



Swansea University  
Prifysgol Abertawe



## Cronfa - Swansea University Open Access Repository

---

This is an author produced version of a paper published in:  
*IEEE Transactions on Visualization and Computer Graphics*

Cronfa URL for this paper:

<http://cronfa.swan.ac.uk/Record/cronfa38785>

---

### **Paper:**

Roberts, R., Laramée, R., Smith, G., Brookes, P. & DCruze, T. (2018). Smart Brushing for Parallel Coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 1-1.

<http://dx.doi.org/10.1109/TVCG.2018.2808969>

---

This item is brought to you by Swansea University. Any person downloading material is agreeing to abide by the terms of the repository licence. Copies of full text items may be used or reproduced in any format or medium, without prior permission for personal research or study, educational or non-commercial purposes only. The copyright for any work remains with the original author unless otherwise specified. The full-text must not be sold in any format or medium without the formal permission of the copyright holder.

Permission for multiple reproductions should be obtained from the original author.

Authors are personally responsible for adhering to copyright and publisher restrictions when uploading content to the repository.

<http://www.swansea.ac.uk/library/researchsupport/ris-support/>

# Smart Brushing for Parallel Coordinates

Richard C. Roberts, Robert S. Laramée, Gary A. Smith, Paul Brookes, Tony D’Cruze,

**Abstract**—The Parallel Coordinates plot is a popular tool for the visualization of high-dimensional data. One of the main challenges when using parallel coordinates is occlusion and overplotting resulting from large data sets. Brushing is a popular approach to address these challenges. Since its conception, limited improvements have been made to brushing both in the form of visual design and functional interaction. We present a set of novel, smart brushing techniques that enhance the standard interactive brushing of a parallel coordinates plot. We introduce two new interaction concepts: Higher-order, sketch-based brushing, and smart, data-driven brushing. Higher-order brushes support interactive, flexible, n-dimensional pattern searches involving an arbitrary number of dimensions. Smart, data-driven brushing provides interactive, real-time guidance to the user during the brushing process based on derived meta-data. In addition, we implement a selection of novel enhancements and user options that complement the two techniques as well as enhance the exploration and analytical ability of the user. We demonstrate the utility and evaluate the results using a case study with a large, high-dimensional, real-world telecommunication data set and we report domain expert feedback from the data suppliers.

**Index Terms**—Multivariate Visualization , Parallel Coordinates, Call Center, Glyph, Brushing, Interaction Techniques.

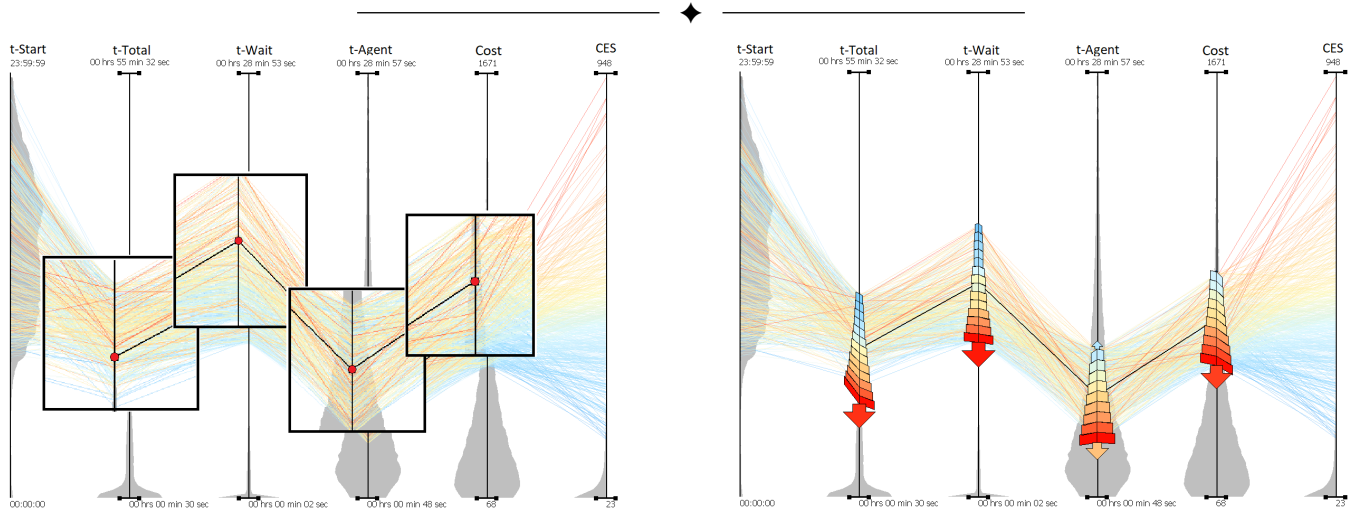


Fig. 1: These images demonstrate higher-order brushing for n-dimensional exploration of parallel coordinates. The left image indicates a series of snapshots from the sketch-based interaction of brush definition from left-to-right. The right image shows smart brushing incorporating the interval range histogram glyphs that provide interactive guidance to the user. Red dots indicate mouse-click positions.

## 1 INTRODUCTION

PARALLEL Coordinate plots (PCPs) are widely used to overcome the challenges associated with the visualization of high-dimensional data. The n-dimensional capabilities of the PCP enables complex relationships to be plotted with simplicity. However, the parallel plot is still subject to overplotting and occlusion if the number of polylines becomes too large. This challenge may be addressed through brushing and filtering techniques directly applied to the plot.

Current methods of brushing (summarized in Figure 2) enable users to query the PCP through direct data interaction or axis selection [1], [2] – however these methods are generally not supported by meta-data. Our methods involve sketching flexible, high-dimensional patterns in a chain-like sequence such that multi-dimensional patterns in the data can more easily be explored, discovered, and analyzed. We introduce interval range glyphs that provide real-time feedback to the user whilst querying the data using the sketch based interaction.

We propose a number of user interaction-based features to

integrate with PCPs in order to improve the search and discovery of n-dimensional patterns and relationships. We present a brush interaction which enables a smooth, more intuitive, sketch-based interaction whilst providing context-aware feedback to guide the user in their brush definition and placement. We define this brushing mechanism as higher-order brushing. For a visual demonstration of this see Figure 1 on page 1. The sketch element of the brush freely enables brushes to be applied across multiple axes quickly and naturally by drawing arbitrary patterns on top of the PCP. It is more intuitive than conventional brushing because it follows the patterns of the polylines themselves. Together with the glyph-based feedback, these methods compliment each other to facilitate quick and powerful exploration of large datasets.

Previous brushing techniques are limited in terms of intelligence. They offer little-to-no guidance on placement and definition. Any intelligence involved is based solely on the users own knowledge and perception of the data. In this paper, we use the phrase ‘Smart Brushing’ to encompass our ideas of both

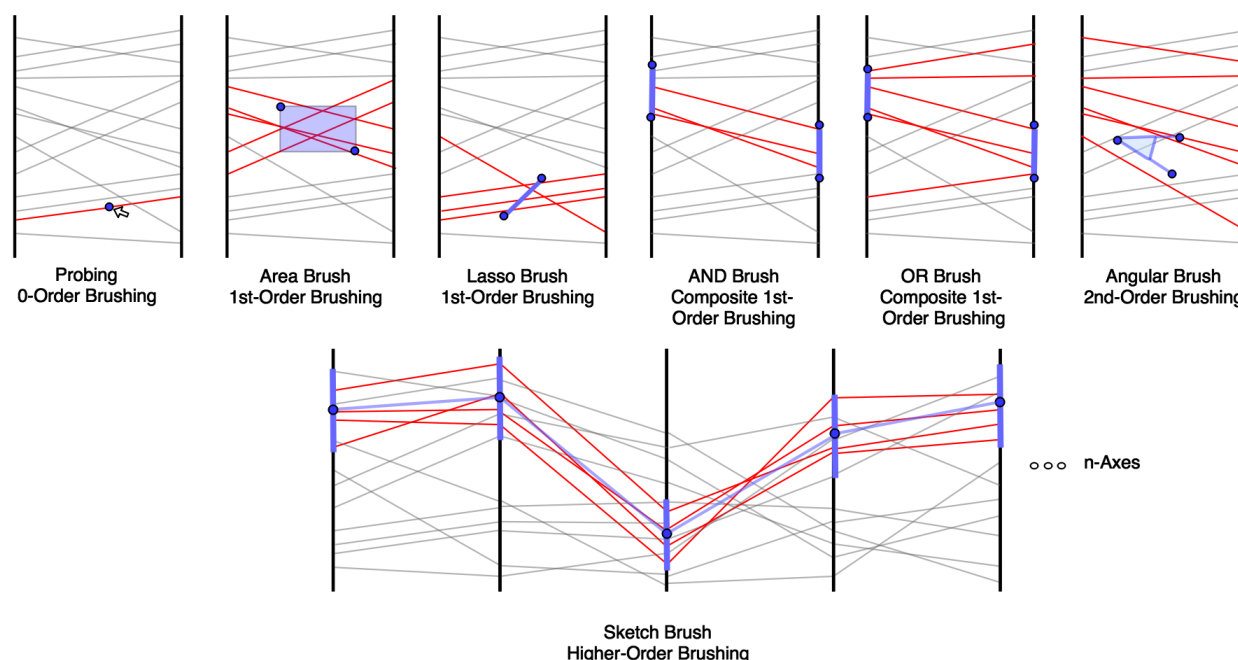


Fig. 2: This image is adapted and inspired by Raidou et al. [3] and summarizes the current State-of-the-Art brushing techniques. We propose the next step in these brushing interactions with higher-order brushing and smart, data-guided brushing.

higher-order brushing and smart brushing. In addition to the higher-order brushing techniques we present a modified, data-guided, smart brush that provides real-time feedback to the user, based on the current mouse pointer location and surrounding data. Modified, dynamic angular histograms [4] are integrated into brushes to display the high-frequency, n-dimensional trends during the interaction. This may highlight the divergent points or areas at which the data suddenly changes.

Smart, higher-order brushes can also be applied automatically based on the most densely populated axis sections or other properties of the data. This auto-complete feature can guide the user in interactive placement of brushes such that the most important high-dimensional trends can be observed. Such guidance is useful for new users of higher-order brushing or experienced users looking to speed up the interaction process. We also implement a smart, data-guided axis-scaling approach that magnifies a data-guided, user-specified interval brush range to the full height of the respective axis when a smart brush is applied. This maximizes the utility of the space available and is particularly valuable in dense areas of datasets. An auto-scale feature also enables the user to apply axis scaling such that a given number of standard deviations is depicted automatically around the brush-axis intersections. This provides the user more advanced n-dimensional search options for large datasets and enables direct filtering of outlying data. To demonstrate utility and evaluate the techniques, we undertake a multi-stage feedback process with domain expert testing using large, high-dimensional, real-world data from the telecommunications industry.

In summary, the contributions of this paper include;

- A novel, sketch-based approach to support higher-order brushing,
- Smart, data-guided brushing techniques that provide interactive guidance to the user,
- Data-guided, automatic application of brushes and axis scaling to accelerate the search for n-dimensional patterns,

- The demonstration and evaluation of the new brushing techniques in a real-world case study with our partners from the telecommunications industry.

