

# PADE: Supporting Collaborative Visual Analysis of Patient Administrative Systems Data with a Large Touch Display System

Søren Knudsen\*  
University of Calgary

Kasper Hornbæk†  
University of Copenhagen



Figure 1: **Middle:** Analysts use PADE to explore age distributions of medical treatments related to pregnancy and labor across the five Danish regions. This scenario is explained in Section 5. **Left:** Create views by dragging data fields onto the canvas. **Right:** Explore parts of data in a view by dragging data bars to the canvas. This creates a drilled down view.

## ABSTRACT

We present a visual analytics tool for collaboratively exploring data from patient administrative systems on large touch displays in meeting contexts. Large touch displays are becoming commercially available, but we have limited knowledge about how they might be used in such a context. We investigated this with our system, PatientAdministrativeDataExplorer (PADE). We designed PADE based on inquiries with healthcare data analysts tasked with understanding expenses in a healthcare system that serve about six million residents. Our goals in designing the system were to enable the analysts to collaboratively construct hypotheses, quickly generate and execute strategies, and support ad hoc discussions and Q&A sessions during meetings. We created a set of interaction techniques that let users create new visualizations and combine parts of existing ones, and illustrate these possibilities through an analysis scenario. Finally, we discuss the possibilities and limitations of PADE, its interaction techniques, and future work in this direction.

## 1 INTRODUCTION

While data analysts in teams rarely conduct collaborative analyses, they frequently communicate and discuss their analysis approaches, scripts, data sources, and results [33]. This is particularly relevant when analysing healthcare data, which often require interdisciplinary collaboration between healthcare professionals, computer scientists, statisticians, and related fields. Additionally, tools that support this process are mostly designed for single users. Thus, when analysts meet in face-to-face meetings, their tools fall short. Similarly, when analysts meet with analysis consumers, they typically prepare static reports and charts before such meetings. However, in contrast to static reports, it might be possible to support these contexts through collaborative visual analytics.

We see many options for such support. Distributed, multiple-device analytics systems might enable analysts to use their existing mobile devices to collaborate on data analysis (e.g., [4, 37]). Although this might work when all analysts have access to devices, collaboration issues might arise from decoupled visualization workspaces. Alternatively, large, high-resolution displays offer space for showing many freely arrangeable views at the same time, allowing analysts to use space to make sense of data [1, 29, 52]. Combined with touch interaction, they might also support walk-up use and collaboration by offering a ‘shared information visualization workspace’ [51]. Teams can move through data, gather relevant excerpts, and discuss these [54]. They might further help analysts collaboratively keep an overview of analysis provenance, findings, and decisions [43]. Thus, visualizations on such displays promise to advance collaboration during exploratory analyses of large data sets and so improve data-based decision making.

In this paper, we introduce PatientAdministrativeDataExplorer (PADE). We designed PADE for a team of healthcare data analysts. As part of our collaboration, we conducted contextual interviews and design workshops with the team. Their analysis challenges led us to design a set of interaction techniques for co-located collaborative impromptu data analysis on large displays. The goal of these techniques was to ease collaborative exploration of large multi-dimensional data sets by enabling creation and combination of visualizations, in the context of walk-up use. PADE provides touch interaction techniques for visualizations on large displays. These interaction techniques provide a novel approach to querying multi-dimensional data and support drilling down, filtering, and grouping data. We based PADE’s interaction techniques on bar charts and use direct manipulation to interact with and combine database queries using visualizations of previous database query results through touch interactions. PADE enables analysts to collaboratively construct hypotheses, quickly generate and execute strategies, and support ad hoc discussions and Q&A sessions during meetings.

We contribute: (i) a **characterization** of a team of healthcare **data analysts’ work**; and (ii) a set of **interaction techniques**, that facilitate quick and flexible collaborative data exploration using touch on large displays that were informed by this work domain.

\*e-mail: sknudsen@ucalgary.ca

†e-mail: kash@diku.dk

## 2 RELATED WORK

We first discuss the area of interactive data analysis, then review visualizations in the context of large displays and touch interaction.

Stolte et al. [50] present Polaris (laying the foundation for Tableau), which facilitates exploratory data analysis through drag and drop interaction with database schema fields and with a focus on constructing queries to produce visualizations. Since the original publication, Tableau has gained capabilities for showing multiple visualizations side by side. However, once a visualization is created, it is not possible to derive new visualizations from the visualization itself, but only from menu interactions. Drucker et al. [15] study these alternatives for bar chart interactions. They discuss how a gestural interface guides participants with low experience with data analysis towards solving tasks, but at the same time limits more experienced participants. Zgraggen et al. [56] contribute a system design for coordinating visualization views by linking them together. They note that legends were not necessary due to their use of views on a canvas. Instead, legends “are visualizations themselves and can be derived from an existing visualization.” We rely on these attributes of visualization views in this paper.

We see a growing body of large-display visualization research. This is exemplified by novel visualization designs (e.g., [3]), interaction techniques (e.g., [24, 28]), lab studies (e.g., [9, 44]), and lab evaluations (e.g., [3]). Large displays allow for co-located collaboration, discussion, exploration, and analysis using information visualizations, by providing a shared workspace with sufficient room for both individual and group work. Andrews et al. [2] note that information visualization work has mainly focused on single visualizations or multiple visualizations fixed at particular spatial positions. They argued that large displays can “shift interaction emphasis onto spatial interaction, such as drag and drop and spatial arranging.” Large displays have been used to let users lay out and make sense of documents (e.g., [25, 29]). Andrews et al. [1] design an environment for text document sensemaking tasks on a tiled desktop display. They suggested the increased display space increased users’ ability to carry out data analysis by giving “space to think.” Like documents, visualizations can be laid out on large displays to support sensemaking. However, only a few examples of this approach exist. Tobiasz et al. [51] create a meta-visualization to link visualization views. Their meta-visualization shows a visualization pipeline representation of relationships between views. In contrast, the approach is more common in virtual canvas’ es (e.g., [16, 20, 30, 56]). These systems seem inspired by the data-flow paradigm (e.g., [6, 11]) but focus on providing an overview of visualization provenance [43]. Close to our work, Dunne et al. [16] demonstrate the importance of showing links between views that leave a trail of visual breadcrumbs representing users’ exploration, thus providing a visual analysis provenance. This supports users’ analyses by helping them understand how a view was created, recall the exploration history, and share analyses with collaborators. Knudsen et al. [35] characterize links between views that show analysis history and data processing as trails of thought and suggest these supports backtracking and fluid exploration of alternatives. We follow this line of thinking in our work.

