Scribble Query: Fluid Touch Brushing for Multivariate Data Visualization

Matthias Nielsen
Alexandra Institute
Denmark
matthias.nielsen@alexandra.dk

Niklas Elmqvist
University of Maryland
College Park, MD, USA
elm@umd.edu

Kaj Grønbæk
Department of Computer Science
Aarhus University, Denmark
kgronbak@cs.au.dk

ABSTRACT
The wide availability of touch-enabled devices is a unique opportunity for visualization research to invent novel techniques to fluently explore, analyse, and understand complex and large-scale data. In this paper, we introduce Scribble Query, a novel interaction technique for fluid freehand scribbling (casual drawing) on touch-enabled devices to support interactive querying in data visualizations. Inspired by the low-entry yet rich interaction of touch drawing applications, a Scribble Query can be created with a single touch stroke yet have the expressiveness of multiple brushes (a conventionally used interaction technique). We have applied the Scribble Query interaction technique in a multivariate visualization tool, deployed the tool with domain experts from five different domains, and conducted deployment studies with these domain experts on their utilization of multivariate visualization with Scribble Query. The studies suggest that Scribble Query has a low entry barrier facilitating easy adoption, casual and infrequent usage, and in one case, enabled live dissemination of findings by the domain expert to managers in the organization.

Author Keywords
Touch interaction, interaction techniques, information visualization.

ACM Classification Keywords
H.5.2 [Information Interfaces and Presentation]: User Interfaces—Input devices and strategies, H.5.2 [Information Interfaces and Presentation]: User Interfaces—Interaction Styles

INTRODUCTION
Touch interaction has been shown to be intuitive, familiar, and easy to learn (Harrison et al. 2014). As a result, touch dis-plays have been deployed in more than 2 billion smart-phones, tablets, computers, and large displays all over the world (eMarketer 2014), and fluid touch interaction such as scribbling, swiping, and shape writing has found high preference with users everywhere (Nguyen et al. 2012). This represents a potent challenge for designers of interactive data visualizations because as users have come to prefer touch-based interfaces so will users expect to be able to just as easily use their touch-screen devices to interact with data visualizations. Conventionally interactive data visualizations are designed for mouse and keyboard interaction, however, fingers lack the unambiguity of keystrokes on keyboards and the pixel-precision of mice-pointing. Therefore, touch-based interaction techniques require robust and fault-tolerant interaction that leverages the strengths of touch input and mitigates its weaknesses.

In the information visualization (InfoVis) field a commonly adopted approach in recent years' interaction research has been to apply conventional touch gestures (tap, drag, swipe, pinch, etc.), to trigger contextualized commands specific to a single visualization (e.g. a swipe gesture on a bar chart orders the bar chart numerically) (Baur et al. 2012; Drucker et al. 2013; Kosara 2010; Kosara 2011; Nielsen et al. 2013; Rzeszotarski et al. 2014; Schmidt et al. 2010). Others have explored mapping touch input to control interaction techniques designed for mouse-input, such as Kosara's (Kosara 2010; Kosara 2011) work for indirect touch brushing and Nielsen et al.'s (Nielsen et al. 2013) work on direct touch brushing. However, most such approaches often fail to take full advantage of the touch medium's unique capabilities for fluent and non-abstruse interaction because they map touch-input to specific commands or existing interaction techniques thus introducing indirection between the touch input and resulting action.

In this paper, we extend the body of touch-based interaction techniques for visualization by proposing Scribble Query: freehand fluid finger input Scribbling for flexible and advanced querying of multivariate datasets in interactive visualizations. Specifically, we extend on the work investigating using touch-sketched paths on line chart visualizations of time-series data for filtering lines based on trajectory-matching (Eichmann et al. 2015; Holz et al. 2009; Wattenberg 2001). Like these contributions, we seek to leverage touch-input by interpreting user input as-is and not converted it into general commands or mapped to interaction techniques. We have found inspiration for Scribble Query in the casual drawings—scribbles—that people draw casually in tablet drawing applications or on the back of napkins. An example use case of Scribble Query is shown in Figure 1. The Scribble Query technique goes beyond the previously described approaches of mapping touch input

OzCHI '16, November 29-December 02, 2016, Launceston, TAS, Australia
 ISBN 978-1-4503-4618-4/16/11…$15.00
DOI: http://dx.doi.org/10.1145/3010915.3010951