In this paper we first discuss the related work on brushing in PCPs. We describe the limited interaction and features offered by current state-of-the-art brushing techniques. Then we present the concept and implementation of higher-order brushing. Following this, we describe the smart, data-guided brushing techniques with examples of utility and then provide evaluation with domain expert feedback involving a case study from the call center industry. We finish with conclusions and future work directions.

## 2 RELATED WORK

**Parallel Coordinates:** Though first published in 1885 [5], the parallel coordinates plot was popularized almost one hundred years later by Inselberg in 1980 [6], [7]. The design addresses n-dimensional data visualization challenges. Follow-up papers by Inselberg [8], [9], [10] further enhance the modern version of the visualization we use today. This n-dimensional technique has evolved into a popular area of research in the field [11]. The PCP approach has proved to be so popular that a book dedicated to the topic was published in 2009 [12]. Other multivariate forms of analysis have also been developed [13].

Gruendl et al. [14] use parallel coordinates as an alternative to time series plots. Tory et al. present a parallel coordinates interface for exploring volumetric datasets [15]. The span of parallel coordinate use cases is vast. Because of this, State-of-the-Art reports are useful to review the complete body of parallel coordinates literature and a number of closely related surveys have been published [1], [2], [16], [17], [18]. Most notably, Heinrich and Weiskopf provide a unique classification of parallel coordinates research [17].

**Interactive Brushing Techniques for Parallel Coordinates:** Brushing was introduced for the examination and dissection of multidimensional data on scatterplots [19]. Though the term

was later coined by Becker and Cleveland in 1987 [20] whereby individual data points could be selected by application of a ‘brush’ which can be used for activities such as highlighting, data manipulation, or labeling [20], [21].

We define brushing based on interaction with a single point as 0-order brushing. The term 0-order is inspired by the orders of approximation terminology conventionally used in science, engineering, and mathematics. Range brushes can be applied to individual axes [12]. We define this as a 1<sup>st</sup>-order brushing technique because it is based on interaction with two points. Typically, the application of a brush creates filters by which the data is processed [22]. The number of polylines rendered are reduced to minimize clutter and reveal patterns in the data. Raidou et al. present an orientation-enhanced brushing technique for PCPs [3]. In their paper, a visual description of State-of-the-Art brushing techniques is detailed ranging from probing (0-order brushing), to composite AND OR brushes (composite 1<sup>st</sup>-order brushing). See Figure 2 on page 2. Avidan and Avidan [23] define the queries ‘interval’, ‘pinch’ and ‘angle’ presented together with the use of boolean operators to form compound queries. A more detailed discussion of the design, motivation, and implementation is given in chapter 10 of Inselberg and Dimsdale [10]. These queries precede this paper.

Angular brushing enables data to be filtered based on the angle a polyline forms when intersecting an axis [23]. In other words, angular brushing is based on three points (and two edges). We call this 2<sup>nd</sup>-order brushing. The higher-order brushing techniques we introduce enable brushes to be applied using an arbitrary number of points e.g.  $n \geq 3$ . Combinations of brushes using logical operators can in theory enable n-dimensional filtering [23], [24], [25]. We define the result of this AND operation of brushes to be composite 1<sup>st</sup>-order brushing. The reason for this is because interaction with these brushes is based on manipulation of individual points. Interacting with one point has no effect on the others. From an interaction standpoint, they are effectively de-coupled. As a result, such interaction is slow and error prone. An interactive exploratory search for n-dimensional patterns using composite, 1<sup>st</sup>-order brushes is generally not supported.

Brushes are also applied such that focus is placed on a specific point or area of interest. Instead of a discrete context rendering, a decreasing polyline opacity value as a function of distance away from the point of interest is used [24], [26], [27].

Hierarchical brushes that use wavelets and clustering are sometimes used on large datasets [28], [29], [30], [31].

Axis scaling (also referred to as dimensional zooming) is often used as a means to focus the scope of analysis by modifying the upper and lower bounds of an axis [29], [32]. This dimensional zooming can be used to align axes to a common base [23]. In general these limited scaling methods offer basic, slow, manual axis control to the user. The smart axis-scaling approaches we describe enable fast, automatic n-dimensional axis scaling guided by the properties of the data.

Theron develops clever brushing techniques that enable the user to identify the variance between a selected polyline and the remaining polylines using color [33].

**Touch-based Interactions:** A large body of research explores the value of exploring visualizations on touch screen devices. Sadana and Stasko study the effectiveness of varying multi-touch and multi-handed gestures on standard graphs [34]. Similar research focuses more on the touch interaction than the visualization features [31]. Kosara identifies the time consuming

nature of applying brushes to parallel coordinates, and develops a multi-touch interaction for PCP brushing on a tablet [35].

Most similar to our method, Nielsen et al. develop scribble query, a touch brushing interaction for multivariate visualization [36]. The querying tool enables the user to select patterns with their finger and apply brushes similar to that of the pattern. Our work extends from this by implementing complete brush translations as well as brush component pinning. We also implement these features for use with a standard computer without the use of a touch screen.

**XmdvTool:** The most notable public domain Parallel Coordinates tool is XmdvTool. The software is used extensively for the study of multivariate data [24], [37], [38]. The current features enable users to explore multivariate data using a number of visual interaction methods. Primarily, parallel coordinates are used, however scatter-plots and glyphs can also be used to represent the same data.

In Ward [37], the hyperbox facilitates n-dimensional brushing through a net-like composite of individual 1<sup>st</sup>-order brushes that are manipulated on a point-by-point basis over the plot to filter polylines. The creation of a new hyperbox seems undefined. The hyperbox extends over each axis and defines the range of the brush on each axis. The upper and lower bounds of the brush ranges are manually edited, point-by-point, to apply a specific filter to the data. Points are dragged individually up and down on each axis. From an interaction point of view, manually placing, updating, an manipulation of individual points could be expedited and less error prone. Other parallel coordinates software such as XDAT use similar, limited methods for brushing [39].

Although the functionality of higher-order brushing is supported through the logical AND of individual point-based brushes, from an interaction point of view this is still 0-order. That’s because all interaction is based on updates to individual points and edges. Theoretically, a user could shift the entire brush upwards by a fixed distance by editing the position of each individual point along the brush the same given fixed distance. However, this type of interaction is slow, cumbersome, and error prone because the points are de-coupled. A better method would be to define any wave pattern in parallel coordinates space with a sketch-like interaction. The user should be able to shift the entire n-dimensional brush around in parallel coordinates space to facilitate interactive exploration of flexible n-dimensional patterns and relationships quickly and easily. This is one of the benefits of higher-order brushing.

We present a more efficient method of applying an n-dimensional brush filter through the implementation of a sketch-based brush that can quickly be applied and manipulated as both a complete n-dimensional brush pattern and individual brush points.

**Telecommunications Data:** The case study we incorporate for evaluation is based on real-world, n-dimensional data from the call center industry. The dataset contains millions of phone calls.

Whilst statistical analysis is common for data of this nature, the multidimensional aspect of the data challenges the potential for real valuable insight [40], [41]. A large body of research focuses on the processes involved in business operations [42], [43], [44]. Trends and patterns are observed more easily through visualization than through numerical representation of data. Roberts et al. have previously worked in this area by segmenting the event-based data into temporal nodes on a treemap [45]. Some trends can be observed through the temporal analysis, but other elements of the meta-data are less explored. Customer satisfaction is integral to the success of a business, and so research into the customer feedback

is highly valuable [40], [46], [47], [48]. The previous research generally takes a more quantitative approach without exploring potential visual designs and other aspects.

### 3 CALL CENTER DATA DESCRIPTION

The dataset throughout this paper stems from our industry partner from the telecommunications industry. The multivariate dataset is provided by QPC Ltd., our industry partner who specialize in call center data collection and analysis. We have been collaborating with QPC Ltd. since 2014 and have worked on visual analytics and information visualization projects together both academically and industrially.

The dataset we study contains almost 5 million calls over a period of one month. We provide new observations about the call center behavior that have been discovered using our smart brushing features. But first we provide the background of the data we use in the analysis.

As the sale of products and services transitions from bricks and mortar based locations to digital spaces, consumers spend a decreasing amount of direct contact time with companies. Call centers are often the first and only point of contact for customer interactions with businesses and so it is important that the process of contacting companies through the medium is streamlined and efficient. The customer experience of a call center journey can be recorded, but the process of analyzing the data presents many challenges.

The dataset contains a number of metrics that collectively quantify customer contact with a call center. Each call entering the call center can be broken down into four types of events: *IVR Events*, *Wait Events*, *Hold Events*, and *Agent Events*.

The IVR (Interactive Voice Response) is the routing system that the user navigates when they initially make contact with the call center. Wait events occur when the caller is placed in a queue, normally after an IVR event. Hold events occur when the caller is placed on hold by an agent. Agent events are instances of a customer talking directly to an agent. Each occurrence of each event type is recorded as well as their duration.

The estimated cost of the call is derived as well as two measures of customer satisfaction. The first being the Customer Effort Score (CES), a measure based on the accumulation of known detractor events in a call such as long queue times. The second is the Net Promoter Score (NPS), a direct customer feedback measure via a survey reflecting customer satisfaction. Only a small number of phone calls have the NPS metric recorded - around 8,000 of 230,000 calls in a 24 hour period. See Figure 11 a) on page 8 to see the full number of calls rendered.

The default axes in our figures are as follows:

- Start Time of each call ( $t$ -start)
- End Time of each call ( $t$ -end)
- IVR Duration - length of the time spent in the IVR ( $t$ -IVR)
- Wait Duration - length of time spend in the queue ( $t$ -wait)
- Agent Duration - length of time speaking with agent ( $t$ -agent)
- Hold Duration - length of time placed on hold by agent ( $t$ -hold)
- Total Call Duration -  $t$ -total
- Customer Effort Score (CES) - a derived metric of work put in by customer
- Net Promoter Score - actual feedback given by customer
- Cost - estimated cost of call

We use,  $t$ , to denote a specific time and,  $\mathbf{t}$ , to denote a duration of time throughout this paper. We use this real-world dataset to demonstrate the utility of our proposed visual designs as opposed to using synthetic data in an optimum use case to attest the utility of the visual design. Each figure that exemplifies the smart brushing features uses this dataset.

Whilst the dataset we use is large, the software is capable of handling the data on commodity machines. We developed the

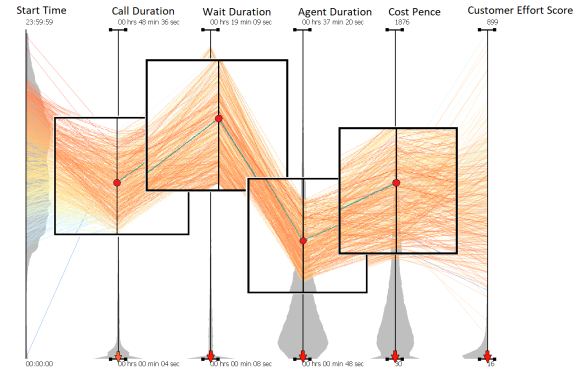


Fig. 3: Higher-Order Brush Definition: Each brush-axis intersection is placed with a single click whereby the interval range of the brush can be adjusted with the mouse wheel. In this figure, the brush-axis intersections are placed from left-to-right and a connecting line is drawn between each intersection point automatically to show the pattern represented by the higher-order brush. The entire n-dimensional brush can be moved around the PCP space interactively, as a unit, to enable fast exploration of arbitrary multi-dimensional patterns. Red dots indicate mouse-click positions.