In addition to the research described above on large-display visualizations, much research has been done on touch interaction techniques for visualizations, exemplified by novel touch interaction techniques (e.g., stacked area charts [5], scatter plots [45], and node-link diagrams [36, 48]) and lab evaluations (e.g., [15, 36]). Touch and pen interaction may also be used to facilitate data exploration by providing intuitive and effortless interaction techniques. SketchVis [12] allow users to create data-driven charts by drawing on an interactive whiteboard, supporting easy exploration of data. In a follow-up study, Walny et al. [53] study combinations of pen and touch input with SketchVis—later realized in a concrete system [40]. These contributions focus on interaction techniques for a single visualization view and facilitating close-up work on large displays. In contrast,

we focus on between-view interactions. Sadana & Stasko [46, 47] contribute touch interactions techniques for several visualization techniques. Most importantly for our work, they provide techniques for multiple views [46]. We are inspired by their work to consider more free-form creation and manipulation of views.

Several calls for understanding the space of collaboration, large displays, visualizations, and touch have been made [2, 26, 39]. However, while recent work combine the use of touch and large displays (e.g., [24]), such combinations are still rare in comparison to mouse and keyboard interaction. Additionally, we have limited understanding of how to support collaborative data analysis in these contexts. We think collaborative data exploration can benefit from both large displays and dedicated techniques, not only for using touch to create and configure visualization views, but also for combining existing views to create new ones. We present a system that combines these ideas and describe the design process in the following section.

## 3 DESIGN WORK

We conducted the design work for PADE with a team of healthcare data analysts (hereinafter ‘analysts’). This team performs analysis and documentation for the Danish healthcare system which comprises about 50 public hospitals that serve a population of around 6 million people, and handle about 13 million patient contacts per year. We first describe the domain.

### 3.1 Domain inquiries

We conducted observations and contextual interviews [8] over two weeks at the analysts’ workplace to obtain a thorough understanding of their tasks and relied on qualitative methods for designing and evaluating information visualizations [14, 27]. Later, we returned to the site for shorter day-visits to understand how their work varied in the course of a year. The analysts’ work reminds of tasks and contexts characterized by [33], and their level of expertise falls somewhere between hackers and scripters. The group of analysts comprise about 10 employees, and are part of a group of about 35 employees that work with documentation of healthcare services. They have mixed backgrounds, including economics, political science, mathematics, statistics, medicine, public health science, and computer science. They primarily use SAS, SQL, and MS Excel for data analysis. They use these tools to create visualizations, such as bar charts, line charts, and scatterplots.

**Domain Tasks and Data:** The analysts’ main task was to compute annual rates for hospital treatments (Diagnosis-Related Groups [18]). The government uses these for financial compensations. To compute the rates, the analysts regularly obtained data from all national hospitals. The data is primarily used to map hospital activities data to expenses.

**Rate Foundation Table:** The received data comprised medical activities and financial accounts data. Medical activities data describes what has happened at a hospital (e.g., patient admittance and discharge dates from the wards and blood test metadata from clinical biochemistry labs). Financial accounts data describes the expenses incurred at a hospital (e.g., doctor and nurse salary expenses, implant costs, and overhead costs for each hospital department).

To compute the rates, the analysts mapped medical activities and financial accounts data in what they refer to as the **rate foundation table**. They constructed the table based on a set of scripts—a subset of which relate to individual hospitals. The table contains a row for each patient (about 13 million per year). Each row describes a patient contact (an admission and discharge for inpatients and comparable information for outpatients) and comprises columns of patient information (e.g., age, gender, diagnoses), treatment information (e.g., procedures, duration, ward, hospital), and cost information (e.g., diagnosis-related group, salaries, overhead). For example, codes describe medical procedures in a hierarchy of about 9,000 procedures and location definitions in a hierarchy of about 20,000

wards that describe physical locations with changing names and id over time. As a side note, these data sets might warrant separate visualizations.

The analysts' work is characterized by constant adaptation to changing healthcare policies. To keep up with changing healthcare policies, the analysts **revised the scripts** used to create the rate foundation table. Thus, a large part of their work lay in 'data wrangling' [32,33], including revising data flows and understanding where errors have occurred in the process. Due to policy changes, information codes might be added, changed, or removed. For example, new administrative patient pathway codes, changes of codes describing in- and outpatients, and introductions of new medical procedures, might require new description codes. Thus, while the analysts have scripts from previous years available, they will rarely work due to syntactic, semantic, and structural changes in the data between the previous years and the current. To develop new scripts, they look at scripts from previous years, rely on knowledge in the team, and debugging based on script outputs.

**Context of Work:** To discuss ongoing work, the analysts held **weekly team meetings** with their manager. In these meetings, each analyst provided a brief status update, including data, analysis, and scripting problems. Then, the issues were discussed between the analysts. After the meeting, the analysts returned to their desks. For example, in one meeting, an analyst presented a scripting problem relating to implant costs from a specific hospital. Another analyst queried "did you look into whether [the data] contained all implants," since they had experienced implant types that caused problems in their scripts. This question required the analyst to look at the data again in order to be able to answer the question. In addition to weekly analysis meetings, pairs of analysts often met for smaller half to two-hour meetings to work on a shared task in front of a computer. These meetings took place in the context of an informal work environment divided in three- to four-person offices in which the analysts frequently interrupted each other with quick questions. For example, "do you remember the code for the new cancer treatment?"—not unlike 'blast-mails' described by Kandel et al. [33].

### 3.2 Design goals

The analysts often discussed issues regarding the rate foundation table during meetings, and looked at the data individually after the meetings. With PADE, we aimed to let the analysts discuss problems with the rate foundation table and their scripts while **interacting with the data in meetings**. We imagined the analysts could use PADE collaboratively to quickly **construct hypotheses, generate strategies, and perform analyses**, thus supporting ad hoc discussions and immediate answers to questions about data. To support mid-meeting collaborative analysis we defined these design goals:

**DG 1** Walk-up use in a collaborative scenario. We support this by providing techniques for quickly obtaining high-level information, constructing overviews of data, and by using sensible defaults.