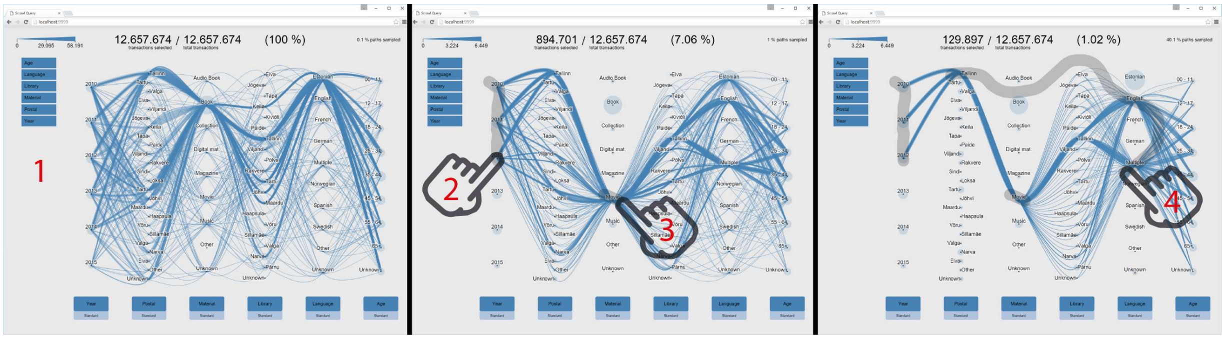

Figure 1. Screenshot storyboard of a use case for using Scribble Query to create touch brushes on a multivariate data visualization of over 12 million library loans based on (Nielsen et al. 2015). (1) shows the visualization's initial state. (2) shows a Scribble Query making a consecutive range query. (3) shows a single element selection using Scribble Query. (4) shows a multidimensional, non-consecutive query using Scribble Query, while crossing dimensions without making selections.

to specific commands, converting existing interaction techniques to touch, or performing trajectory-matching queries. Instead, Scribble Query is based on a design exploration of how touch interaction can be leveraged for fluid interaction in visual data analysis. The technique involves using one or several touches to specify one or several filters using a single gesture of a finger, as depicted in Figure 1. A Scribble Query is directly represented in the visualization as a trace of the freehand sketched input, thus persisting the user's input.

The contributions of this work are the following: (1) outlining and discussion of our design space for developing touch interaction techniques for interactive multivariate data visualizations; (2) the Scribble Query interaction technique for touch-based free-hand filtering; (3) the application of Scribble Query to a parallel coordinate inspired visualization of multivariate data; and (4) observations from two deployments of Scribble Query with a multivariate data visualization with a total of five domain experts.

The remainder of this paper is structured as follows: First, we review the literature in relation to Scribble Query, focusing on touch interaction with interactive data visualizations. Second, we introduce our design space developing for touch interaction for multivariate data visualizations, including the design rationale for multivariate data visualizations that apply touch interaction, before we elaborate and discuss the Scribble Query interaction technique and discuss its limitations. Third, we report on our findings from two deployment studies investigating Scribble Query in use with domain experts from two domains. Fourth, we report observations from the deployments of Scribble Query. Fifth, we discuss our observations and their generalizability, applying Scribble Query to other types of visualizations, and the limitations of the technique. Finally, we outline future work and conclude the paper with a summary of our contributions.

BACKGROUND

In this section, we position Scribble Query in relation to direct manipulation and post-WIMP interaction, recent work in touch interaction with interactive data visualizations, as well as general multivariate visualization.

Direct Manipulation and Post-WIMP Interaction

Our approach is inspired by direct manipulation (Shneiderman 1983), which is based on the idea of allowing users to perform direct, iterative interactions on continuously-updated data items rather than through complex and abstract syntax. To operationalize this approach, we look to instrumental interaction (Beaudouin-Lafon 2000), which defines three properties—indirection, integration, and compatibility—to operationalize design and analysis of interaction instruments in post-WIMP interfaces. Specifically, we seek to lower indirection and increase integration and compatibility for creating queries. Finally, we seek inspiration from fluid interaction (Elmqvist et al. 2011) by designing an interaction technique for interactive data visualization that promotes flow, supports direct manipulation of domain objects, and minimizes the gulfs of interaction (Norman et al. 1986).

Recent years has seen an increased focus on challenging conventional mouse and keyboard interaction in the InfoVis community. This is exemplified by recent contributions on InfoVis specific interaction design considerations (Lee et al. 2012), as well as a data visualization-specific interaction model (Jansen et al. 2013), extending upon instrumental interaction, for visualizations utilizing new interaction technologies for data visualization.

Touch Interaction for Data Visualizations

The increasing availability of touch mobile devices has made touch interaction for visualization a hot topic in recent years. However, creating touch-based visualizations is often challenging. The simplest approach converts mouse interaction to touch, such as in Rizzo (Vlaming et al. 2010), a multi-touch interaction technique that allows users to perform mouse precision input on multi-touch devices, enabling applications that are dependent on high precision input— visualizations especially—to work on touch devices.

However, fully harnessing touch requires customizing common touch input gestures to each specific visualization. Schmidt et al. (Schmidt et al. 2010) propose touch gestures for node-link diagrams, TouchWave (Baur et al. 2012) provides single and multi-touch gestures for interacting with stacked graphs, TouchViz (Drucker et al. 2013) adapts touch gestures to bar charts, Kinetica (Rzeszotarski et al. 2014) uses multi-touch interaction to create virtual instruments for interacting with scatterplots and histograms, and Sadana et al. develop multi-touch gestures for scatterplots (Sadana et al. 2014). Common for these gestures, techniques, and interfaces are that they rely on touch gestures that quickly have become standardized (swipe, pinch, dwell). Specifically, for parallel coordinates, recent work includes performing indirect multi-touch brushing (Kosara 2010; Kosara 2011), using a computer touchpad for input, and direct single-touch for filtering (Nielsen et al. 2013). These touch gestures are then mapped to visualization-specific commands related to the objects that are interacted with; e.g. dragging the vertical axis in TouchViz (Drucker et al. 2013) orders the bar chart. There are, however, exceptions, like DimpVis (Kondo et al. 2014), which utilizes fluent, continuous touch input to explore a temporal dimension in a visualization.