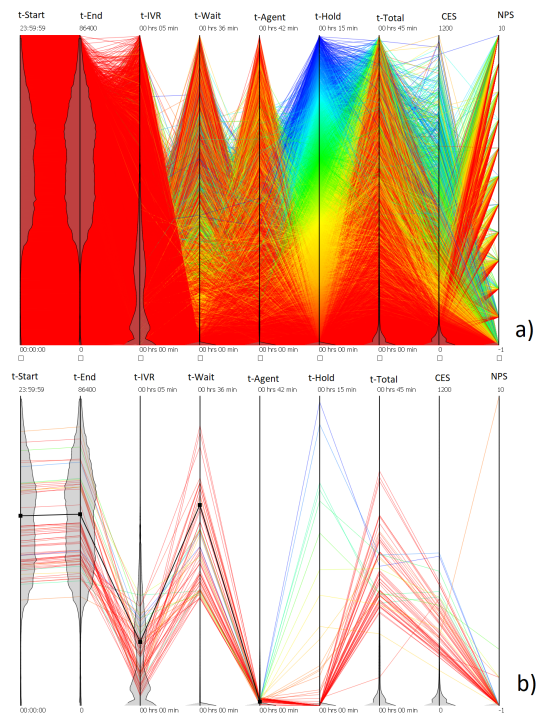


Fig. 4: Image a) shows a the parallel coordinates over-plotted with data. Image b) shows the higher-order brush applied to the plot, whereby abandoned calls are explored through the brushing of very low agent durations.

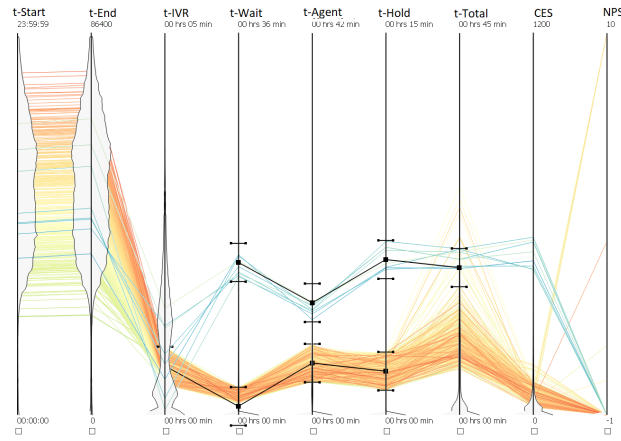


Fig. 5: Rigid brush translation in the PCP. Four connected brush components are translated both upwards, and across one axis whilst maintaining the brush pattern. Individual brush points can also be translated if the user wishes to adjust the pattern’s shape. This example of two different higher-order brushes shows the same pattern identifying both high  $t$ -wait with low agent duration calls, as well as low  $t$ -wait and high agent duration calls. This contrast shows the effect each pattern has on both the CES and NPS scoring metrics.

software in Qt C++ using OpenGL for the implementation. The primary desktop PC used for the development featured 12GB RAM, a 6th Gen i7, and a GTX 1070.

## 4 HIGHER-ORDER BRUSHING

### 4.1 Sketch-Based Brushing

Higher-order brushes enhance standard brushing by enabling the user to sketch arbitrary, multi-dimensional patterns with a natural sketch interaction. Previous high-dimensional brushes are used as a tool to filter and search for  $n$ -dimensional patterns in multivariate data. However, the application of these brushes is time consuming, inaccurate, and non-intuitive. This is because the interaction is based on individual points (0-order). Also the interaction with previous brushes based on points is generally based on click-and-drag operations moving up and down on the axis. This is non-intuitive because the brush pattern is orthogonal to the orientation of the polylines themselves. From an interaction point of view, the user should be able to interrogate the  $n$ -dimensional data by stating, “I would like to see all the multivariate patterns that look like this.” At which point they sketch a polyline, typically from left to right, that represents the pattern for which they are searching. We have developed a sketch-based brush that can easily be applied using an arbitrary  $n$ -dimensional click interaction and manipulated with precision so that the user can maximize their exploration and analysis of the data.

The sketch-based brush is applied using a sequence of connected mouse clicks across the PCP on each axis at the desired brush-axis intersection location resulting in a flexible  $n$ -dimensional filtering pattern. The interval range on each axis is applied automatically. We use the term ‘brush interval range’ to describe the upper and lower boundaries of the brush as it intersects each individual axis in the PCP. Scrolling the mouse wheel adjusts the brush range interval, or boundaries which define how far from the polyline-axis intersection point is considered in focus. See Figure 3 on page 4. Each box frame in Figure 3 represents an instance of time

during the interaction. During the interaction and before a brush-axis intersection point is placed, a preview outline of the brush size is rendered at the mouse point to inform the user how large the next interval range is. Once the brush-axis intersection point is placed, the data outside the range of the brush is filtered and rendered as context. This enables the user to more intelligently define the  $n$ -dimensional brush through a natural sketch-based interaction. We use focus + context rendering methods so that polylines outside the  $n$ -dimensional brush boundaries are still viewed as context. A user option can turn this off such that out of focus polylines are not rendered. This is useful in cases where the context of polylines is so large that it does not contribute to user understanding.

**Example 1: Abandoned Call Identification:** We demonstrate the utility of the higher-order brushing feature in Figure 4 on page 4. In this example we identify and explore a new category of abandoned calls in the call center by applying the higher-order brush to very low agent duration, whereby any callers would not have spoken to an agent for long enough to solve any problem they might have ( $t < 10$  seconds). Previously QPC Ltd. only identified abandoned calls that never reached an agent ( $t = 0$ ). Each call associated with the call center is represented by a polyline in the PCP. The left image shows the completely occluded parallel coordinates image contrasted against the right image which shows the higher-order brush applied to a range of axes. The  $t$ -start and  $t$ -end axes select the peak times in which calls are made, the  $t$ -IVR brush is placed to cover the main distribution of calls, and the  $t$ -wait brush selects higher wait time callers. The final brush covers very low  $t$ -agent ( $t$ -agent), selecting abandoned calls. The result of this shows a subset of calls that typically would be considered negative due to the high wait times and abandoned agent interactions. The IVR and start/end time brush components filter out anomalous polylines so that all data in focus is representative of typical abandoned callers. In this example, calls are quickly identified out of over 200,000 as a new category if abandoned caller.

### 4.2 Interactive Translation

After the initial  $n$ -dimensional brush is defined by the user, the whole group or the individual brush points can be updated through a simple click-and-drag translation. A user option enables a natural, click-and-drag interaction to translate the entire  $n$ -dimensional brush in any direction as a rigid object, maintaining shape consistency in the brush pattern sketched. The translation works across axes which means that the exact  $n$ -dimensional pattern can be applied to any combination of axes. See Figure 5 on page 5. This rigid  $n$ -dimensional brush translation, exploration and interaction mode enables precise  $n$ -dimensional patterns to be explored and analyzed in the PCP in an easy and fast way. Minor adjustments can be made to the placement of the brushes updating individual points, or as complete  $n$ -dimensional polylines.

Users can apply the sequence of brush sketches in any order they wish. The advantage of this is that prioritized axes can be filtered first. Subsequent axes can be chosen based on the feedback of the previous filter. The user can also skip over any axis they would not like to include in their filter. Applying higher-order brushes to multiple axes creates a cluster-like filtering of the relational patterns being explored. The features we implement enhance the data selection, filtering, and exploration process such that subsets of the data can be highlighted for further analysis and pattern recognition.

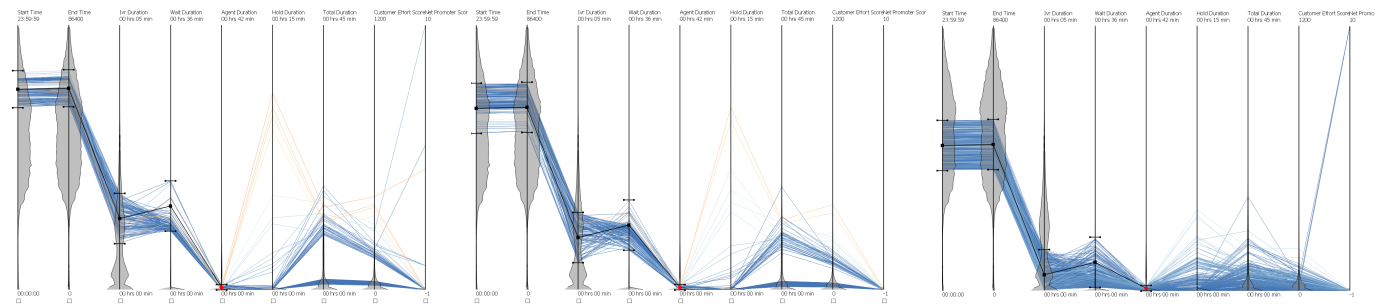


Fig. 6: Higher-order vertical translation. The left image shows a sketch brush placement exploring very low agent  $t$  calls. We pin this brush component in place and then translate the rest of the brush components vertically to explore the rest of the dataset whilst maintaining a low agent  $t$  (middle and right image).

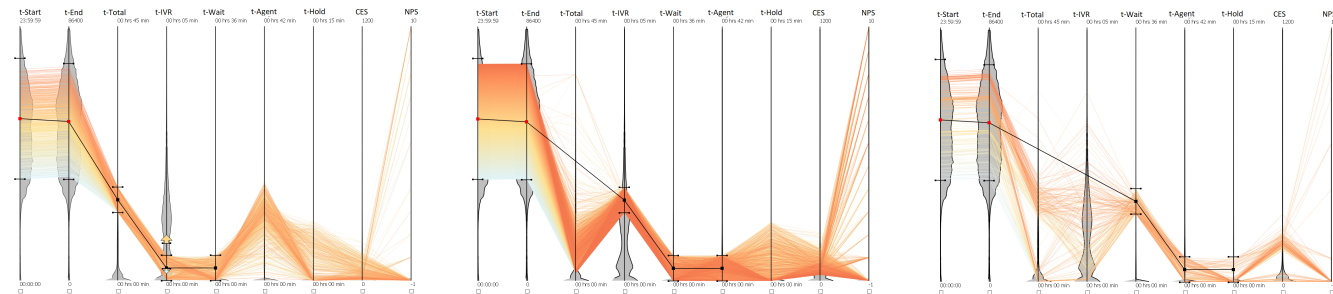


Fig. 7: Horizontal higher-order brush translation feature. In this example, the three images show a sketch brush translated from left to right, exploring a pattern across multiple dimensions.

The user can also pin individual brushes in place so that the rigid brush feature can be used whilst maintaining one or more constant brush points.

**Example 2: Abandoned Call Exploration:** Figure 6 on page 6 demonstrates the higher-order brush being translated vertically. The three images show higher-order brushes placed on the  $t$ -start,  $t$ -end,  $t$ -IVR, and  $t$ -agent axes with the intention of exploring abandoned calls - very low  $t$ -agent. We translate the complete brush upwards, exploring the higher ranges within the axis, however, as we intended to explore abandoned calls, the rigid brush translation moves the agent duration brush placement out of the intended range. Because of this, we use the brush pinning feature which enables the user to pin individual brush components in place whilst translating the remaining sketches. Because of this, we can explore the full range of the data whilst maintaining the anchor on abandoned calls. We observe that as the wait time decreases, callers are less likely to be placed on hold as well as an increasing Net Promoter Score.