**DG 2** Quick and immediate analysis, rather than intermediary menus to allow the analysts to conduct impromptu analyses during data meetings.

**DG 3** Allow one person to continue the analysis thoughts of another to support flexible collaboration, communication, and arguing with data and visualizations.

We conducted **design workshops** to kickstart the design process. (see Knudsen et al. [35], workshop A and C). Here, the analysts worked creatively with us to come up with novel interactive visualization designs inspired by their domain. For example, participants in the design workshops considered support for exploring why the number of patients admitted for specific treatments dropped from one year to the next and analyzing the cost distributions of specific treatments across hospitals. The analysts frequently discussed the

use of multiple views and how to represent relationships between them in these workshops.

The **analysts asked** for ways to efficiently collaborate and continue each others' analysis actions, as a form of shared dialogue with data. They imagined ways to understand and integrate their collaborators analysis actions. From this pre-design empiricism, we formulated the following design goals:

**DG 4** Multiple views for provenance.

**DG 5** Shared visual representation of analysis history.

**DG 6** Collaborative data exploration.

To better understand the requirements for a fully implemented system, we **constructed lo-fi prototypes** and evaluated them with the analysts. For example, we presented a mockup implementation that showed a range of visualization views that could be positioned freely on a large display. Each view showed their data set filtered in a manner similar to their existing work practices. Presenting the mockups to the analysts enabled us to inquire about how they imagined this would be used in a collaborative analysis context, and thus, better understand the requirements for an implemented system.

By **discussing the mockups**, we learned more details about the analysts needs. For example, they imagined spatially exploring patient contacts for each hospital within a region in detail, when looking at a region overview. Continuing to explore other regions in the overview would allow them to compare to other regions. They also asked for ways to record findings in the middle of collaborative analyses, to explore alternative data analysis approaches, or to split an analysis into two comparable branches, for example, allowing collaborating analysts to compare cost distributions for large and small hospitals in two separate views. Further, they also asked specifically for the ability to be able to both use spatial and temporal juxtapositioning. Finally, the analysts asked for ways to understand data transformation implications and exploration of what-if scenarios. They imagined combining different parts of data sets and understand how they could map information from multiple data fields. From this, we defined the following design goals:

**DG 7** Facilitate creating analysis alternatives and branches.

**DG 8** Analysis of data transformation implications.

**DG 9** Comparison of visualization views side-by-side and by alternating between them (i.e., spatial and temporal juxtaposition [19]).

**DG 10** Facilitate comparing visualization views across values of a table column.

**DG 11** Facilitate drilling down.

Understanding the analysts' needs and objectives, we started designing and implementing PADE. We used post-WIMP guidelines [7,17,39] to consider feedback, reducing indirection, and integrating interactions and visual representations.

In designing the interaction techniques, we aimed to support data exploration by enabling analysts to create and combine visualizations based on quick and flexible interactions.

**Quick:** In aiming for quick interactions, we designed techniques that have a short interaction time span, and aimed to provide immediate results to interface actions rather than intermediary menus. The argument is that it allows users to quickly gain an overview of datasets and obtain valuable insights. By precomputing common aggregations based on a data cube model (see next section), the system provides quick results to complex queries. A second factor in aiming for quick interactions, was to design for an unbroken series of interaction steps. We designed the system such that results of actions open the possibility for new interactions.

**Flexible:** In aiming for flexible interactions, we created interaction techniques that allow for variation in data exploration. For example, there are many possibilities and ways of combining user interface elements to produce different outcomes and users' can accomplish many goals in several ways.

### 3.3 Technological need

Our inquiries showed that the analysts were able to perform data analyses on their own and had useful tools that supported them in doing so. However, their existing tools fell short when they needed to collaborate. In short, we observed break-downs in the collaboration between analysts. To bridge the collaboration gaps, we aimed to support this by implementing a system, which we describe next.

## 4 PATIENT ADMINISTRATIVE DATA EXPLORER

PADE provides a novel approach to querying multi-dimensional data and receiving visualization views as query results. The interaction techniques in the system support creating visualization views by combining, extending, and re-using existing visualizations. In designing the techniques, we aimed to support data exploration by enabling analysts to **create**, **modify**, and **combine** visualization views using quick and flexible interactions. In designing the interaction techniques, we aimed to support data exploration by enabling analysts to easily **create** (DG 1, 2), **modify** (DG 3), **extend** (DG 2, 3), **re-use** (DG 3), and **combine** (DG 3) visualization views (DG 4). The goals of the interaction techniques were to ease collaborative data exploration of large data sets. We designed the system such that visualization components and data fields can be dragged onto as many elements in the user interface as possible—this combination is driven by our goal of quick collaborative analysis (DG 2-3, 6). For all combinations, we reviewed how well they supported the required low-level visualization tasks (i.e. how people perform a task), and how these low-level tasks fit in the larger context of analyses, such as data discovery, exploration, and comparison (i.e. why people perform a task) [10]. When using PADE, analysts should be able to create many related visualizations. Thus, we focused the interaction techniques on changes **within views** and **between views**. We describe the interaction techniques using these categories.

The interaction techniques can be adapted to many visualization techniques. However, the analysts frequently used bar charts in their work. Therefore, we decided to use bar charts in our design and implementation. We return to this choice in the discussion.

### 4.1 Data model

Due to the nature of the rate foundation table (multidimensional, highly hierarchical, millions of rows), and our design goals (primarily DG 2), we chose to base the system's visualization and interaction techniques on a **materialized data cube model** [21]. This model facilitates fast slicing, drilling, and pivoting according to any of the tables' columns to enable quick data exploration and analysis, and thus enabled us to meet the design goals (primarily DG 9-11). We used the data model to facilitate combinations of user interface elements. The model consists of dimensions, levels, members, and measures, and is explained in Figure 2. **Dimensions** in data cubes are constructed on the basis of nominal data columns in a data table. For example, a date dimension might be constructed based on year, month, and day columns that are mapped to levels of the date dimension hierarchy. Instances of these levels are known as members. For example, a date dimension may contain year as a level, which contains 2017 as a member. Likewise, **measures** in data cubes are constructed on the basis of quantitative data columns in a data table. For example, a cost measure might be constructed based on a cost column. Measures contain aggregates of raw data columns, grouped by dimensions. Measures can be binned to facilitate histograms (e.g., a histogram of costs). Therefore, the data model can contain data fields, which are possible to use as both measures and dimensions. We refer to these as binned measures. We also selected **default values** for dimensions and measures (DG 1, 2). In contrast to approaches that primarily use single views for data exploration and analysis (e.g., [50]), we used the data cube model for supporting analysis with many visualization views.

