, and A. G. Hussin. An alternative diagrammatical representation of wind data. In *Proceedings of the 12th WSEAS International Conference on Applied Mathematics*, pages 292–294. World Scientific and Engineering Academy and Society (WSEAS), 2007.

[43] Pixabay Public Domain Images. `http://pixabay.com/en/wind-farm-windmills-turbines-energy-538576/`, 2014. [Online; accessed 6-January-2015].

[44] Pixabay Public Domain Images. `http://pixabay.com/en/windr%C3%A4der-wind-park-see-energy-408596/`, 2014. [Online; accessed 6-January-2015].

[45] Tableau Software Inc. Visual Analysis Best Practices - Simple Techniques for Making Every Data Visualization Useful and Beautiful. `https://www.tableau.com/visual-analysis-best-practices`, 2018. [Online; accessed 31-August-2018].

[46] A. Inselberg and B. Dimsdale. Parallel coordinates for visualizing multi-dimensional geometry. In *Computer Graphics 1987*, pages 25–44. Springer, 1987.

[47] SBC Energy Institute. Leading the energy transition. `http://www.sbc.slb.com/SBCInstitute/Publications/~/media/Files/SBC%20Energy%20Institute/SBC%20Energy%20Institute_Wind_Factbook_May%202013.pdf`, 2013. [Online; accessed 6-January-2015].

[48] C. Jeong and A. Pang. Reconfigurable disc trees for visualizing large hierarchical information space. In *Proceedings IEEE Symposium on Information Visualization (Cat. No.98TB100258)*, pages 19–25, 149, Oct 1998.

[49] J. Jimenez, J. I. Echevarria, T. Sousa, and D. Gutierrez. SMAA: Enhanced subpixel morphological antialiasing. *Computer Graphics Forum*, 31(2pt1):355–364, 2012.

[50] D. A. Keim. Visual Techniques for Exploring Databases. *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining Tutorial Program*, 1997.

[51] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, Jan 2002.

[52] D. A. Keim, G. Andrienko, J. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. Visual analytics: Definition, process, and challenges. In *Information visualization*, pages 154–175. Springer, 2008.

[53] D. A. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann. Mastering the information age: Solving problems with visual analytics. *Eurographics Association*, 2010.

[54] D. A. Keim, F. Mansmann, J. Schneidewind, and T. Schreck. Monitoring Network Traffic with Radial Traffic Analyzer. *2006 IEEE Symposium On Visual Analytics And Technology*, pages 123–128, October 2006.

[55] D. A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler. Challenges in visual data analysis. In *Tenth International Conference on Information Visualisation (IV'06)*, pages 9–16, July 2006.

[56] D. A. Keim, F. Mansmann, A. Stoffel, and H. Ziegler. Visual Analytics. Technical report, University of Konstanz, Germany.

[57] A. Kerren, J. T. Stasko, J. Fekete, and C. North, editors. *Information Visualization*, volume 4950 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008.

[58] W. Kienreich, R. Wozelka, V. Sabol, and C. Seifert. Graph visualization using hierarchical edge routing and bundling. In *Proceedings of the 3rd International Eurovis Workshop on Visual Analytics (EuroVA). The Eurographics Association*, pages 1–5, 2012.

[59] N. Kirchner-Bossi, L. Prieto, R. García-Herrera, L. Carro-Calvo, and S. Salcedo-Sanz. Multi-decadal variability in a centennial reconstruction of daily wind. *Applied Energy*, 105:30–46, May 2013.

[60] K. W. Kolence and P. J. Kiviat. Software unit profiles & kiviat figures. *SIGMETRICS Perform. Eval. Rev.*, 2(3):2–12, September 1973.

[61] J. Lamping and R. Rao. The hyperbolic browser: A focus+context technique for visualizing large hierarchies. *Journal of Visual Languages & Computing*, 7(1):33–55, 1996.

[62] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 401–408, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[63] R. G. Lavender and D. C. Schmidt. Pattern languages of program design 2. chapter Active Object: An Object Behavioral Pattern for Concurrent Programming, pages 483–499. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1996.

[64] Y. Livnat, J. Agutter, S. Moon, and S. Foresti. Visual correlation for situational awareness. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005*, pages 95–102, Oct 2005.

[65] T. Lottes. FXAA. *NVIDIA white paper*, 2:8, February 2011.

[66] K. V. Mardia and P. E. Jupp. *Directional statistics*. Wiley Series in Probability and Statistics. Wiley, Chichester, 1999.

[67] R. McGill, J. W. Tukey, and W. A. Larsen. Variations of box plots. *The American Statistician*, 32(1):12–16, 1978.

[68] T. Mühlbacher, C. Arbesser, and H. Piringer. Statistical forecasting in the energy sector: Task analysis and lessons learned from deploying a dashboard solution. In *Short Paper in Proceedings of IEEE VIS 2015 (Visualization in Practice Track)*, 2015.

[69] T. Mühlbacher and H. Piringer. A partition-based framework for building and validating regression models. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1962–1971, Dec 2013.

[70] T. Mühlbacher and H. Piringer. Task-tailored dashboards: Lessons learned from deploying a visual analytics system. In *Proceedings of IEEE Vis 2014*, 2014.

[71] C. Möllinger and W. Ludwig. Flow visualization. `https://www.cg.tuwien.ac.at/courses/Visualisierung/2010-2011/Beispiel2/Moellinger_Ludwig/index.html`, 2011. [Online; accessed 28-August-2015].