Sketching, or scribbling, touch or pen input seems to promise a more engaging interaction paradigm. Using sketching as input on touch-enabled devices has been explored by Wattenberg (Wattenberg 2001), Ryall et al. (Ryall et al. 2005), and Holz et al. (Holz et al. 2009); the latter matches line trajectories, within some margin, in time-series visualizations and filters based on the match. Eichmann et al. (Eichmann et al. 2015) expand upon this work and investigates users' perception of tolerance of matching line trajectories to sketched input. Furthermore, scribbling input for information visualization has also been utilized in NapkinVis (Chao et al. 2009) and SketchInsight (Lee et al. 2015), which both explore the use of sketched gestures to create manipulate visualizations. Brownee et al. (Browne et al. 2011) and SketchStory (Lee et al. 2013) further explores using sketched input on interactive whiteboards for creating visualizations by drawing components of visualizations (e.g. bars in a bar chart) to visualize data. SketchSliders (Tsandilas et al. 2015), on the other hand, explores scribbling ad hoc widgets on tablet devices in order to explore data visualizations on large wall-mounted displays.

Our deployed implementation of Scribble Query (see Figure 1) is combined with an adapted parallel coordinates visualization (Inselberg et al. 1991), akin to (Nielsen et al. 2015). Furthermore, our implementation expands on the work by Apitz et al. and Perin et al. on the crossing interaction paradigm (Apitz et al. 2008; Perin et al. 2015), because a Scribble Query instantiates queries based whether and where a Scribble Query intersects, or crosses, a dimension, as can be seen in Figure 1 gesture (4)

In Scribble Query, we seek to look beyond conventional touch gestures and let naturalistic fluid scribbling interaction take precedence over point and click instruments for interaction by maintaining a high affinity between a user's interaction and the resulting query. Furthermore, we seek to utilize freehand naturalistic scribble input for directly querying data visualizations.

Multivariate Data Visualization

As mentioned previously, we have combined our deployed implementation of Scribble Query with a parallel coordinates-like visualization. This both relates our work to multivariate data visualizations in general as well as interaction techniques for such visualizations.

For datasets with more dimensions, a common approach is to generate several small visualizations and organize them side by side, a technique often called small multiples (Tufte 1991). For example, multiple scatterplots can be grouped into so-called scatterplot matrices (SPLOMs) (Becker et al. 1987). Elmqvist et al. propose using SPLOMs as an overview for navigating and exploring multivariate data (Elmqvist et al. 2008). However, multiple views imply that it can become challenging to correlate data in one view with another. Coordinated and multiple views (Roberts 2007) and brushing and linking (Becker et al. 1987) are general approaches for updating and highlighting selected items in other views to make understanding small multiples easier. The idea of brushing can be extended to virtually any type of data and visualization, e.g. Timeboxes (Hochheiser et al. 2004), which is an interaction technique for querying time-series data by constructing modifiable rectangular boxes similar to brushes, enabling constraining data on multiple dimensions.

Some visualization techniques are designed specifically to represent multiple dimensions in the same view. As a case in point, parallel coordinate plots (Inselberg 1997; Inselberg et al. 1991) visualize high-dimensional data by representing data dimensions as axes organized in parallel, transforming data points into polylines connecting the axes. Parallel sets (Kosara et al. 2006) extends parallel coordinates to accommodate very large datasets of categorical data or categorized continuous data using aggregated bands. Combining the two, PivotViz (Nielsen et al. 2015) plots high-dimensional categorical data as paths of varying thickness and opacity on axes with evenly spaced ticks.

Finally, current visual analytics systems for multivariate data connect multiple visualization views placed on an infinite canvas with data connections between the views. Examples of such so-called data flow systems include DataMeadow (Elmqvist et al. 2007), GraphTrail (Dunne et al. 2012), and ExPlates (Javed et al. 2013). Common between all of them is that they allow users to create advanced multidimensional data queries that update as one view along the query path is brushed and filtered.

Scribble Query expands on this existing work because it combines touch-scribbled input with multivariate data visualization in a crossing interface that enables users to create advanced queries of multivariate data.

DESIGNING FREEHAND TOUCH INTERACTION

In this section we will outline our design space for designing and developing touch interactions for fluid interactive visual data analysis for non-programmers. We define our design space by distinguishing between novel vs. conventional touch interaction techniques and fluent vs touch-command input. We have focused our investigation on the brushing interaction, which we have both adapted for direct-touch concurrent multi-touch interaction, as well as a completely redesigned interaction technique – Scribble Query – designed with direct-touch-interaction in mind.

In order to explore the design space of touch brushing on multivariate data visualization, we first developed a touch friendly adaption of parallel coordinates (Inselberg 1997; Inselberg et al. 1991) and parallel sets (Kosara et al. 2006) for visualizing large datasets. The visualization (Figure 1), visualizes discretised multivariate data by plotting dimensions in the data as vertical parallel axes, with a circle for each category on the individual axis, and uses paths to visualize inter-dimensional relations in data. Discretization, or data binning, is the process of counting frequency of occurrences in a number of bins, and it is a technique that is commonly applied when creating histograms of continuous data (e.g. age groups, months and years in time-series data, and profit margins in intervals). We used discretised data to afford direct-touch input, because discretised categories are easier to distinguish and deliberately select than continuous data entries, which can be difficult to distinguish visually as well as interactively. Thereby we were able to apply Wang's et al. (Wang et al. 2009) guideline for size of elements for finger-touch interaction by ensuring that the discretised categories were placed with enough distance between to facilitate accurate selection using direct-touch.