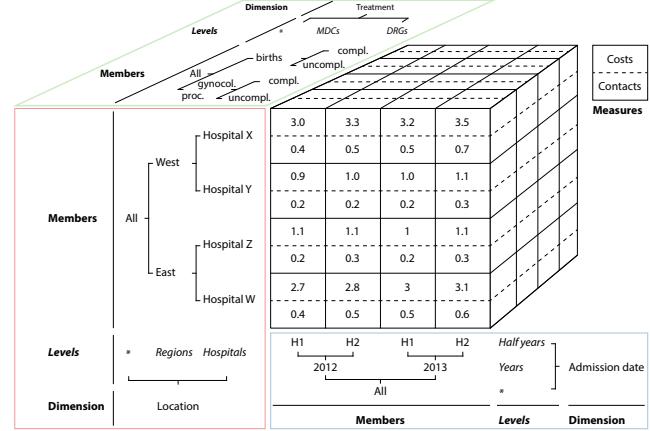


Figure 2: A simplified representation of the data cube model that shows three dimensions and two measures. The admission date (light green border), location (light red border), and treatment dimensions (light blue border) are shown together with their respective levels and members. The measures are cost and contact totals. The measures' values are given in millions (contacts) and billions (costs). The figure only shows measures broken down on the third level. The data model used had up to seven levels for some dimensions and measure aggregates were stored for all level combinations. For example, the dimension for admission diagnosis codes according to ICD-10 [55] contains four levels

### 4.2 Interaction concept

In PADE, access to data happens through a data field menu in the top part of the display (Figure 1, left). The **menu shows dimensions and measures** from the data cube model. Data fields can be dragged from the menu and released on relevant parts of the user interface, which are highlighted when a compatible field is dragged over it. We denote data fields that can be released on a part of the user interface, as compatible with the part. We opted for a **permanent** and **fixed** position of the data field menu. This solution requires few instructions to get started and thus works well for walk-up use (DG 1). **Views** are the main UI element of PADE and can be dragged with one finger. They show aggregate data in bar charts, where x-axes encode dimensions and y-axes encode measures (Figure 1, left).

In the following, we describe the interaction techniques in PADE. We then continue to describe each technique, discuss design alternatives, and open issues. The first techniques are simple but necessary for exploring data; the latter techniques are more complex. Figures 1 and 3–7 show the techniques as sketches; a video in supplemental material provides more detail and examples of use.

### 4.3 Creating views

To create a view, data fields can be dragged to the canvas (the background area), which results in a bar chart that shows the released data field (Figure 1, left). We designed view creation with a focus on speed and of ease of use, since creating a view is a necessary first step in most tasks, and thus frequently used. Because the dragged data field may represent both a dimension and a measure, the two possibilities provide slightly different results. A dragged dimension or binned measure result in a view that encodes the dragged field on the x-axis, whereas a dragged measure results in a view that encodes the dragged field on the y-axis. The y-axis is scaled according to:

$$\min_{x \in \{1, 5, 10\}} (x \times 10^{\lfloor \log_{10}(\text{bar}_{\max}) \rfloor})$$

where  $\text{bar}_{\max}$  is the maximum value for a data bar in the bar chart. This results in the axis maximum to be either 1 or 5 times ten to the

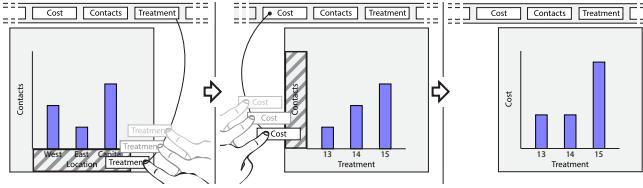


Figure 3: View axes are **configured** by dragging data fields to them.

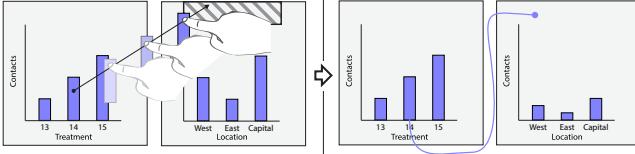


Figure 4: A view can be **filtered** by a data bar by dragging it to a view's filter area. Note the filter circle in the rightmost view.

n'th power. This satisfies three criteria: (1) the scale and tick marks are easily readable, (2) the likelihood that different views use the same scale is high, thus allowing analysts to compare views (**DG 9**), and (3) data encodings use much of the space within views. The other axis shows a default data field provided by the data model, thus minimizing interaction steps (**DG 2**). Alternatively, all data fields could be configured manually to ensure that analysts are aware of axis mappings. For the studied domain, we set the defaults as treatment location (dimension) and number of admissions (measure).

The quick and simple technique of obtaining high-level information is similar to Tableau. However, Tableau requires users to select both dimensions and measures. Aiming for quick, high-level overviews, we opted to define default measures and dimensions. This reduces the amount of interaction, potentially at the price of users' reduced awareness of selected encodings.

#### 4.4 Modifying views

In the following, we describe how views are modified in PADE.

**View Configuration:** To configure a view's axis, data fields can be dragged to them (Figure 3). This enables the most essential configurations of a view. Releasing the data field (re-)configures the axis, without relying on menus (**DG 2**). Dimension and measure fields are compatible with x-axes and y-axes, respectively. This provides the opportunity to alter views as needed, and to select alternatives to the view configuration.

Other work has facilitated view configuration by direct axis interaction (e.g., Sadana & Stasko [45]). Tableau also uses drag and drop to configure axes. Releasing data fields on axes facilitates replacement, while releasing them on the “shelf” provides access to more complex encoding possibilities. PADE facilitates data exploration through other means of interaction. There is thus less need to create complex visualizations within a view.