[72] M. J. Mohammadi-Aragh and T. J. Jankun-Kelly. Moiretrees: Visualization and interaction for multi-hierarchical data. In *Proceedings of the Seventh Joint Eurographics / IEEE VGTC Conference on Visualization*, EUROVIS'05, pages 231–238, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.

[73] M. F. Morris. Kiviat graphs: Conventions and figures of merit. *SIGMETRICS Perform. Eval. Rev.*, 3(3):2–8, October 1974.

[74] T. Munzner. H3: laying out large directed graphs in 3d hyperbolic space. In *Proceedings of VIZ '97: Visualization Conference, Information Visualization Symposium and Parallel Rendering Symposium*, pages 2–10, Oct 1997.

[75] Foraker Labs – Provider of Usability & Web Design Services. Glossary » scientific visualization. `http://www.usabilityfirst.com/glossary/scientific-visualization/`, 2015. [Online; accessed 2-January-2015].

[76] offshoreWIND.biz. Mhi vestas 8 mw turbine breaks world record. `http://www.offshorewind.biz/2014/10/17/mhi-vestas-8-mw-turbine-breaks-world-record/`, 2014. [Online; accessed 5-January-2015].

[77] S. Pajer and H. Piringer. Challenges in creating task-tailored analytical dashboards for the automotive industry. In *Poster Proceedngs of IEEE Vis 2015 (VIS in Practice)*, 2015.

[78] H. Piringer, C. Tominski, P. Muigg, and W. Berger. A multi-threading architecture to support interactive visual exploration. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1113–20, 2009.

[79] W. Playfair. *The Statistical Breviary*. Wallis, London, 1801.

[80] M. G. Reggiani and F. E. Marchetti. A proposed method for representing hierarchies. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):2–8, Jan 1988.

[81] T. Rhyne, M. Tory, T. Munzner, M. Ward, C. Johnson, and D. H. Laidlaw. Information and scientific visualization: separate but equal or happy together at last. In *IEEE Visualization, 2003. VIS 2003.*, pages 611–614, Oct 2003.

[82] G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone trees: Animated 3d visualizations of hierarchical information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, pages 189–194, New York, NY, USA, 1991. ACM.

[83] M. J. Saary. Radar plots: a useful way for presenting multivariate health care data. *Journal of Clinical Epidemiology*, 61(4):311–317, April 2008.

[84] T. Saito, H. N. Miyamura, M. Yamamoto, H. Saito, Y. Hoshiya, and T. Kaseda. Two-tone pseudo coloring: Compact visualization for one-dimensional data. *Proceedings - IEEE Symposium on Information Visualization, INFO VIS*, pages 173–180, 2005.

[85] M. Sedlmair, M. Meyer, and T. Munzner. Design Study Methodology: Reflections from the Trenches and the Stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, December 2012.

[86] B. Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, 11(November):70–77, 1994.

[87] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[88] B. Shneiderman. Extreme visualization: squeezing a billion records into a million pixels. *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 3–12, 2008.

[89] J. Stasko and E. Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *IEEE Symposium on Information Visualization, 2000. INFOVIS 2000*, pages 57–65, Oct 2000.

[90] C. D. Stolper, A. Perer, and D. Gotz. Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1653–1662, Dec 2014.

[91] M. Tavanti and M. Lind. 2d vs 3d, implications on spatial memory. In *IEEE Symposium on Information Visualization, 2001. INFOVIS 2001*, pages 139–145, Oct 2001.

[92] The GTK+ Team. The GTK+ Project. `http://www.gtk.org`, 2015. [Online; accessed 24-March-2015].

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

[94] C. Tominski and H. Schumann. Enhanced interactive spiral display. *Proceedings of the Annual SIGRAD Conference, Special Theme: Interaction*, pages 53–64, November 2008.

[95] M. Tory and T. Möller. Human factors in visualization research. *IEEE Transactions on Visualization and Computer Graphics*, 10(1):72–84, Jan 2004.

[96] M. Tory and T. Möller. Rethinking visualization: A high-level taxonomy. In *IEEE Symposium on Information Visualization*, pages 151–158, 2004.

[97] J. W. Tukey. *Exploratory data analysis*. Addison-Wesley Publishing Co., Massachusetts, 1977.

[98] Alternative Energy Tutorials. Wind energy turbines. `http://www.alternative-energy-tutorials.com/wind-energy.html`, 2015. [Online; accessed 5-January-2015].

[99] M. O. Ward and B. N. Lipchak. A visualization tool for exploratory analysis of cyclic multivariate data. *Metrika*, 51(1):27–37, Jul 2000.

[100] M. Weber, M. Alexa, and W. Muller. Visualizing time-series on spirals. In *IEEE Symposium on Information Visualization, 2001. INFOVIS 2001*, pages 7–13, 2001.

[101] L. Wilkinson. *The Grammar of Graphics, Second Edition*. Springer Science+Business Media, Inc., 2006.

[102] J. Yang, M. O. Ward, and E. A. Rundensteiner. Interring: an interactive tool for visually navigating and manipulating hierarchical structures. In *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002*, pages 77–84, 2002.

[103] J. S. Yi, Y. A. Kang, J. Stasko, and J. A. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–31, 2007.

[104] G. Yngve, J. F. Brinkley, D. Cook, and L. G. Shapiro. A Model Browser for Biosimulation. *AMIA Annual Symposium Proceedings*, pages 836–840, 2007.

[105] W. Zeng, C. W. Fu, S. M. Arisona, A. Erath, and H. Qu. Visualizing mobility of public transportation system. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1833–1842, Dec 2014.