**Example 3: Identifying Outlier Calls:** Additionally we demonstrate horizontal translation in Figure 7 on page 6 to explore outlying calls. We place two brushes on the  $t$ -start and  $t$ -end axes covering the operationally busy hours. Then we place a high brush point on the  $t$ -IVR axis and low brush points on the  $t$ -wait and  $t$ -agent axes. Values are distributed generally around zero, any large values are considered outliers. The first image shows outlying calls from high  $t$ -IVR. We then place a pin in the  $t$ -start and  $t$ -end axis brushes and translate the others one axis to the right, (middle image) showing outlying calls from high  $t$ -wait. Finally, we translate the rigid brush right again to highlight outlying calls on the  $t$ -agent axes. Horizontal translation might not always be considered useful if the axes have different units, however, this can be addressed by

simply normalizing the range of values for each dimension.

### 4.3 Logical OR of n-Dimensional Brushes

Multiple n-dimensional brushes can be applied and combined on the PCP. This feature enables the user to define two or more patterns that can be used in two modes of operation. Previously, the OR operator is applied to independent brushes on separate axes. This application of the OR operator is applied to multiple higher-order brushes on each individual axis. The composite, higher-order

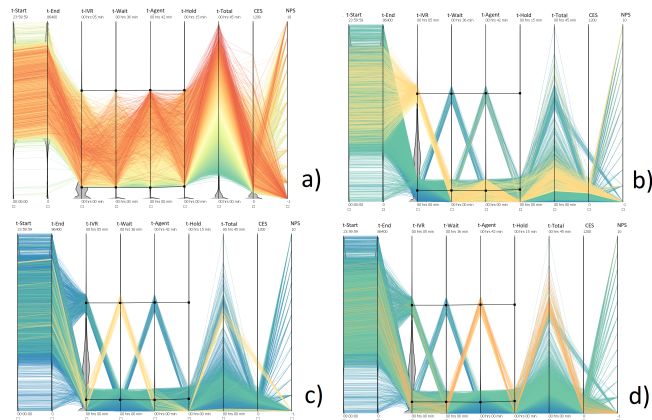


Fig. 8: Logical OR brushing in action. Image a) shows multiple brush points on an individual axes used as edge markers for the brush. Images b), c), and d) show the logical OR feature in use where the priority rendering feature has selected different axes. See section 8.6 for a full description of the feature. Polylines can pass through either of the brushes to remain in context.

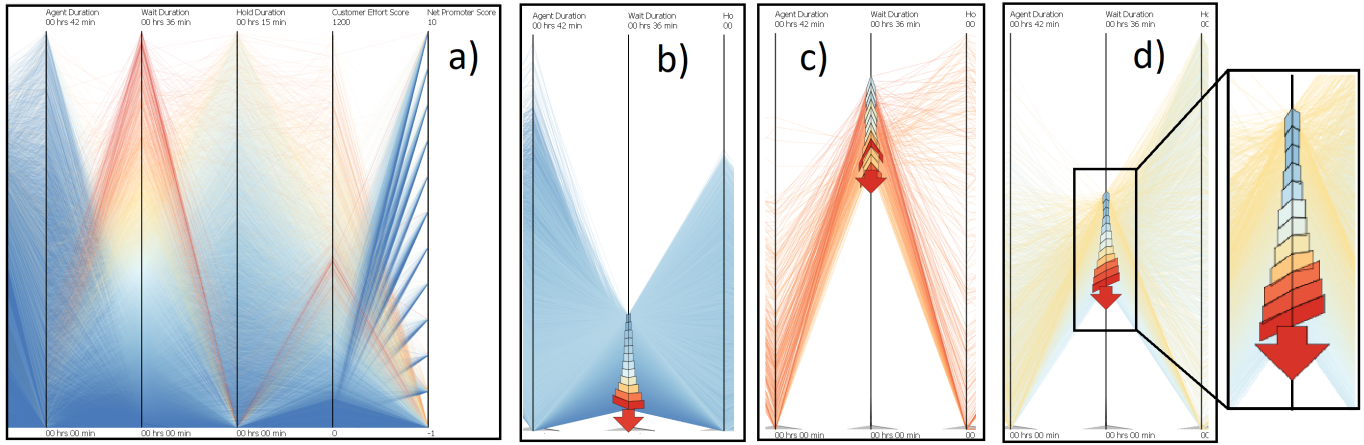


Fig. 9: Dynamic brush interval range glyph – exploring the inverse relationship between  $t$ -hold and  $t$ -wait. This brush interval range glyph enable easy identification of crossover intervals. Image a) shows the unbrushed PCP. Images b), c), and d) show the dynamic brush glyph applied to different points on the axis. Image c) shows the brush glyph at the center point where the relationships between the  $t$ -wait axis and  $t$ -hold axis crossover.

brush points can be combined to form an OR operator filter which displays data records that fall into either brush interval range along the higher-order brush polyline. The two brush positions can alternatively define the upper and lower bounds of the composite, higher-order brush. Thus utilizing each higher-order brush to define the edge of the brush ranges. See Figure 8 on page 6. This enables the user to identify complex patterns in the data that previously would not have been observed by a single  $n$ -dimensional brush intersecting  $n$ -axes.

**Example 4: Clustering Customer Journeys:** Figure 8 (on page 6) demonstrates how multiple brush combinations on a single axes can be used for advanced analysis. Image a) uses the two brush points as boundaries whereby the user can move the upper or lower edge of the axis brush range with precision. The

two higher-order brushes have been positioned on the  $t$ -IVR,  $t$ -wait,  $t$ -agent, and  $t$ -hold axes. The lower brush sits along the bottom of each axis, covering the lower ranges of data. The upper brush runs parallel to the lower brush in the higher ranges of the data. Images b), c), and d) show the logical OR feature enabled so that polylines can pass through either brush point on an axis. Using the priority rendering feature, we focus on the upper brush points revealing the path by which those calls take on the PCP. Each group of polylines follow distinct paths and do not intersect, i.e. no polyline travels from the high  $t$ -IVR to high  $t$ -wait, or from high  $t$ -wait to high  $t$ -agent. These features reveal separate clusters of callers with shared experiences. The priority rendering also enables the user to see how these clusters are distributed over time of day by looking at the first to axes. e.g. b) shows high  $t$ -wait at specific times of day, whereas a) and c) are more evenly distributed throughout.

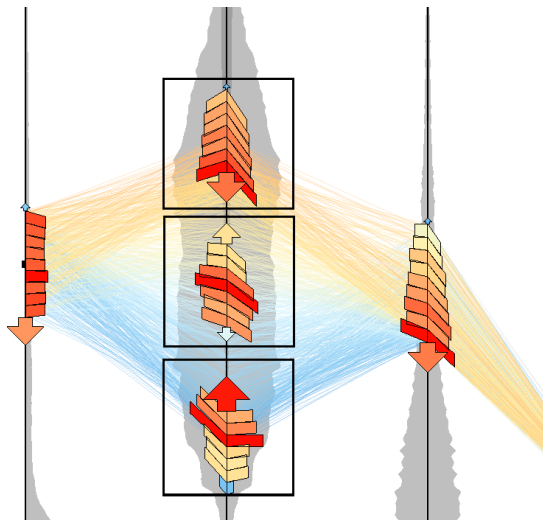


Fig. 10: Brush Interval Range Glyph: This shows the angular histogram glyph displaying a histogram of polyline-axis intersections distributed along the brush range. The three stacked images show the same brush translated along an axis at different times during the interaction. The angle and color of the bars update automatically to represent the average angle of the polylines intersecting the axis within the histogram bar range.

## 5 SMART, DATA-GUIDED BRUSHING

Smart brushing guides the user during the interaction by reflecting the properties of the data at run time and encoding the meta-data in the brush itself. As an extension of the sketch-based interaction and application of  $n$ -dimensional PCP brushes, we implement a series of data-guided features that inform the user in both their creation and sketching of new brushes and in their analysis of the data.

The inspiration behind smart, data-guided brushing is natural. For a new user or a user who is unfamiliar with the data they are exploring, they may be interested in some guidance or help during the higher-order (or any) brushing interaction. A user may need assistance when deciding on an  $n$ -dimensional pattern to search for. They may not know where to start in their multi-dimensional relationship search process. This is precisely what smart, data-guided brushing offers. It does so by deriving meta-data and conveying the meta-data to the user during the interaction. Some examples of meta-data include the number or density of polylines intersecting an axis, the proportion of polylines above or below a brush boundary interval, the average angle of polylines intersecting an axis at a given interval, the most dense pattern of polylines in the PCP space, standard deviations from the mean value along a parallel coordinates axis, and distribution patterns of polylines.



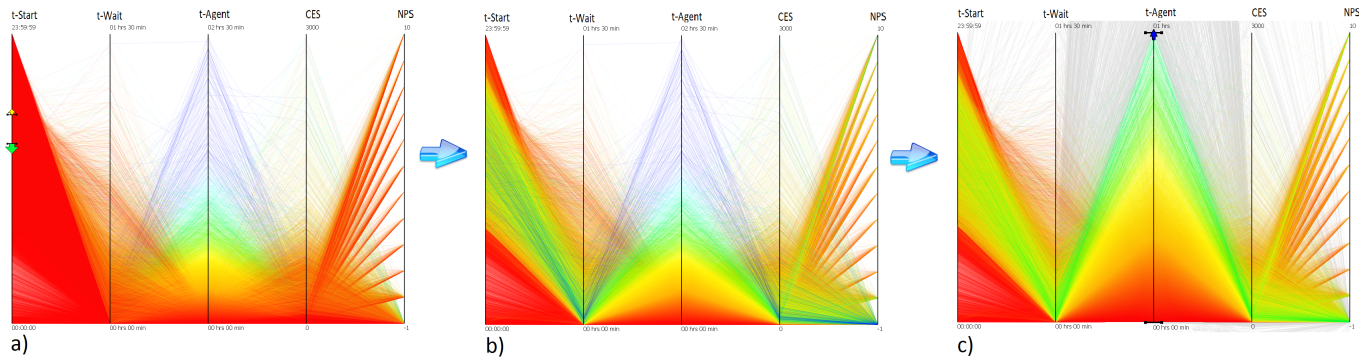


Fig. 11: Smart Axis Scaling and Priority Rendering: a) shows the parallel plot drawn by default, color mapped to *t-agent*. Due to the large number of polylines it is difficult to see the trends on the *t-agent* axis (3<sup>rd</sup> axis). b) If we apply priority rendering to the upper range of this axis we can now see those polyline occluded in the PCP. c) Here we scale the agent duration axis from an upper range of 2 hours and 30 minutes to just an hour. Glyphs on the scaled axis show the user that the axis has been scaled. In addition to this, the focus + context lines show the out of focus, out of axis range polylines rendering upwards out of the screen border. We can see these out-of-range polylines for the total call duration, and Customer Effort Score axes also. This informs the user as to what proportion of data is within focus.