**View Filtration:** To filter a view, data bars can be dragged to another view's filter area (Figure 4). The technique allows for using views as filter palettes, thus supporting flexible exploration in other views (**DG 6**). Filters on one dimension are logically 'OR'ed and are shown as one circle (since range queries are not part of PADE, it is not possible to create two filters on the same dimension). Filters on different dimensions are logically 'AND'ed and are shown as different circles. Flicking up or down on a filter circle inverts the filter. Flicking left and right on the filter circle enables or disables the filter. This technique also works for view exploration, and both supports temporal juxtaposition (**DG 9**).

**View Synchronization:** To synchronize views' y-axis scales, a view can be dragged such that its side area overlaps another view's

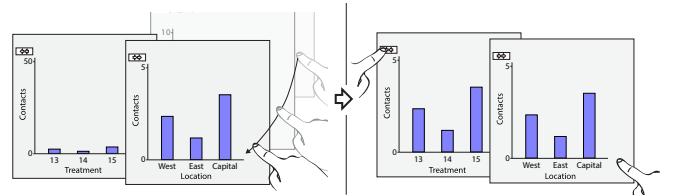


Figure 5: Two views' y-axis scales can be **synchronized** by dragging views such that their axes overlap and tapping a synchronize button.

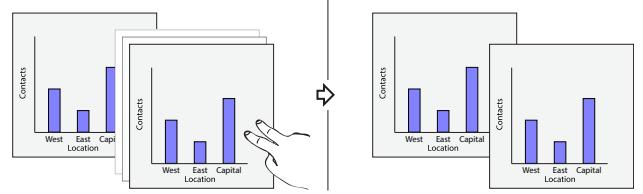


Figure 6: A view can be **cloned** by dragging a view with two fingers.

side area (Figure 5). When views that encode the same measure overlap, a synchronize button appears above the y-axes in both views. While holding onto the view, the view adopts the scale of the other view when tapping the synchronize button. This allows analysts to compare views side by side (**DG 9**) [19, 35], and offers a way to obtain consistent y-axis domains, as recently discussed [34, 37, 42].

We designed the technique with the aim to minimize unintentional synchronizations while arranging views. To reduce the likelihood of errors, this technique does not support changing the y-axis measure. The synchronization technique exemplifies how we considered the use of two hands in a single task. The hand used to position the view sets the context of the work done by the other hand [22].

#### 4.5 Creating new views from existing views

In the following, we describe how new views are created from existing views in PADE.

**View Exploration:** To explore parts of data in a view, data bars can be dragged to the canvas, resulting in a new view showing the dragged data (Figure 1, right), thus allowing drill down and more detailed data exploration (**DG 6, 11**). The technique works similarly to view filtration. The metadata necessary to provide a useful result is obtained from the data model that provides child members (e.g., 2017, September, or 22nd) at a level below the data bar (e.g., year, month, or day). PADE shows these child members on the x-axis in the child view. If no child members exist for the dragged member, the dragged data bar is shown in the new view. To add additional data bars' child members to the child view, these bars can be dragged to the child's filter area, making it possible to select multiple items from a view to analyze in more detail. A line represents the parent-child relationship from the parent data bar to the child view's filter area. Creating a series of views, where each child is the parent of another child, shows the exploration history (**DG 3, 5, 6, and 8**).

**View Cloning:** To clone a view, views can be dragged with two fingers (Figure 6). This enables continued exploration in a clone (**DG 6**), while the original view is preserved. Cloning works similar to drag, and leverages the added efficiency provided by chunking [13] drag and clone interactions. To rearrange a view and create a clone, a view is first dragged. Adding a second finger after positioning the original view, results in a clone operation. Continued dragging positions the cloned view. Thus, cloning provides a way to preserve a view in the middle of an analysis (**DG 4, 5**), to use two alternative approaches to a data analysis (**DG 3, 7**), or to split an analysis into two comparable branches (**DG 9**). For example,

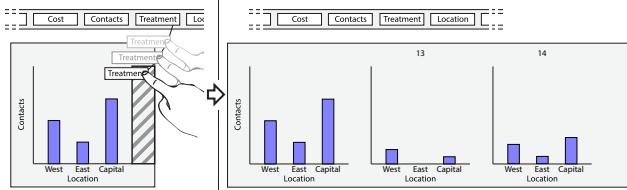


Figure 7: A view can be **exploded** by dragging data fields on them to create copies of the view, filtered for values of the released field.

in analysing the cost distributions of a hospital, the analysts might start from a view that encodes the number of patient contacts on the y-axis. Before configuring the y-axis to encode costs, they can clone the view to be able to refer back to the number of patient contacts. Because the idea of working with many views is integral to PADE, we chose to provide a quick method to clone views, and designed view cloning for single-handed two-finger operation. Our approach contrasts most other systems (e.g., Tableau, which offers a two-step process). Perhaps closest to PADE, Tobiasz et al. [51] used a clone technique based on dragging fingers from the visual representation of the information visualization pipeline.

**View Exploding:** To explode a view according to members of a data field, data fields can be dragged to a view's right-hand side to break down the original view by the dragged data field by its members (Figure 7). This is similar to small multiples (**DG 2, 10**). Releasing the data field generates views for each member of the dimension. The large display space allows member views of similar size and scale to the original view, which facilitates comparison. The area of the original view is increased to contain the original view, as well as the member views to the right of the original view. If view exploding creates more than four views, the members are shown in a 3-view wide scrollable list. The views hidden by scrolling are aggregated to the right of the list. Data bars from other views may be released on the explode area, just like data fields, to explode the view by child members of the aggregate represented by the bar.

## 4.6 Implementation and apparatus

We implemented PADE in Java using a combination MT4J [38] and the Prefuse data visualization toolkit [23]. The Prefuse renderers were ported to MT4J where they generate and update MT4J components. The data and data model were stored in an MSSQL server and MSSQL Analysis Services cube, respectively. We used Olap4J [41] to build data cube queries.

PADE works with different display sizes and touch systems. For prototyping and developing PADE, we used a 3 by 1 meter touch display at a resolution of 7680 by 3240 pixels (25 Mpixels), Figure 1 shows analysts using PADE on a 3840x2160 pixels, 1.9 by 1.0 meter (84") touch display [49]. While the 84" display is sufficiently large to demo the interaction techniques, the larger display allows collaborators to use the system to its full potential.