Novel vs Conventional Touch Interaction Techniques

We distinguish between novel and conventional touch interaction techniques based on to which degree the interaction technique is designed from scratch for touch input or whether it is an existing technique that is adapted to work with touch input. From our review of related work, DimpVis (Kondo et al. 2014) is a good example of the former and Kosara's indirect-touch brushing on parallel coordinates (Kosara 2010; Kosara 2011) is a good example of the latter.

By creating this distinction, we do not mean to imply that one approach per definition is better than the other. One can easily imagine a setting where user familiarity with an interaction technique weighs highest and therefore adapting the interaction technique for touch input (if touch input is required) is the best option. With this mindset, our first approach was to develop a touch-adapted version of conventional brushes for the parallel coordinates visualization. These brushes support interactions conventional for brushes—selection of single categories and ranges, and moving brushes. Furthermore, our touch-adapted brushing supports merging two or more brushes if they overlap, multi-touch input for specifying extent, and finally more advanced indirect single and multi-touch input. This way we emulated, and extended a bit, the current state-of-the-art of multi-touch brushing.

However, although we prior to our exploration had considered adapting brushes for concurrent multi-touch and direct-touch input as a novel interaction technique, we were unconvinced that we had sufficiently explored our design space. Therefore, we decided to start from scratch and set out to design a direct-touch interaction technique that took offset in the elemental feature that touch-enabled displays afford: pointing and touching objects of interest. Therefore, we based Scribble Query on the easily approachable interaction of direct-touch drawing, sketching, or scribbling known from drawing applications on touch-devices. We will elaborate and discuss this in the next main section.

Fluid Touch vs. Touch-command Interaction

We furthermore distinguish between fluid touch and touch-command interaction in our design space. We base fluid touch interaction on Elmqvist's et al. notion of fluid interaction (Elmqvist et al. 2011), and we use it to describe interaction that is consistent and adaptable and scalable with user sophistication. We contrast fluid interaction to touch-command input such as mapping an established touch gesture (e.g. swipe) to trigger a specific command in an interactive visualization (e.g. reordering a bar chart). Holz' et al. relaxed selection technique (Holz et al. 2009) is an example of the former, and TouchViz (Drucker et al. 2013) is an example of the latter.

When we set to explore our design space we imagined that adapting brushes to multi-touch direct-touch input would facilitate fluid interaction, because the brushing interaction technique is commonly applied to multivariate visualizations (Becker et al. 1987; Elmqvist et al. 2008; Hochheiser et al. 2004; Roberts 2007; Theron 2006). However, initial deployments of our multi-touch adapted brushes, together with the aforementioned parallel coordinates inspired visualization, indicated that the construction of brushes and subsequent interaction with brushes inhibited fluid interaction. This was because that after creating a brush, it becomes an instrument when the user modifies the brush in subsequent interactions.

Therefore, we decided to redesign the functionality of brushes as simplified interaction technique that incorporated similar expressiveness, yet retained a simplified conceptual model of interaction. We discuss these properties of Scribble Query in the upcoming section.

SCRIBBLE QUERY: FREEHAND TOUCH BRUSHING

Scribble Query is designed for fluid touch brushing on multivariate data visualizations, applying a simplified and consistent interaction paradigm. The central principle in Scribble Query is that a user scribbles a path on a visualization rendered on a touch-enabled display. The scribbled path filters data as it crosses and intersects axes and points in the visualization, and the path thus becomes both the visual representation of the query and a persistence of the user's interaction. While a Scribble Query is being created, the visualization is continuously updated as the Scribble Query intersects with categorical or discretised items on visualized data dimensions. In Figure 1, a use case of a sequence of Scribble Query interactions are performed to analyse library loan data. In this use case, three Scribble Query strokes are performed sequentially by a user to rapidly perform a sophisticated

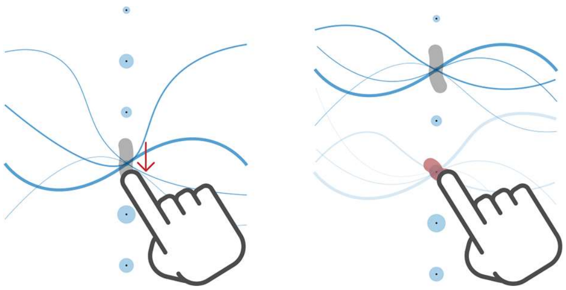

Figure 2. Selecting single category (left) and removing a Scribble Query with a single tap (right)

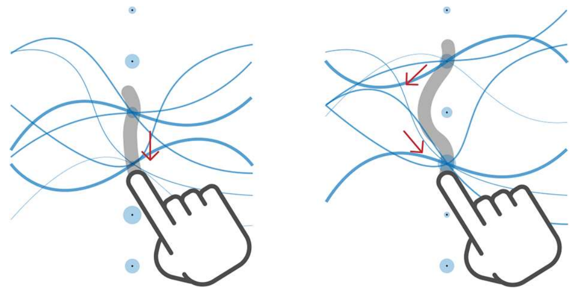

Figure 3. Selecting a range of categories (left). Selecting a non-consecutive set of categories (right)

query of a multivariate dataset. The central points in this use case are the consistency and flexibility of Scribble Query. Consistency, because all Scribble Query strokes in Figure 1 are made with the same elemental interaction – a touch-down and drag movement that filters as objects of interest are crossed. Flexibility, because despite the consistent interaction, a Scribble Query can scale and adapt with the sophistication of user queries, ranging from single select (Figure 1 (3) and Figure 2 (left)) to high-dimensional queries (Figure 1 (4) and Figure 4).