Previous brushing techniques offer little guidance to the user. One way of formulating this lack of guidance is that the brushes lack intelligence. Previously, all of the knowledge or intelligence required for brushing comes from the user and the user’s prior knowledge of the data.

### 5.1 Dynamic Brush Interval Range Glyph

**Interactive Angular Histogram::** In order to provide the user with real-time feedback and guidance whilst applying the interval range sketch-based brush, we use interval range glyphs to represent characteristics of the underlying and surrounding data. The glyph is made up of angular histograms, adapted from the work of Geng et al. [4], whereby angular histograms replace the PCP polylines completely. Our software overlays the angular histogram as a means of providing feedback to the user about the underlying data. The interval range glyph displays a frequency-based angular histogram that summarizes the dense polylines passing through the brush range. See Figure 10 on page 7. These glyphs provide information about the distribution of polylines in an interval brush

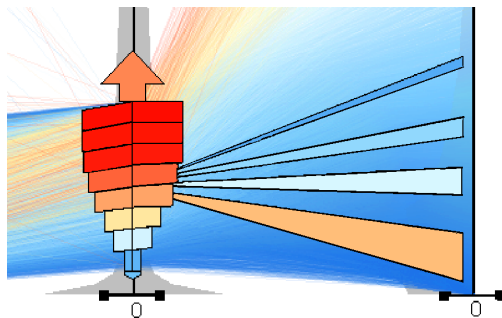


Fig. 12: This figure demonstrates the the dynamic polyline edge that changes thickness when the user is hovering the mouse pointer over an axis. This image shows four different edges that increase in size as the data density increases. When the PCP is occluded, the density and distribution of data can be difficult to accurately observe. This feature addresses that challenge by providing real-time visual cues to the user.

range. The angle of each histogram bar is defined by the average angle of polylines intersecting the axis within the bar range. This is particularly valuable when the polylines are very dense or overplotted and can direct the user to apply the next brush in the same direction as or against the data trend.

When the user interactively adjusts the position of a brush, the angular histogram smoothly animates its update in position and data. See Figure 10. With the addition of these brush interval range glyphs, the user is given real-time feedback about whether the data that falls within selected brush ranges is part of a trend or is considered an outlier. Combining this with the higher-order sketch-based brush we can discover clustered groups of data records of any number or density. The number of bins in the glyph histogram depends on the range of the brush interval. The amount of meta-data detail increases and is reflected by the glyph when brushing with longer ranges.

**Interactive Data-Guided Arrow Glyphs::** To complement these features, we provide the user additional feedback about the underlying data being explored. When the mouse is held over an axis, the brush interval range is shown on screen as well as upward and downward arrow glyphs reflecting how many polylines intersect that given axis above and below the boundaries of the brush interval range. See Figures 10 on page 10, Figure 9 on page 7, Figure 12 on page 8, and Figure 15 on page 11. This provides guidance to the user in their brush placement and is especially helpful for big data when overplotting is a challenge. The user is then informed about how many polylines intersect the given axis above and below the current brush interval boundaries.

**Smart Brush Edges::** When the user is in mid-sketch creating a higher-order brush, a dynamic edge is drawn from the previous axis interaction brush point to the current mouse point. Whilst the user is hovering over an axis, the thickness of this edge changes according to the proportion of polylines passing through that point on the axis, as shown in Figure 12 on page 8. Both the color and thickness of the edge dynamically reflect how dense the underlying data in the area on the axis is with polyline intersections. This real-time feedback guides the user towards the main relational trends in the data or to outliers - whichever the user may be interested in. This interactive, data-guided feedback mechanism creates an easy to use environment for users who

wish for some guidance when defining a higher-order brush or in order to analyze the data at a deeper level. The dynamic nature of this feature necessitates viewing the supplementary video for a demonstration.

**Example 5: Exploring relationships between axes::**

Figure 9 on page 7 illustrates the Brush Glyph’s ability to reveal inverse relationships between data dimensions as well as the crossover point in these relationships. The left image shows the overplotted PCP, where color is mapped to *t-wait*. If we place a brush glyph at the bottom of the *t-wait* axis, we can see that the right side angular histogram bars are all pointing upwards towards a higher *t-hold*. The next image shows the same brush glyph translated upwards towards the top edge of the axis. Here the angular histogram bars are all pointing downwards. If we translate the brush to the middle of the axis we can see the histogram bars tend inwards towards the middle of the brush. The center bar is at a right angle to the axis, revealing the crossover point in the inverse relationship between *t-wait* and *t-hold*.

**5.2 Smart Axis Scaling**

**Dimensional Zooming::** We observe that many datasets, especially in our case study, exhibit polylines clustered in one region on an axis. Inspired by this common occurrence, we implement a data-guided dimensional zooming feature that offers guidance to the user when selecting the upper and lower range of any axis. Through a combination of both higher-order brushes and axis scaling we can change the focus of a set of polylines from millions to just a few data record quickly and easily. See Figure 11 on page 8.

The user can automatically change an axis scale by fixing the range to be a multiplier of the standard deviation. This enables the user to view the majority of the data and automatically drill down into an area of interest with more detail. Manual axis scaling can also be activated through a click-and-drag interaction to the desired new range on an axis whilst the shift key is held.

To convey a smooth transition between the old and new scales, we implement an animated transition whereby two arrow glyphs slide along the axis towards the new interval range boundaries where the scale is applied and then smoothly updates to the upper and lower bounds of the axis. See Figure 11 on page 8. These glyphs remain in position at the edges of the axis until the user resets the scaling. This helps orient the user when navigating the data by keeping them informed as to which axes have been scaled.

**Context Polyline Rendering::** In order to provide feedback to the user about the context data that is no longer inside the magnified axis range, we implement a feature that renders the context polylines that fall out of the magnified axis range. We do this by extrapolating the current axis scale, and rendering the context polylines off-screen at their position on the extended axis. The user is still informed about how much data falls outside of the magnified axis range. See Figure 11 on page 8. This is also demonstrated in the accompanying video.

**Polyline Data Distribution::** In big, dense, relational datasets, overplotting and occlusion often make observation of the data distribution challenging. It may be difficult to gauge where on an axis the data is most or least dense. This limits the user in both the placement of brushes, and in their potential exploration and analysis of the data.

To overcome this, we have implemented data distribution plots surrounding each axis. These plots span outwards from the center

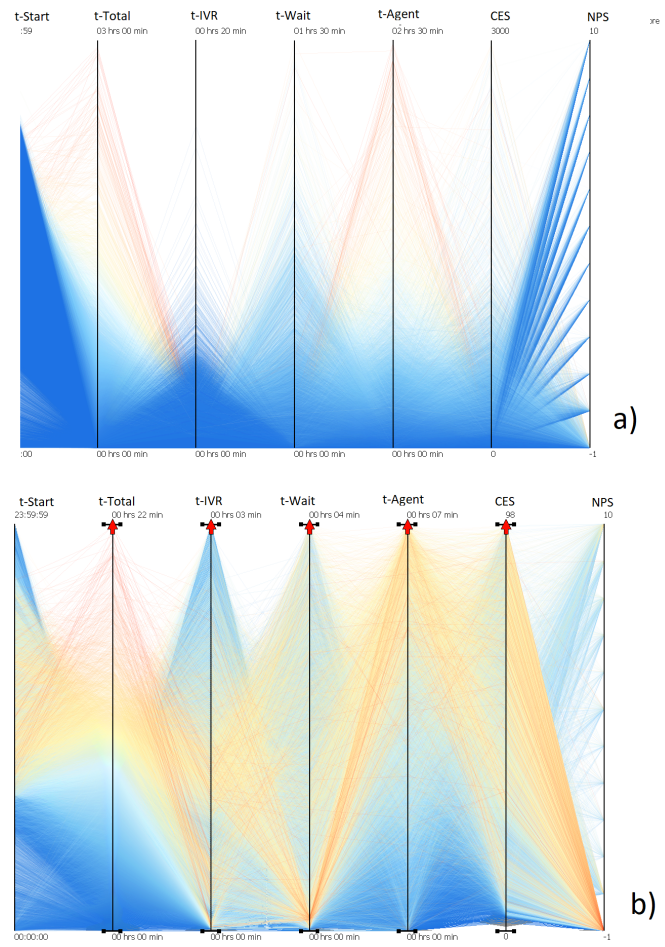


Fig. 13: This figure shows a before and after screen capture of the automatic axis scaling. Image a) shows the axes unscaled and where no priority rendering is set. Polylines are colored by total duration of the call. Image b) displays the automatic axis scale which is applied from the *t-total* axis to the Customer Effort Score axis. The rendering priority has been given to the call duration. The upper ranges of these axes are calculated as 3 standard deviations from the mean of each axis. Whilst the PCP axes now have a much smaller range (maximum 20 minute call) almost 80% of the 230,000 polylines are still in focus. The automated calculations of the auto-complete axis scaling enable precise analysis of large and dense datasets quickly and easily.

of an axis creating a symmetrical pattern showing the distribution of the data. The plot updates automatically as the axis is scaled. We calculate the distribution by splitting the axis into 100 intervals, summing the total number of polyline intersections that fall within each interval, and then normalizing each of those totals by the most populated segment on each axis. Polylines that are out of focus are still included in this calculation. See Figure 1 on page 1, Figure 8 on page 6 & Figure 10 on page 7.

**5.3 Smart Automatic Analysis**

The previous features aid users in the placement of brushes and also in n-dimensional pattern search and analysis. Since higher-order brushing may be difficult to a new user, we also have added fully automatic higher-order brushing to our implementation. The higher-order auto-complete brushes offer guidance to the user during initial brush placement. This guidance, or added intelligence,

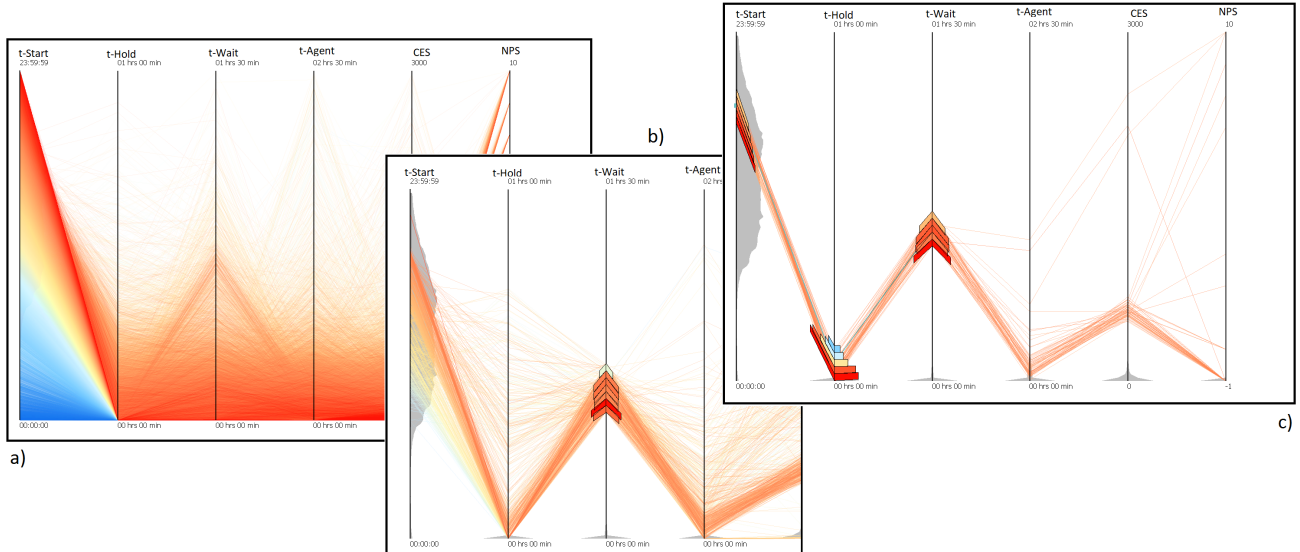


Fig. 14: This figure shows the sketch-based brush data. a) shows an unfiltered PCP where each axis represents a duration metric followed by the two customer scoring metrics. Immediately we see a dense set of  $t$ -wait calls ( $3^{\text{rd}}$  axis) clustered mid-way up the axis. b) shows a brush applied to that cluster of polylines. c) shows the main trend within the cluster using the higher-order brush. A sketch pattern is defined to narrow the scope of polylines in focus. The remaining calls in focus show a high  $t$ -wait but a low  $t$ -agent and all seem to occur within half an hour - from 19:00-19:30.

is derived from properties of the data. To maximize the analytical output of the PCP software we implement several automatic features that can provide feedback and apply features with more precision.