## 5 ANALYSIS SCENARIO

To explain how PADE might be used, we describe a scenario where a data scientist (Christine) and a medical specialist (Kamal) use PADE for collaborative analysis. The analysts discuss the age distribution of birthing women. Knowing that view *creation* allows them to quickly get high-level information about dimensions in the data cube, Christine first drags treatment types from the menu and releases it on the canvas. This creates a view that shows admissions across treatment groups. Kamal has background information about these groups, and quickly identifies group 14 (birth and labour). He *explores* this bar, which creates a new barchart of admissions per treatments in this group. By dragging patient age to the x-axis, Christine *reconfigures* the view to show age distribution in 10-year bins

across all admissions in group 14. Kamal notes the barchart *reconfiguration* shows age bins that seem wrong (for example, 210–219 years). Frequently seeing and handling data quality issues, Christine argues that they are seeing errors in the data set, and suggests to focus on the most common cases. Thus, Christine *explores* the 20–29 age bin, to obtain a barchart that shows a bar for each year from 20 to 29. Then, she drags the 30–39 age bin to the top of this barchart, to add each of these years as individual bars. They now have a clear sense of the age distribution in the entire data set. They agree to *explode* this view by releasing the treatment location onto the right part of the view to understand how the age of patients compares across regions. This exploded barchart provides the analysts with a barchart of age distributions for each of the five Danish regions. While Christine is new to Denmark, Kamal has lived in Denmark his entire life. He can thus fill in with knowledge about the regions.

From the exploded barchart, the analysts see that women in the metropolitan areas on average are more than 30 years old when giving birth, whereas they are less than 30 years old in rural areas. However, they also see a lot of variation—for example, many women in metropolitan areas give birth in their mid-twenties and many women in rural areas give birth in their mid-thirties. In the scenario, view *exploding* allowed the analysts to easily compare differences between regions, while still keeping the exploded view for reference, comparison, or for further analysis.

The video in supplemental material shows similar analyses.

## 6 DISCUSSION

In designing PADE, we wanted to enable analysts to create many views with little effort. We think a key strength of the interaction techniques in PADE is the direct mapping between what is shown in the system, the possible actions, the potential results, and the visual representation of this.

Some of the most interesting possibilities arise when **using several interaction techniques in combination**. For example, view cloning is simple when considered alone, but becomes powerful when used with the other techniques. View cloning also makes other techniques work better. For example, we chose not to allow x-axis configuration in views that have children, because the relation to the child views would get unclear. However, it is fast to create a view clone, and then reconfigure this to allow for further exploration. Additionally, view filtration influences child views. This means that an entire analysis trail can be filtered simply by filtering the parent view. This allows analysts to quickly construct comparative setups where one trail shows what is included in a filter and another what is left out, thus providing a good overview of the filtration. Similarly, filters can be turned on and off by interacting with the filter dot thus providing temporal juxtaposition [19], letting analysts understand filtering implications after constructing a view trail. The idea of **using existing visualization views to modify or create** other views is to our knowledge novel. While, view exploration and view filtration interaction techniques appear similar on the surface, the latter allows analysts to use parts of views to interact with other views.

Our goals of providing **quick and flexible** interaction deserves discussion. The quick and simple technique of obtaining high-level information by creating a view reminds of Tableau. The key differences are that PADE provides defaults to reduce the amount of interaction, potentially at the price of analysts' reduced awareness of selected encodings. In contrast, other systems facilitate view configuration by direct axis interaction. For example, Sadana & Stasko [45] used a data field list from which to choose a data field replacement on the axis. To keep the overall design consistent across PADE's interaction techniques, we only provide techniques based on dragging, which further allows for using data bars in place of data fields. The clone technique in PADE contrasts visualization systems that use copy and paste—a two-step interaction—for copying visualizations (e.g., Tableau). Although Lark [51] is an exception, Lark

bases cloning on interacting with the visualization pipeline, where PADE's technique offers low spatial and temporal offset [7] by using the view itself for interaction. Based on demoing the system to people, we consider the quick and flexible qualities to be important to evaluate. Based on Jo et al.'s [31] recent observations, we are curious whether our defaults might confuse the analysts. We also consider the connection between quick interactions and planning and execution to be interesting study objects.

PADE only supports a **single visualization type** (bar charts); we prioritized instead to make it work with large-scale data. However, while the system is limited to one visualization type, the ideas behind the interaction techniques are not, and might therefore be applied to other visualization techniques. To help analysts understand **relationships between views**, we designed the parent-child links used in PADE with inspiration from other work [16, 34, 35]. We have not discussed how PADE's model of view relations might be aligned with other visualization types. We acknowledge that aggregates in bar charts provide a natural grouping of data items (although other ways to select groups of data exist), and that the model chosen might not work as well to represent different kinds of relationships between other visualization types.

We expect PADE will help the analysts to collaboratively do complex data explorations quickly, some of which takes hours of trial-and-error in their current system. While the analysts might benefit from any kind of visualization, we see the aggressive creation and expansion of visualizations in PADE as a key benefit. For these reasons, it is **obvious future work to understand how the techniques might be used by the analysts**, and by other analysts. To this end, we plan to report on lab and deployment studies in the near future, but consider this to be beyond the scope of this paper.

## 7 CONCLUSION

We presented PADE, which combines information visualization, large displays, and touch interaction to support collaborative data exploration of large healthcare data sets. We think our work points to interesting possibilities for integrating data wrangling and analysis within the context of analysis meetings. Additionally, the **large display space** for which we designed PADE provides the opportunity of showing many views, and the system provides quick ways of creating them. In particular, the techniques facilitate comparison between multiple views in many ways, which is perhaps one of the most important analytic tasks supported by visualization [10]. PADE points to an interesting design area that leverages and relies on display space to provide interaction techniques that make use of all this space for visual analytics.

## ACKNOWLEDGMENTS

This work has been supported in part by the Danish Council for Strategic Research grant 10-092316 and the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 753816.