In this section, we elaborate the functionality of the Scribble Query interaction technique, we discuss it in terms of its interaction paradigm based on direct fluid interaction and direction manipulation, as well as its consistent conceptual model of interaction. Furthermore, we discuss Scribble Query and its application in our parallel coordinates inspired visualization as a crossing-inspired interaction technique for fluently scribbling queries in interactive multivariate data visualizations.

Scribble Query Functionality

Whenever a Scribble Query intersects a category on any axis in the multivariate data visualization presented in Figure 1, the visualization is filtered to only show lines intersecting this category. Multiple Scribble Query strokes can be added to the visualization in any combination as overlapping Scribble Query strokes only adds the same filter once. As a Scribble Query is being scribbled, the visualization is updated continuously as new categories are crossed and lines intersecting with this category are added to the visualization instantaneously.

Selecting a single category (Figure 3Figure 2 (left)) is performed by scribbling a short stroke covering a single element on an axis in the multivariate visualization. This is the simplest query that can be performed with Scribble Query and all interactions covered in remainder of this section can be constructed as a series of selections of single categories.

Removing a Scribble Query (Figure 2 (right)) is performed by tapping a single time on a Scribble Query. When a Scribble Query is removed, all lines intersecting the category are removed from the visualization. Unless no Scribble Query brushes are left, then all lines are shown in the visualization.

Selecting a range (Figure 3 (left)) is performed by scribbling a path intersecting multiple categories on an axis. A Scribble Query can cover any number of categories including all categories on an axis.

Selecting a non-consecutive set (Figure 3 (right)) is performed by simply scribbling around a category that should not be included in the selection.

High-dimensional filtering (Figure 4) is performed by initiating a Scribble Query on any axis, selecting one or more categories, and extending the Scribble Query onto other axes to select more categories. A Scribble Query can intersect all axes multiple times in any order. If the user does not intend to select categories on a specific axis, the Scribble Query can be drawn above, under, or between (and not intersecting) two categories.

Modifying a Scribble Query (Figure 5) is performed by dragging on a Scribble Query. Categories neighbouring the point where the dragging was initiated locks the Scribble Query. This can be used to unselect a category from a high-dimensional Scribble Query if the dragging is initiated on a selected category. Modifying a Scribble Query is not something we have implemented. However, it could be a possible addition if future evaluations indicate a need for such functionality.

Interaction Paradigm

Scribble Query allows users to use freehand touch input to create queries on data visualizations. Besides the effect of a Scribble Query on the underlying interactive visualization, the interaction in Scribble Query is based on the principle of direct manipulation (Shneiderman 1983). This is because creating a Scribble Query is a process of scribbling a stroke, and does not involve creating virtual instruments such as brushes, and therefore the interaction with the interaction technique itself embodies direct manipulation. Furthermore, removing, redoing, and adding a Scribble Query is a simple process of tapping and re-scribbled, thus facilitating reversible and rapid incremental querying.

As mentioned previously, we sought inspiration from the casual, yet engaging, interaction in scribbling/doodling applications commonly available on tablet devices, in the design of Scribble Query. The functionality of such applications— creating drawings or doodles using fingers as input on a touch-screen—is characterized by a low entry barrier and a encompasses many of the properties of fluid interaction (Elmqvist et al. 2011). Besides supporting direct manipulation, as discussed above, Scribble Query assists users in bridging Norman's

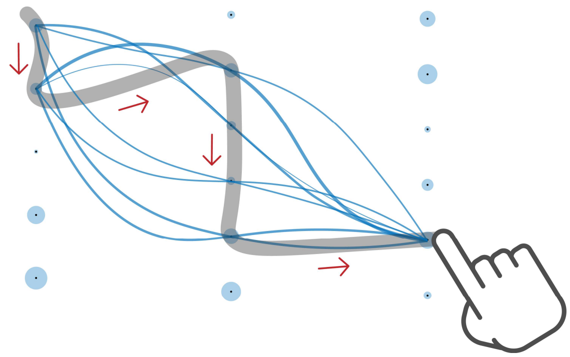

Figure 4. Performing an advanced, high-dimensional Scribble Query across multiple data dimensions.

gulfs of evaluation and execution. It supports bridging the gulf of evaluation because the interaction of a user is directly persisted as a stroke tracing the user's movement of finger(s) across the interface, thus retaining a high affinity between the user's interaction and the resulting Scribble Query. Furthermore, Scribble Query assists in bridging the gulf of execution because it has a consistent conceptual model of interaction, as we will elaborate on in the next section.

Consistent Conceptual Model of Interaction

Because the Scribble Query functionality follows a small simplified set of actions (Figure 2 through Figure 5), Scribble Query imposes a consistent conceptual model of interaction on users. This is achieved because the same basic interaction—scribbling—allows users to perform queries ranging from selection of single categories (Figure 3) over non-consecutive selections (Figure 4) to high-dimensional queries (Figure 5). Thereby, a user's Scribble Query interactions can evolve as the user becomes either better capable of, or more interested in, performing increasingly advanced queries.