### 5.3.1 Statistical Analysis

We offer three statistical metrics to the user to either display directly on the PCP or facilitate smart, automatic application of higher-order brushes - the clustered centroid, the mean, and the standard deviation. Each provides valuable insight into the n-dimensional trends in the dataset and can either enable an informed manual application of high-order brushing or can automatically apply the smart axis scaling and magnification described below.

**Clustered Centroid:** We calculate the clustered centroid incorporating a user specified interval range (default 10%). We then loop over each interval range of the distribution data summing all the polyline intersections within each interval range. We continue this loop until each position on the distribution has been processed and then use the central point of the range that contains largest number of intersections.

**Mean Polyline:** To calculate the mean polyline, we sum the position of each polyline intersection with an axis and then divide by the total number of polylines. The resulting mean positions are saved as points. They can be drawn as an overlay on the PCP so that the user can see the average data line. A user option enables and disables the rendering of this line. See Figure 15 on page 11.

**Standard Deviation:** Once the mean is calculated, we can add the standard deviation from the mean on each axis. We calculate this by taking the set of polylines  $N$  containing  $x_n$  where  $\mu$  is the mean and applying the following [49]:

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu)^2}$$

This is calculated for each axis. The upper and lower bounds of the SD can be calculated:  $\mu \pm \sigma$ . The user can toggle the drawing of upper and lower standard deviation lines at 1, 2, and 3 standard

deviations away from the mean. See Figure 15 on page 11. The standard deviations are then shown as tick marks along the axis labeled with the  $\sigma$  value. A user-option also displays their actual data values on screen.

### 5.3.2 Higher-Order, Auto-Complete Brushing

We offer an n-dimensional brush, auto-complete feature. The statistical methods described previously can be used to automatically apply higher-order brushes. Depending on the method, user-provided interval brush ranges may be adjusted from their default value to complete the process. If the user's chosen method involves a single point (mean or clustered centroid) then an interval range value is provided by default as input so that the brush can be applied. If standard deviation is selected then the user is presented with options of 1, 2, or 3 standard deviations to apply the higher-order brush range. Alternatively the user can type in a custom decimal value to control the size of the resulting brush ranges.

Once the brush interval range option is selected, the user can specify (all axes by default) the axes which they wish to apply the auto-complete smart brushing to and then run the process. The higher-order brush is then automatically applied to the specified axes and the polylines are filtered.

**Automatic, Smart Axis Scaling and Magnification:** In addition to the auto-complete brushing we implement an automatic n-dimensional axis scaling feature whereby we use the same meta-data guidance as with the brushes and then apply an axis scaling and magnification to the respective brush interval ranges across the PCP axes. This is a logical follow up to the automatic brush application in that the same number of polylines are rendered out-of-focus, however the axis scaling provides a closer, more detailed view at the data within focus and alleviates the challenges posed by occlusion and overplotting efficiently. See Figure 13 on page 9.

**Example 6: Automated Data Focus:** To demonstrate the utility of the automatic axis scaling feature, Figure 13 on page 9 illustrates how this feature can make use of the screen space whilst maximizing the data being visualized. Here we apply an automatic axis scale to 3 standard deviations on the axes call duration,  $t$ -IVR,

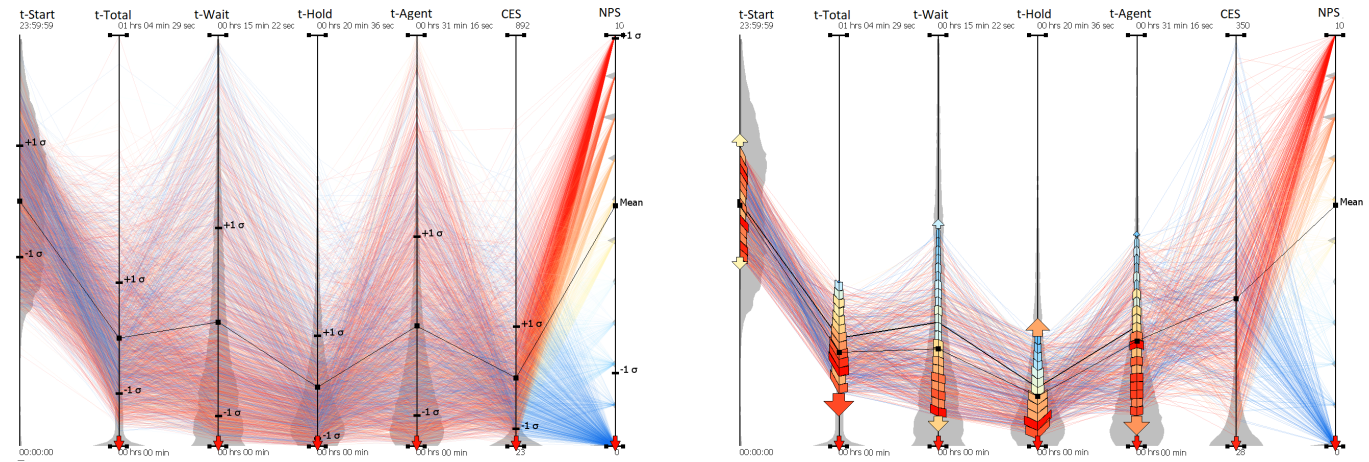


Fig. 15: This figure shows axis scaling and auto-complete brush in use. The left image shows a PCP where the axes have been scaled to view the lower ranges of data. The mean and standard deviations are plotted on the axes. A slight inverse trend can be seen between the NPS and the CES, but occlusion prevents definitive results. Applying an auto-complete brush to 1 standard deviation from the mean reduces this overplotting, and confirms the hypothesis that low CES is an indicator of high NPS.

**t-wait**, **t-agent**, and customer effort score. After smart dimensional zooming, 80% of the original data is still in focus, and yet the duration ranges on each axis have been reduced to a small fraction of their initial values. This concentrated view provides the user with a specific insight into the typical call in the call center. Without this feature, the user would have to apply multiple brushes with pinpoint precision, taking time and effort without any ability to replicate their brush patterns if they wish to attempt this more than once. With the priority rendering set to high **t-total** values, we can see the paths that lengthy calls typically follow.

## 6 COMPLEMENTARY USER OPTIONS

To maximize utility of the PCP software, we ensure each feature has a user option that can enable/disable use of each. In some cases, this is simply to disable the rendering of a glyph but also can be in the form of alpha sliders that adjust the opacity of polylines rendered. When polylines are difficult to see due to being sparsely distributed along an axis, the user can increase the opacity value making them more visible. When overplotting is a challenge, the user can lower the opacity such that the most prominent trends can be seen more clearly.

We provide the user with a choice as to which features they enable and to combine the features in new ways to explore previously unobserved trends or n-dimensional patterns in the data.

### 6.1 Priority Polyline Rendering

Parallel coordinate plots of large datasets like those used in our case study are often subject to overplotting. Density plots can sometimes reveal trends in the polylines, but often the data is simply too dense for meaningful insight. We provide a priority rendering option that enables the user to select any point on an axis and the PCP will prioritize the rendering. This means that polylines intersecting the axis closest to that point will be rendered last and therefore on top. Previous implementations of this do not re-order the draw list but instead apply an opacity value to the lines such that the context becomes non-binary [24], [26], [27]. This implementation uses the color of the polylines to overcome occlusion and so can still be used alongside traditional forms of focus + context brushing and rendering. See Figure 11 on page

8. This feature is implemented as both a continuous mouseover feature whereby the focus is continuously updated according to the mouse position, or through a single click interaction on an axis.

### 6.2 Mouseover Color Selection

This feature dynamically applies the color map to the axis currently being studied. The user moves the mouse over an axis and the polylines will be color mapped according to that axis. The renderer also draws the polylines in order from that axis, enabling a more detailed view into how the dataset interacts with the given axis.

### 6.3 Color Maps

Polylines can be color-mapped by any dimension available in the dataset. The user can select from a list of any data dimensions, not just from the axes currently being rendered. A range of color maps are also available to choose from. We provide a number of diverging color maps from ColorBrewer [50] that are color-blind friendly. We also include an improved rainbow color map from Alex Telea [51]. See Figure 11 on page 8.

## 7 CASE STUDY FROM INDUSTRY - OBSERVATIONS

In addition to the previous examples, we demonstrate the utility of our visual designs using real-world data by exploring previously unseen trends in the call center dataset.

**High t-wait/t-agent ratio:** After loading in the data and selecting a set of dimensions to explore, the first thing we notice is a dense set of polylines around the middle region of the **t-wait** axis (Figure 14a) on page 10. If we apply a brush to that area, both the resulting filter and histogram glyph show a majority of polylines projecting downwards to the lower end of the **t-agent** metric. These high wait time callers do not speak to an agent for very long. If we examine this cluster of polylines on the **t-start** axis, we can see that a large number of calls congregate around a small time frame - at 6:30pm (Figure 14b) on page 10.

If we apply a new brush around that time frame, the plot is still occluded due to the high volume of calls. Using the sketch-based higher-order brushing we can apply the next brush to a low **t-hold** (Figure 14c). We observe that the original dense cluster of calls around the mid point of the **t-wait** axis still remains so we continue

the sketch brushing by applying the next brush to the cluster of high **b**-wait calls. Remaining in focus is a subset of the polylines that represent callers who have had a negative experience with the call center. High **t**-wait and low **t**-agent times are considered a poor quality of service. Now that this cluster of callers has been identified, the reasoning behind this behavior can be explored and recommendations can be given to the call center. Due to the small number of NPS responses, the customer satisfaction may be unclear in these cases.