## REFERENCES

- [1] C. Andrews, A. Endert, and C. North. Space to Think: Large, High-Resolution Displays for Sensemaking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*, pp. 55–64, 2010.
- [2] C. Andrews, A. Endert, B. Yost, and C. North. Information visualization on large, high-resolution displays: Issues, challenges, and opportunities. *Information Visualization*, 10(4):341–355, 2011. doi: 10.1177/1473871611415997
- [3] B. A. Aseniero, T. Wun, D. Ledo, G. Ruhe, A. Tang, and S. Carpendale. STRATOS: Using Visualization to Support Decisions in Strategic Software Release Planning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15*, pp. 1479–1488. ACM, New York, NY, USA, 2015. doi: 10.1145/2702123.2702426
- [4] S. K. Badam, A. Mathisen, R. Rädle, C. N. Klokmose, and N. Elmquist. Vistrates: A component model for ubiquitous analytics. *IEEE transactions on visualization and computer graphics*, 25(1):586–596, 2018.
- [5] D. Baur, B. Lee, and S. Carpendale. TouchWave: Kinetic Multi-touch Manipulation for Hierarchical Stacked Graphs. In *Proceedings of ACM International Conference on Interactive Tabletops and Surfaces (ITS '12)*, pp. 255–264, 2012.
- [6] L. Bavoil, S. P. Callahan, P. J. Crossno, J. Freire, C. E. Scheidegger, C. T. Silva, and H. T. Vo. Vistrails: enabling interactive multiple-view visualizations. In *VIS 05. IEEE Visualization, 2005.*, pp. 135–142, Oct 2005. doi: 10.1109/VISUAL.2005.1532788
- [7] M. Beaudouin-Lafon. Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces. In *Proceedings of the 18th international conference on Human factors in computing systems (CHI '00)*, number 1 in 2, pp. 446–453. ACM Press, 2000. doi: 10.1145/332040.332473
- [8] H. Beyer and K. Holtzblatt. *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann, 1993.
- [9] A. Bezerianos and P. Isenberg. Perception of visual variables on tiled wall-sized displays for information visualization applications. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2516–2525, Dec 2012. doi: 10.1109/TVCG.2012.251
- [10] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–85, 2013. doi: 10.1109/TVCG.2013.124
- [11] K. Brodlie, A. Poon, H. Wright, L. Brankin, G. Banck, and A. Gay. GRASPARC-A problem solving environment integrating computation and visualization. In *IEEE Conference on Visualization, 1993. Visualization '93, Proceedings*, pp. 102–109, Oct. 1993. doi: 10.1109/VISUAL.1993.398857
- [12] J. Browne, B. Lee, S. Carpendale, N. Riche, and T. Sherwood. Data analysis on interactive whiteboards through sketch-based interaction. In *Proceedings of ACM international conference on Interactive tabletops and surfaces (ITS '11)*, p. 154. ACM Press, New York, New York, USA, 2011.
- [13] W. Buxton. Chunking and phrasing and the design of human-computer dialogues. In R. M. Baecker, J. Grudin, W. A. Buxton, and S. Greenberg, eds., *Readings in Human-Computer Interaction, Interactive Technologies*, pp. 494 – 499. Morgan Kaufmann, 1995. doi: 10.1016/B978-0-08-051574-8.50051-0
- [14] S. Carpendale. Evaluating Information Visualizations. In *Information Visualization: Human-Centred Issues and Perspectives*, chap. I., pp. 19–45. Springer Berlin Heidelberg, 2008.
- [15] S. Drucker, D. Fisher, R. Sadana, J. Herron, and M. Schraefel. TouchViz: A Case Study Comparing Two Interfaces for Data Analytics on Tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*, pp. 2301–2310. ACM, Paris, France, 2013.
- [16] C. Dunne, N. H. Riche, B. Lee, R. A. Metoyer, and G. G. Robertson. GraphTrail: Analyzing Large Multivariate, Heterogeneous Networks while Supporting Exploration History. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, pp. 1663–1672, 2012.
- [17] N. Elmquist, A. V. Moere, H.-C. Jetter, D. Cernea, H. Reiterer, and T. Jankun-Kelly. Fluid Interaction for Information Visualization. *Information Visualization*, 10(4):327–340, 2011. doi: 10.1177/1473871611413180
- [18] R. B. Fetter, Y. Shin, J. L. Freeman, R. F. Averill, and J. D. Thompson. Case mix definition by diagnosis-related groups. *Medical Care*, 18(2):i-53, 1980.
- [19] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–09, 2011. doi: 10.1177/1473871611416549
- [20] S. Gratzl, N. Gehlenborg, A. Lex, H. Pfister, and M. Streit. Domino: Extracting, comparing, and manipulating subsets across multiple tabular datasets. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2023–2032, 2014. doi: 10.1109/TVCG.2014.2346260
- [21] J. Gray, S. Chaudhuri, A. Bosworth, Andrew Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-