An essential part of the conceptual model of Scribble Query is that it avoids constructing dispensable virtual instruments for interaction, which can introduce indirection in interaction if the consequence of interacting with an instrument occurs with an offset from where the interaction is performed (Beaudouin-Lafon 2000). The only time a Scribble Query becomes an instrument for interaction is when the user deletes it by single-tapping it (Figure 2 (right)), or when the Scribble Query is modified by dragging (Figure 5). This way, Scribble Query does not impose a complex set of instructions, nor does it require that the user alternate between one set of interactions when constructing a query and another set of interactions when modifying a query, as is commonly the case with brushes.

Crossing-based Interaction

Scribble Query also follows the crossing interaction paradigm (Apitz et al. 2008), which is designed to rethink conventional mouse-based individual interactions with fluent sequential interactions crossing multiple interface elements. Scribble Query as a crossing-based interaction is demonstrated in the interaction sequence in Figure 1. First, in Figure 1 (2) and (3), the Scribble Query exhibits basic crossing when it is used to filter a range (2) and a single element (3). The Scribble Query in Figure 1 (4)

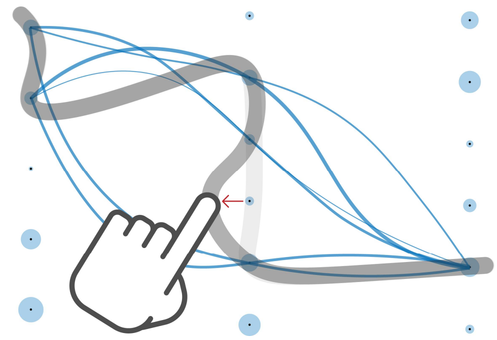

Figure 5. Modifying a Scribble Query by dragging.

uses crossing when it crosses dimensions without making selections, as well as when it performs non-consecutive filtering when it crosses non-neighbouring points on a dimension. This demonstrates two types of crossing using Scribble Query intentional crossing (intersecting categories on axes to select data) and non-crossing (crossing axes without intersection categories). However, there is another intentional non-crossing functionality, namely that crossing, intersecting, following paths between axes does not perform any filtering in the visualization. Our first version of Scribble Query was implemented to perform filtering by following paths, in a way similar to Holz's et al. relaxed selection techniques (Holz et al. 2009). However, through small-scale test we found that such filtering was counter-intuitive, which perhaps is because that lines in parallel coordinates visualizations only shows relationships between data points on axes, and as such carries information secondary to information on the axes.

Limitations of a Scribble-based Approach to Querying

The main limitations of using finger-based touch-scribbling for querying data visualizations are the relatively low accuracy of finger touch-input (compared to mouse-input) and the potential occlusion of parts of the visualization by the user's hand or fingers.

This is a limitation that should be handled by interaction technique and interface (in this case visualization). Through their in-depth investigation of finger-input on touch-screens, Wang et al. (Wang et al. 2009) conclude that interface elements meant for touch input should have a physical size of at least 11.52 mm to facilitate users to accurately target an element. We have utilized this guideline in the design of our visualization of multivariate by discretising data, which enables us to maintain adequate spacing between categories on axes. Discretising data does come at a loss of detail, however, at the same time it alleviates over-plotting issues, which is often occur when visualising large datasets.

Furthermore, occlusion caused by a user's fingers or hands, also discussed by Wang et al. (Wang et al. 2009), is a considerable concern as it is a fundamental property of finger-touch interaction. In Scribble Query, we attempt to alleviate issues caused by occlusion through the simple conceptual model of interaction. Specifically, because all queries can be performed by series of single selections (Figure 2 (left)), the user can construct the query step-by-step. Moreover, removing and redoing a

Scribble Query is a simple process, in case a Scribble Query is scribbled erroneously.

Using a pen for touch interaction could be considered an alternative to finger-touch input. However, using a pen causes indirection in the interaction and not all users will have a pen available meaning the interaction technique would be usable on less devices. Scribble Query could be performed using a pen nonetheless if the pens touch input is mapped to finger touch input.

Implementation Notes

Discussing all technical details of the implementation is outside the scope of this paper; here we summarize the issues most relevant to the Scribble Query technique.

Scribble Query and our touch-friendly visualization for large datasets is implemented in the JavaScript programming language and developed for use in common web browsers. Our implementation uses the D3 (Bostock et al. 2011), JQMultiTouch (Nebeling et al. 2012), jQuery, and Crossfilter JavaScript libraries to handle, visualize, and interact with data in the SVG format. Due to potentially very large datasets, we have implemented subsampling of data entries to be rendered as well as progressive rendering for retaining browser responsiveness. This is necessary because the visualization is implemented as SVG, which are DOM elements and inserting or manipulating many DOM elements can stall even modern browsers on powerful computers.

Our implementation is organized into three modules— a data management module, a visualization module, and an interaction module—that are decoupled and coordinated through a simple protocol. When a user scribbles a query, the interaction module translates user input into a stroke and communicates the stroke's axes crossings to the data management module, which in turn filters the underlying dataset. On completion, the data module triggers the visualization module to render the newly filtered data state. Furthermore, when a Scribble Query finalised, it becomes ready for interaction, and the user can remove it (Figure 2 (right)), triggering a similarly update cycle.