**Customer Effort Score:** The CES is derived by QPC using a combination of factors that influence the effort exerted by the caller. A number of measures such as **t**-wait, transfer count, and **t**-hold are composited together to predict the customer feedback scores given in the NPS. The advantage of the CES is that it can be derived for each call. To validate the CES against the NPS we need to examine the most important metrics involved in the CES calculation. See Figure 15 on page 11. We load in a number of duration metrics as well as transfer count metrics and plot the PCP. We immediately filter out all non-response NPS data entries and scale the axes so that we can clearly see the customer satisfaction. The majority of polylines intersect the very bottom of each axis representing a near zero duration for most axes. If we apply auto-complete axis scale to the very dense lower segments of each axis we can zoom in and see the data more clearly. Immediately we notice that the trend between NPS and CES is accurate. The majority of zero scoring NPS show a high CES. Also low CES trend with high NPS. This contributes towards the validation of the CES measure. It is also important to note that it appears low CES is better at predicting higher NPS due to the more clearly defined trend between low CES and high NPS.

If we try the automated features in the software, we can remove a lot of noise in the plot by automatically applying higher-order brushes to the **t**-total duration axes at 1 standard deviation from the mean. The result further validates the CES calculation but also reveals some additional trends in the calls. Again we see a cluster of low **t**-total duration calls that originate at a certain time. **t**-wait and **t**-hold appear to be roughly correlated, especially in the lower values. See Figure 15 on page 11.

## 8 INDUSTRY DOMAIN EXPERT FEEDBACK

We undertake a multi-stage domain expert feedback process with our industry partner QPC Ltd. in order to demonstrate the real-world efficacy of the smart brushing features. For the duration of the project, we worked closely with the business that provide us with a large call center activity dataset.

At the preliminary evaluation session we presented the software to three of the data experts to demonstrate its explorative and analytical capabilities. During this session we received a number of suggestions improving the current features and some ideas for new features. As a result of this session we implemented a number of user options that give the user more control over the exploration and analysis. A range of features were also conceived during this session. See Figure 16 on page 13.

### 8.1 Higher-Order Brushing

When demonstrating the higher-order brushing feature, the domain experts are intrigued at the concept of pattern discovery through a sketch-based interface, *“That’s really interesting to know that that’s the average customer journey. It would be good to be able to plot a series of those lines and be able to move them up and down*

*or how the plot changes when you move the lines around slightly”*. We then show them the translation feature that facilitates just this. We hand control over to the experts and they place a brush sketch pattern that explores the odd cluster of high wait time calls. They continue placing brushes at strategic points on the **t**-start and agent duration axes and are left with a subset of calls that are clearly related. This sparks an in depth discussion focused on the possible causes for this phenomena. *“I don’t know the answer, but it’s doing its job already. That’s interesting.”* states an expert as they look at the visualization. We output the data in focus to a .csv file for the analysts to examine at a later date. We continue the exploration, looking at the customer effort score metric that the experts derived. Using the higher-order brushing we looked at how it compares against the net promoter score. Expert 1 states *“So I think what this was hinting at, is that it’s visually reinforcing some of the things we think know about the data. It’s hinting at the fact that there are definitely relationships in the data, things you would expect, but now we can prove them. Now the next thing is to help quantify these relationships.”*

### 8.2 Dynamic Brush Glyph

We begin to demonstrate the dynamic brush glyph. Although the experts had seen images of the glyphs previously, they had not seen them used on their own data with a full explanation of the features. *“It definitely feels like a good investigation type approach. It helps us select things of interest.”* Expert 2 claims. We apply a glyph brush to the **t**-wait axis and translate the brush up and down. The result highlights a crossover point whereby higher **t**-wait calls tend to have lower durations on average. Expert 3 states *“[They] give you that confidence level, that ability to see if things are out of the norm. Are there errors that shouldn’t be there? But also you can see the distribution quickly as well which is helpful.”*

The demonstration continues as the experts explore the software, applying the brush glyphs alongside the higher-order brushing feature to study the data. Expert 3 says, *“And you’ve still got an idea of what you’ve filtered out compared to what you’re viewing. I suppose you’ve got the context lines as well to reinforce that but it can be noisy. The glyphs create a good signal-to-noise ratio.”*. We found that once the glyph had been explained and the experts understood exactly what the angular histogram represented, the feedback was positive. However the initial explanation was more complicated than we anticipated. It appears that interpretation of the glyph requires a more thorough explanation of its features than the higher-order brushing. Expert 3 finished the glyph demonstration by saying *“The brushing glyphs are great. They give you a good indication of how data is that range is spread. I like how you can see the trends in the data outside of the brush range. Indicating if you’re brushing against the flow.”*

### 8.3 Smart Automatic Scaling

Throughout the feedback session, the experts make it clear that the interesting data is clustered very close to the bottom edge of each axis. We have explored data in the higher ranges, but in reality they are seen as outlying data points. At this point we demonstrate the automatic scaling feature to take a closer look at the data in the lower ranges. We apply the automatic axis scale to all the duration axes and set the value to two standard deviations from the mean and the scaling animation slides up the screen showing which axes have been scaled. Expert 1 states - *“This is definitely useful. It’s more of a necessity than an additional feature when used on our dataset. Without it we wouldn’t really be able to look closely at*



Fig. 16: An image from the domain expert feedback session.

*those standard calls at the bottom.*” We ask if the feature would be used, *“We would definitely need to use this feature when analysing the data. It would take far too long to manually change each axis scale according to our needs.”*

Expert 2 claims - *“In a more general sense, this is useful for most datasets as it’s a way of zooming in on the data which we know is replicable. If we use the same data, a 2 standard deviation scale is going to look the same. It removes the time consuming aspect of selecting very small axis scales.”*

#### 8.4 Smart Automatic Brush Application

We go on to show the automated brush application feature - demonstrating how and where this is useful. We apply an automatic axis scale to the duration metrics, and then apply automatic brushes to the same axes. Expert 3 says *“It’s good to see these in action. Whilst the outcome is very similar to the automatic axis scaling, it’s nice to have the context of the current axis scaling to remain consistent so you can better see the patterns and trends in the data.”*

*“This feature is probably better suited to someone who doesn’t really know the data. I imagine we would use the automated brush applications less than someone who had limited exposure to the data.”* Expert 3 states. We show them a few different configurations of the automated brush. *“I like that you have the ability to chose which type of automation you can apply”* states Expert 2, *“Either the automated scaling or the automated brush. It’s easier to see where the data has been filtered with the brush, but the scale increases the resolution of the data.”*

#### 8.5 Smart Guided Features & Off Screen Rendering

We show the experts the guided features that help the user navigate the data whilst applying brushes. *“The fact that you’ve got a summary of distribution is really good. It’s helpful to see how the data is spread out. I’ve not seen that before”* Expert 3 states. *“It’s stopping that bunching of data to the bottom of the axis it just helps when you’ve got that awkward distribution”* he continues.

Whilst applying brushes and axis scales, the experts comment on the off-screen brushing feature. Whereby polylines that are out of context due to axis scaling are still drawn as context to a point offscreen. *“I like the off-screen rendering. It helps give perspective once the axis is scaled. I can see it being really useful I would often get lost without it.”* says Expert 2.

#### 8.6 Priority Rendering

Lastly, we demonstrate the priority rendering feature that enables the user to pinpoint an axis value and the renderer ensures data

closest to that point is drawn last. The experts find this to be an excellent means to overcome dense datasets - *“I guess for me, the point is, we need to get some idea of scale. If we go from 10 million data points down to half a million, it still looks the same. This feature helps us look through the data”* states Expert 1.

When using the priority rendering to reduce the overplotted data we can reveal the correlation between the calculated CES collected NPS. Expert 1 states *“This is actually really good because the theory we are trying to prove that is that time is a really important factor. The variation between high and low CES predictions looks like it’s due to the resolution of the call. Whereby oftentimes negative customers are going to dislike the service regardless of outcome due to past experiences. However the positive feedback customers are easier to predict. This visually reinforces that theory.”* The priority rendering helped us look through the occluded visualization and focus on a specific subset of the data without losing the context at all.

#### 8.7 General Comments

At the end of the session we wanted to get some general feedback about the software as a complete package and how useful the insights were to QPC Ltd. We ask the experts how efficient the new features are compared to the traditional brushing and scaling methods that they are familiar with. *“So there is a small training cost when using these features for the first time, but ultimately if you’re going to use this software for real world insight it’s definitely beneficial to spend the two minutes learning the ropes”*. When asked if some features would require more training than others, Expert 2 responded with *“So I think the higher-order brushing is pretty self explanatory. Though I definitely think that the brush glyph would need to be explained before use as that is less intuitive - but still very useful.”* Expert 3 then states *“We think the software is really useful for us to explore our data in a new way, and the tools you’ve provided let us do things we would never have been able to do and see before. I would like to be able to see the software used on our other datasets to see what it can find on those.”*

### 9 CONCLUSION & FUTURE WORK

In this paper, we present a number of novel smart brush interaction and visualization techniques to enhance the parallel coordinates plot. Our first contribution represents the next step in brushing techniques, enabling a sketch-based approach to higher-order brushes facilitating multidimensional pattern searches with new and novel interaction methods. Furthermore we present smart, data-guided brushing that provides the user real-time feedback regarding their interaction with the PCP using meta-data to inform them of their potential brush placements. We design features that address the challenge of large, overplotted datasets and we create smart, automated application of the features to assist users unfamiliar with the data or to aid in fast, precise analysis.

We provide a real-world case study example to demonstrate the analytical and explorative capabilities of the novel interaction features and then conduct detailed interviews with call center domain experts to evaluate the effectiveness of the features. The domain expert feedback attests to the utility of the new features and provided the telecommunications experts with previously unobserved patterns in the large call center data as well as confirming hypothesis they hold about trends and relationships within the data. The interaction and analysis techniques offer novel methods for approaching the challenges.

In the future we aim to continue work in this direction. address the challenges of large, multidimensional visualization. We plan on designing visualizations and techniques that accommodate twelve to eighteen months of call center data. Utilizing the ‘big data’ advantages that the expanded volume of data will bring and explore its potential in the field of overplotting reduction and automatic analysis. Additionally, we could perform detailed user studies so that we can quantify the training cost of using these features as well as a comparison between traditional parallel plot features and our improved methods.

## 10 ACKNOWLEDGEMENTS

We would like to thank all who have contributed to this research. We thank QPC Ltd. for their financial support and their supply of the call centre data. We also thank KESSII who have been the primary funding body for this research. KESS is part funded by the Welsh Government’s European Social Fund (ESF) convergence programme for West Wales and the Valleys. In addition to this, would like to thank Dylan Reese, Liam McNabb, and Joss Whittle for providing valuable feedback on this work and we thank Alfred Inselberg for his inspiring discussions and talks on parallel coordinates.