- Totals. *Data Mining and Knowledge Discovery*, 1(1):29–53, 1997. doi: 10.1023/A:1009726021843
- [22] Y. Guiard. Asymmetric division of labor in human skilled bimanual action: the kinematic chain as a model. *Journal of motor behavior*, 19(4):486–517, 1987.
- [23] J. Heer, S. K. Card, and J. A. Landay. Prefuse: A Toolkit for Interactive Information Visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*, pp. 421–430, 2005. doi: 10.1145/1054972.1055031
- [24] T. Horak, S. K. Badam, N. Elmquist, and R. Dachselt. When david meets goliath: Combining smartwatches with a large vertical display for visual data exploration. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI '18*, pp. 19:1–19:13. ACM, New York, NY, USA, 2018. doi: 10.1145/3173574.3173593
- [25] P. Isenberg, D. Fisher, M. R. Morris, K. Inkpen, and M. Czerwinski. An exploratory study of co-located collaborative visual analytics around a tabletop display. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST '12)*, pp. 179–186. Ieee, 2010. doi: 10.1109/VAST.2010.5652880
- [26] P. Isenberg, T. Isenberg, T. Hesselmann, B. Lee, U. von Zadow, and A. Tang. Data Visualization on Interactive Surfaces: A Research Agenda. *IEEE Computer Graphics and Applications*, 33:16–24, 2013. doi: 10.1109/MCG.2013.24
- [27] P. Isenberg, T. Zuk, C. Collins, and S. Carpendale. Grounded evaluation of information visualizations. In *Proceedings of the 2008 conference on BEyond time and errors novel evalUation methods for Information Visualization (BELIV '08)*, p. 1. ACM Press, New York, New York, USA, 2008. doi: 10.1145/1377966.1377974
- [28] M. R. Jakobsen, Y. S. Haile, S. Knudsen, and K. Hornbæk. Information visualization and proxemics: Design opportunities and empirical findings. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2386–2395, Dec 2013. doi: 10.1109/TVCG.2013.166
- [29] M. R. Jakobsen and K. Hornbæk. Up Close and Personal: Collaborative Work on a High-Resolution multitouch wall display. *ACM Transactions on Computer-Human Interaction*, 21(2), 2014. doi: 10.1145/2576099
- [30] W. Javed and N. Elmquist. ExPlates: Spatializing Interactive Analysis to Scaffold Visual Exploration. *Computer Graphics Forum*, 32(3):441–450, 2013. doi: 10.1111/cgf.12131
- [31] J. Jo, S. L'Yi, B. Lee, and J. Seo. TouchPivot: Blending WIMP & post-WIMP interfaces for data exploration on tablet devices. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17*, pp. 2660–2671. ACM, 2017. doi: 10.1145/3025453.3025752
- [32] S. Kandel, J. Heer, C. Plaisant, J. Kennedy, F. van Ham, N. H. Riche, C. Weaver, B. Lee, D. Brodbeck, and P. Buono. Research directions in data wrangling: Visualizations and transformations for usable and credible data. *Information Visualization*, 10(4):271–288, sep 2011. doi: 10.1177/1473871611415994
- [33] S. Kandel, A. Paepcke, J. M. Hellerstein, and J. Heer. Enterprise Data Analysis and Visualization: An Interview Study. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2917–2926, 2012. doi: 10.1109/TVCG.2012.219
- [34] S. Knudsen and S. Carpendale. View relations: An exploratory study on between-view meta-visualizations. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction, NordiCHI '16*, pp. 15:1–15:10. ACM, New York, NY, USA, 2016. doi: 10.1145/2971485.2971566
- [35] S. Knudsen, M. R. Jakobsen, and K. Hornbæk. An Exploratory Study of How Abundant Display Space May Support Data Analysis. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction (NordiCHI '12)*, pp. 558–567. ACM Press, New York, New York, USA, 2012. doi: 10.1145/2399016.2399102
- [36] B. Kondo and C. Collins. DimpVis : Exploring Time-varying Information Visualizations by Direct Manipulation. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2003–2012, 2014.
- [37] R. Langner, T. Horak, and R. Dachselt. VisTiles: Coordinating and combining co-located mobile devices for visual data exploration. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):626–636, 2018. doi: 10.1109/TVCG.2017.2744019
- [38] U. Laufs, C. Ruff, and J. Zibuschka. Mt4j-a cross-platform multi-touch development framework. *arXiv preprint arXiv:1012.0467*, 2010.
- [39] B. Lee, P. Isenberg, N. H. Riche, and S. Carpendale. Beyond Mouse and Keyboard: Expanding Design Considerations for Information Visualization Interactions. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2689–2698, 2012. doi: 10.1109/TVCG.2012.204
- [40] B. Lee, G. Smith, N. H. Riche, A. Karlson, and S. Carpendale. SketchInsight: Natural data exploration on interactive whiteboards leveraging pen and touch interaction. In *2015 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 199–206, Apr. 2015. doi: 10.1109/PACIFICVIS.2015.7156378
- [41] Open Java API for OLAP. <http://www.olap4j.org>, 2019. Accessed: 2019-08-21.
- [42] Z. Qu and J. Hullman. Keeping multiple views consistent: Constraints, validations, and exceptions in visualization authoring. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):468–477, 2018. doi: 10.1109/TVCG.2017.2744198
- [43] E. D. Ragan, A. Endert, J. Sanyal, and J. Chen. Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–10, 2015.
- [44] K. Reda, A. E. Johnson, M. E. Papka, and J. Leigh. Effects of Display Size and Resolution on User Behavior and Insight Acquisition in Visual Exploration. In *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems*, pp. 2759–2768. ACM Press, 2015. doi: 10.1145/2702123.2702406
- [45] R. Sadana and J. Stasko. Designing and implementing an interactive scatterplot visualization for a tablet computer. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14)*, pp. 265–272. ACM Press, New York, New York, USA, 2014. doi: 10.1145/2598153.2598163
- [46] R. Sadana and J. Stasko. Designing Multiple Coordinated Visualizations for Tablets. *Computer Graphics Forum*, 35(3):261–270, June 2016. doi: 10.1111/cgf.12902
- [47] R. Sadana and J. Stasko. Expanding Selection for Information Visualization Systems on Tablet Devices. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces, ISS '16*, pp. 149–158. ACM, New York, NY, USA, 2016. doi: 10.1145/2992154.2992157
- [48] S. Schmidt, M. a. Nacenta, R. Dachselt, and S. Carpendale. A set of multi-touch graph interaction techniques. In *Proceedings of ACM international conference on Interactive tabletops and surfaces (ITS '10)*, p. 113. ACM Press, New York, New York, USA, 2010. doi: 10.1145/1936652.1936673
- [49] SMART Board ® 8084i-G4 interactive flat panel. <http://smarttech.com/kb/170451>, 2018.
- [50] C. Stolte, D. Tang, and P. Hanrahan. Polaris: a system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65, 2002. doi: 10.1109/2945.981851
- [51] M. Tobiasz, P. Isenberg, and S. Carpendale. Lark: coordinating co-located collaboration with information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1065–72, 2009. doi: 10.1109/TVCG.2009.162
- [52] K. Vogt, L. Bradel, C. Andrews, C. North, A. Endert, and D. Hutchings. Co-located Collaborative Sensemaking on a Large High-Resolution Display with Multiple Input Devices. In *Proceedings of Interact 2011*, vol. II, pp. 589–04, 2011.
- [53] J. Walny, B. Lee, P. Johns, N. Henry Riche, and S. Carpendale. Understanding Pen and Touch Interaction for Data Exploration on Interactive Whiteboards. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2779–2788, 2012. doi: 10.1109/TVCG.2012.275
- [54] C. Weaver. Is coordination a means to collaboration? In *Proceedings - Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization, CMV 2007*, pp. 80–82, 2007. doi: 10.1109/CMV.2007.15
- [55] International Statistical Classification of Diseases and Related Health Problems, 10th Revision (ICD-10). World Health Organization, 1992.
- [56] E. Zgraggen, R. Zeleznik, and S. M. Drucker. Panoramicdata: Data analysis through pen & touch. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2112–2121, Dec 2014. doi: 10.1109/TVCG.2014.2346293