This modularized structure is important in order to easily develop and experiment with interaction techniques as well as rapidly exchanging the input data. In practise this means that the only difference between our implementation of multi-touch direct-touch brushes and Scribble Query is different interaction modules. Furthermore, we have developed Scribble Query (as well as our multi-touch enabled direct-touch brushes) to work with up to 10 concurrent touch inputs, meaning many simultaneous interactions are possible.

DEPLOYMENTS OF SCRIBBLE QUERY

We have deployed Scribble Query with versions of the multivariate data visualization shown in Figure 1 with domain experts from two organizations. The first is an orderly manager from a large hospital who needs to analyse hundreds of thousands of orderly tasks across organization, urgency, time, type of task, etc. The second is four customer relations employees from a large online retailer who need analyse tens of thousands of sales records across customer satisfaction score, delivery company, region, time, price type, etc.

All domain experts were used to analyse data as part of their job function: a customer relations employee would e.g., based on customer contact, like to investigate causes for unusual high customer dissatisfaction amongst customers in a specific region, and the orderly manager would e.g. like to investigate whether certain departments at the hospital would request unnecessarily many urgent tasks. Two of the employees from the online retailer had experience with retrieving data needed for analysis from databases etc., while the two other employees and the orderly manager relied on colleagues for retrieving data and preparing data for analysis in software applications (commonly Microsoft Excel); e.g. creating pivot tables or simple dashboards.

In this section we briefly report on excerpts from the deployments of Scribble Query and the accompanying multivariate visualization. Both deployments used Scribble Query and the same interactive multivariate data visualization closely resembling the one depicted in Figure 1; they differed only in terms of domain specific datasets (sales data and orderly task data).

Excerpts from Deployments

At the introduction all domain experts expressed enthusiasm towards the Scribble Query and the accompanying multivariate visualization. This was especially true for the orderly manager who quickly adopted the tool, and Scribble Query in particular, into organization-political responsibilities. During the deployment he would bring live versions of the interface to meetings with the hospital manager as well as employees from hospital departments to argue for man-power allocations and create Scribble Query live in order to communicate both his findings and the analysis process leading to the findings.

We also conducted interviews with the domain experts in early phases of the deployments. During these interviews, two participants found that they found the Scribble Query easy to decode and understand and that this supported them in performing iterated interaction cycles. This suggests that Scribble Query indeed helps users to bridge the gulfs of evaluation and execution (Norman et al. 1986).

The last lesson we will report on is the surprising observation that none of the participants used more than one finger to query the visualization with Scribble Query. This is despite that our implementation of Scribble Query is technically capable of handling multiple concurrent Scribble Query strokes being created, and that it was explicitly introduced to the domain experts. We speculate that, although multiple simultaneous scribbling interactions are possible, this is due to the fact that advanced queries can be scribbled with a single finger touch-input. Put differently, multiple concurrent touches are not needed to query data with Scribble Query.

However, to evaluate or conclude on our speculation it would require a larger evaluation with more domain experts, which is outside the scope of this paper.

DISCUSSION

The excerpts from our deployment of Scribble Query with domain experts from two organizations indicates that freehand scribbling of queries imposes a straight-forward conceptual model of interaction on the user, which can assist in adoption of the interaction technique. This gives Scribble Query the advantage that it is able to leverage the intuitive and familiar interaction of touch-screen interaction. Furthermore, statements from participants suggest that Scribble Query assists users in bridging the gulfs of evaluation and execution.

Deploying Scribble Query with domain experts in organizations has limited the scope of our study of Scribble Query. Regardless, we still decided to deploy Scribble Query with domain experts because we wanted to investigate the feasibility of the Scribble Query interaction technique for conducting data analysis in real-world settings by professional who need to conduct data analysts as a part of their job function. This approach did, however, provide insights to suggest that Scribble Query assisted the gulfs of evaluation and execution (Norman et al. 1986) in real-world usage and settings. Another approach, which could have enabled us to reach firm quantitative conclusions, would be to perform a strictly controlled study with a significant number of participants on a generic dataset that can be analysed by a wider range of participants.

Nonetheless, we are confident that the Scribble Query interaction technique is indeed capable of facilitating easy access to complex visual analysis of multivariate data. We also believe that Scribble Query is highly promising for use by non-programmer domain experts, who are not trained data analysts. We also believe that Scribble Query is also usable in non-professional settings, enabling a wider range of users— beyond data scientists—to conduct data analysis. Furthermore, we are confident that Scribble Query is applicable to a wider range of data types than tested here. In Figure 6 we have applied it to a mock-up of a visualization of time-series data. More advanced time queries, on par with the queries made available by Timeboxes (Hochheiser et al. 2004), could be added although this could go against the simplified conceptual model of interaction that Scribble Query embodies.

Naturally, our Scribble Query technique has several limitations and weaknesses. One of the challenges of touch interaction is that many of the more advanced interactions are based on composite gestures—such as specific finger postures, kinetic movement, or multiple consecutive actions— that are not easily discoverable and require training. For example, many smartphones support zooming by double tapping on text, an operation that a user may only hope to stumble upon if they are not explicitly taught to use it. The Scribble Query method represents another form of specialized touch gesture that may require (possibly recurring) training. In fact, this is also an argument in favour of using standardized touch gestures, even for visualization. Similarly, additional limitations include low touch precision as well as finger occlusion, both exacerbated by the high-density visual displays typically used in visualization. Future work will have to explore the trade-off between the extra strength provided by our customized gestures and their inherent weaknesses.