## REFERENCES

- [1] J. Johansson and C. Forsell, “Evaluation of parallel coordinates: Overview, categorization and guidelines for future research,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 579–588, 2016.
- [2] L. McNabb and R. S. Laramée, “Survey of Surveys (SoS) Mapping The Landscape of Survey Papers in Information Visualization,” in *Computer Graphics Forum*, vol. 36, no. 3. Wiley Online Library, 2017, pp. 589–617.
- [3] R. G. Raidou, M. Eiseemann, M. Breeuwer, E. Eiseemann, and A. Vilanova, “Orientation-enhanced parallel coordinate plots,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 589–598, 2016.
- [4] Z. Geng, Z. Peng, R. S. Laramée, J. C. Roberts, and R. Walker, “Angular histograms: Frequency-based visualizations for large, high dimensional data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2572–2580, 2011.
- [5] M. d’Ocagne, *Coordonnées parallèles & axiales: méthode de transformation géométrique et procédé nouveau de calcul graphique déduits de la considération des coordonnées parallèles*. Gauthier-Villars, 1885.
- [6] A. Inselberg, “N-dimensional coordinates,” *Picture Data Description & Management*, p. 136, 1980.
- [7] —, “N-dimensional coordinates,” *Proceedings of IEEE Pattern Analysis and Machine Intelligence, Picture Data Description and Management, Asilomar*, p. 136, 1980.
- [8] —, “The plane with parallel coordinates,” *The visual computer*, vol. 1, no. 2, pp. 69–91, 1985.
- [9] —, “Discovering multi-dimensional structure using parallel coordinates,” *Proceedings of the American Statistical Association, Section on Statistical Graphics, Washington, DC*, pp. 1–16, 1989.
- [10] A. Inselberg and B. Dimsdale, “Parallel coordinates,” in *Human-Machine Interactive Systems*. Springer, 1991, pp. 199–233.
- [11] E. J. Wegman, “Hyperdimensional data analysis using parallel coordinates,” *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 664–675, 1990.
- [12] A. Inselberg, *Parallel coordinates*. Springer, 2009.
- [13] N. Elmqvist, J. Stasko, and P. Tsigas, “DataMeadow: a visual canvas for analysis of large-scale multivariate data,” *Information visualization*, vol. 7, no. 1, pp. 18–33, 2008.
- [14] H. Gruendl, P. Riehm, Y. Pausch, and B. Froehlich, “Time-series plots integrated in parallel-coordinates displays,” in *Computer Graphics Forum*, vol. 35, no. 3. Wiley Online Library, 2016, pp. 321–330.
- [15] M. Tory, S. Potts, and T. Moller, “A parallel coordinates style interface for exploratory volume visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 1, pp. 71–80, 2005.
- [16] A. Inselberg, “A survey of parallel coordinates,” in *Mathematical Visualization*. Springer, 1998, pp. 167–179.
- [17] J. Heinrich and D. Weiskopf, “State of the art of parallel coordinates,” *STAR Proceedings of Eurographics*, vol. 2013, pp. 95–116, 2013.

- [18] A. Dasgupta, M. Chen, and R. Kosara, “Conceptualizing visual uncertainty in parallel coordinates,” in *Computer Graphics Forum*, vol. 31, no. 3pt2. Wiley Online Library, 2012, pp. 1015–1024.
- [19] M. A. Fisherkeller, J. H. Friedman, and J. W. Tukey, “An interactive multidimensional data display and analysis system,” 1974.
- [20] R. A. Becker and W. S. Cleveland, “Brushing scatterplots,” *Technometrics*, vol. 29, no. 2, pp. 127–142, 1987.
- [21] R. A. Becker, W. S. Cleveland, and A. R. Wilks, “Dynamic graphics for data analysis,” *Statistical Science*, pp. 355–383, 1987.
- [22] B. Shneiderman, “Dynamic queries for visual information seeking,” *IEEE software*, vol. 11, no. 6, pp. 70–77, 1994.
- [23] T. Avidan and S. Avidan, “Parallax—a data mining tool based on parallel coordinates,” *Computational Statistics*, vol. 14, no. 1, pp. 79–89, 1999.
- [24] A. R. Martin and M. O. Ward, “High dimensional brushing for interactive exploration of multivariate data,” in *Proceedings of the 6th Conference on Visualization ’95*. IEEE Computer Society, 1995, p. 271.
- [25] H. Chen, “Compound Brushing,” in *Proceedings of the IEEE Symposium on Information Visualization*. IEEE, 2003, pp. 181–188.
- [26] H. Hauser, F. Ledermann, and H. Doleisch, “Angular brushing of extended parallel coordinates,” in *Proceedings of the IEEE Symposium on Information Visualization*. IEEE, 2002, pp. 127–130.
- [27] D. Feng, L. Kwock, Y. Lee, and R. Taylor, “Matching visual saliency to confidence in plots of uncertain data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 980–989, 2010.
- [28] P. C. Wong and R. D. Bergeron, “Multiresolution multidimensional wavelet brushing,” in *Proceedings of the IEEE Conference on Visualization*. IEEE Computer Society Press, 1996, pp. 141–ff.
- [29] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner, “Hierarchical parallel coordinates for exploration of large datasets,” in *Proceedings of the IEEE Symposium on Information Visualization*. IEEE Computer Society Press, 1999, pp. 43–50.
- [30] —, “Navigating hierarchies with structure-based brushes,” in *Proceedings of the IEEE Symposium on Information Visualization*. IEEE, 1999, pp. 58–64.
- [31] D. Baur, B. Lee, and S. Carpendale, “Touchwave: kinetic multi-touch manipulation for hierarchical stacked graphs,” in *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces*. ACM, 2012, pp. 255–264.
- [32] G. Andrienko and N. Andrienko, “Constructing parallel coordinates plot for problem solving,” in *Proceedings of the International Symposium on Smart Graphics*, 2001, pp. 9–14.
- [33] R. Theron, “Visual analytics of paleoceanographic conditions,” in *Proceedings of the IEEE Symposium on Visual Analytics Science & Technology*, 2006, pp. 19–26.
- [34] R. Sadana and J. Stasko, “Expanding selection for information visualization systems on tablet devices,” in *Proceedings of the ACM Conference on Interactive Surfaces and Spaces*. ACM, 2016, pp. 149–158.
- [35] R. Kosara, “Indirect multi-touch interaction for brushing in parallel coordinates,” 2011. [Online]. Available: <http://dx.doi.org/10.1117/12.872645>
- [36] M. Nielsen, N. Elmqvist, and K. Grønbaek, “Scribble query: fluid touch brushing for multivariate data visualization,” in *Proceedings of the 28th Australian Conference on Computer-Human Interaction*. ACM, 2016, pp. 381–390.
- [37] M. O. Ward, “XmdvTool: Integrating multiple methods for visualizing multivariate data,” in *Proceedings of the Conference on Visualization ’94*. IEEE Computer Society Press, 1994, pp. 326–333.
- [38] E. A. Rundensteiner, M. O. Ward, J. Yang, and P. R. Doshi, “XmdvTool: visual interactive data exploration and trend discovery of high-dimensional data sets,” in *Proceedings of the ACM Conference on Management of Data*. ACM, 2002, pp. 631–631.
- [39] Enguerrand de Rochefort, *XDAT, XDAT*, Eschweiler, Germany, 2017. [Online]. Available: <http://www.xdat.org/index.php?ref=manual>
- [40] Y. Priyadarshana, K. Gunathunga, K. N. N. Perera, L. Ranathunga, P. Karunaratne, and T. Thanthriwatta, “Sentiment analysis: Measuring sentiment strength of call centre conversations,” in *Proceedings of the IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. IEEE, 2015, pp. 1–9.
- [41] A. Raja, W. Mohsin, N. Ehsan, E. Mirza, and M. Saud, “Impact of emotional intelligence and work attitude on quality of service in the call centre industry of pakistan,” in *Proceedings of the IEEE International Conference on Management of Innovation and Technology*. IEEE, 2010, pp. 402–407.
- [42] W. M. van der Aalst, B. F. van Dongen, C. W. Günther, R. Mans, A. A. De Medeiros, A. Rozinat, V. Rubin, M. Song, H. Verbeek, and A. Weijters, “Prom 4.0: comprehensive support for real process analysis,” in *International Conference on Application and Theory of Petri Nets*. Springer, 2007, pp. 484–494.

- [43] L. Brown, N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, and L. Zhao, "Statistical analysis of a telephone call center: A queueing-science perspective," *Journal of the American statistical association*, vol. 100, no. 469, pp. 36–50, 2005.
- [44] O. Garnett, A. Mandelbaum, and M. Reiman, "Designing a call center with impatient customers," *Manufacturing & Service Operations Management*, vol. 4, no. 3, pp. 208–227, 2002.
- [45] R. C. Roberts, C. Tong, R. S. Laramée, G. A. Smith, P. Brookes, and T. D’Cruze, "Interactive Analytical Treemaps for Visualisation of Call Centre Data," in *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference*, G. Pintore and F. Stanco, Eds. The Eurographics Association, 2016.
- [46] D. Galanis, S. Karabetos, M. Koutsombogera, H. Papageorgiou, A. Esposito, and M.-T. Riviello, "Classification of emotional speech units in call centre interactions," in *Proceedings of the IEEE Conference on Cognitive Infocommunications*. IEEE, 2013, pp. 403–406.
- [47] A. K. Jaiswal, "Customer satisfaction and service quality measurement in indian call centres," *Managing Service Quality: An International Journal*, vol. 18, no. 4, pp. 405–416, 2008.
- [48] N. Malhotra and A. Mukherjee, "The relative influence of organisational commitment and job satisfaction on service quality of customer-contact employees in banking call centres," *Journal of services Marketing*, vol. 18, no. 3, pp. 162–174, 2004.
- [49] P. Newbold, W. Carlson, and B. Thorne, *Statistics for business and economics*. Pearson, 2012.
- [50] M. Harrower and C. A. Brewer, "Colorbrewer.org: an online tool for selecting colour schemes for maps," *The Cartographic Journal*, vol. 40, no. 1, pp. 27–37, 2003.
- [51] A. C. Telea, *Data Visualization: Principles and Practice*. CRC Press, 2014.



**Gary A. Smith** Gary received his MBA from Leicester University in 2008 and a BSc in Mathematics & Statistical Computing from Liverpool John Moores University in 2006. He is also an Associate of the Operational Research society. Gary is currently the Director of Product & Marketing at QPC Group and is responsible for the strategic direction & innovation of QPCs products.



**Paul Brookes** Paul Brookes is currently a lead developer at QPC Limited. Since joining Callscan Limited in 1989 he has continued to work on development of numerous computer telephony integration projects for contact centres providing real time and historic management information. His current interests include new and unique ways visualising of customer journey data for contact centres.



**Richard C. Roberts** Richard Roberts received his bachelors degree in Business Management from Swansea University in 2013. He was awarded a Masters in Computer Science from Swansea University specialising in data visualisation in 2014. After working in research of business data visualisation he was accepted as a PhD candidate at Swansea University in 2015 to continue his work in that field. His interests follow the route of his research with the visualisation and analysis of business data.



**Robert S. Laramée** Robert received a bachelors degree in physics, cum laude, from the University of Massachusetts, Amherst (Zoo Mass) and a masters degree in computer science from the University of New Hampshire, Durham. He was awarded a PhD from the Vienna University of Technology, Austria at the Institute of Computer Graphics and Algorithms in 2005 (Gruess Gott TUWien). From 2001 to 2006 he was a researcher at the VRVis Research Center ([www.vrvis.at](http://www.vrvis.at)) and a software engineer at AVL

([www.avl.com](http://www.avl.com)) in the department of Advanced Simulation Technologies. Currently he is an Associate Professor at the Swansea University (Prifysgol Cymru Abertawe), Wales in the Department of Computer Science. His research interests are in the areas of scientific visualization, information visualization, and visual analytics.



**Tony D’Cruze** Tony is a contact centre expert, with more than 20 years spent in a variety of contact centre roles in the Financial and Outsourcing sectors. Passionate about efficiency and driving improvements in customer service, Tony is now a Consultant with QPC and supports clients with leveraging QPC solutions to transform their customer contact operations.