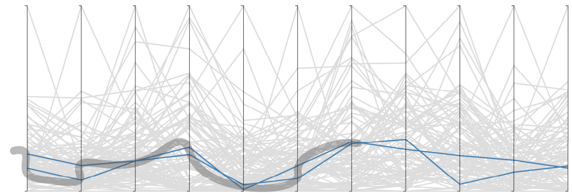

Figure 6. A mock-up of Sketch Query applied to a visualization of time-series data.

CONCLUSION AND FUTURE WORK

We have presented Scribble Query, a new family of direct-touch query mechanisms for interactive data visualization that was designed based on users' familiarity with interaction with touch sketching and drawing applications for multi-touch devices. Unlike existing interaction techniques for touch-enabled visualization, Scribble Query is not constrained to standardized touch gestures, but allow the user to fluently "scribble" their data queries using a single or multiple touch points. Our deployment study indicates propensity towards the new technique for our participants. We also provide an implementation of the technique for multivariate data in parallel coordinate plots and proved an example of how to apply it to time-series data in a line graph.

While much work on touch-enabled visualization has focused on mitigating the drawbacks of touch surfaces—such as occlusion, parallax, and low accuracy—there is comparably little work on taking full advantage of the unique strengths of the input medium for data visualization. We claim that novel Scribble Query technique has shown promise in utilizing touch-devices to provide intuitive and advanced data analysis for non-programmer domain experts. We believe that this is only a first step towards creating a large toolbox of novel analysis tools, harnessing touch in the future: to touch data, as it were. Finally, we have shown that such tools can be provided as cloud services because we utilize standard web technologies to provide the interaction and its coupled interface. This will allow a much larger user population of domain experts to enter the scene of big data analytics.

ACKNOWLEDGEMENTS

The work was supported by Danish DSF grant no. 11-115331.

We would like to thank our colleagues and collaborators from the Human-Computer Interaction Lab at University of Maryland, the Department of Computer Science at Aarhus University, and the Alexandra Institute for collaboration and feedback.

Specifically, we would like to thank for support received from the Aarhus University interdisciplinary centre PIT, the Founddata project at Aarhus University, and the EcoSense project at Aarhus University.

REFERENCES

Apitz, G., F. c. Guimbreti\`ere and S. Zhai. "Foundations for Designing and Evaluating User Interfaces Based on the Crossing Paradigm." ACM Transactions on Computer-Human Interaction 17(2) (2008): 9:1-9:42, 10.1145/1746259.1746263.

Baur, D., B. Lee and S. Carpendale. TouchWave: kinetic multi-touch manipulation for hierarchical stacked graphs. Proceedings of the ACM Conference on Interactive Tabletops and Surfaces(2012): 255-264, 10.1145/2396636.2396675.

Beaudouin-Lafon, M. Instrumental interaction: an interaction model for designing post-WIMP user interfaces. Proceedings of the ACM Conference on Human Factors in Computing Systems(2000): 446-453, 10.1145/332040.332473.

Becker, R. A. and W. S. Cleveland. "Brushing Scatterplots." Technometrics 29(2) (1987): 127-142, 10.1080/00401706.1987.10488204.

Bostock, M., V. Ogievetsky and J. Heer. "D3: Data-Driven Documents." IEEE Transactions on Visualization and Computer Graphics 17(12) (2011): 2301-2309, 10.1109/TVCG.2011.185.

Browne, J., B. Lee, S. Carpendale, N. Riche and T. Sherwood. Data analysis on interactive whiteboards through sketch-based interaction. Proceedings of the ACM Conference on Interactive Tabletops and Surfaces(2011): 154-157, 10.1145/2076354.2076383.

Chao, W. O., T. Munzner and M. V. D. Panne. NapkinVis: Rapid Pen-Centric Authoring of Improvisational Visualizations. Poster Proceedings of the IEEE Conference on Information Visualization(2009).

Drucker, S. M., D. Fisher, R. Sadana and J. Herron. TouchViz: a case study comparing two interfaces for data analytics on tablets. Proceedings of the ACM Conference on Human Factors in Computing Systems(2013): 2301-2310, 10.1145/2470654.2481318.

Dunne, C., N. H. Riche, B. Lee, R. Metoyer and G. Robertson. GraphTrail: analyzing large multivariate, heterogeneous networks while supporting exploration history, Proceedings of the ACM Conference on Human Factors in Computing Systems(2012): 1663-1672, 10.1145/2470654.2481318.

Eichmann, P. and E. Zgraggen. Evaluating Subjective Accuracy in Time Series Pattern-Matching Using Human-Annotated Rankings. Proceedings of the ACM Conference on Intelligent User Interfaces(2015): 28-37, 10.1145/2678025.2701379.

Elmqvist, N., P. Dragicevic and J.-D. Fekete. "Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation." IEEE Transactions on Visualization and Computer Graphics 14(6) (2008): 1539-1148, 10.1109/TVCG.2008.153.

Elmqvist, N., A. V. Moere, H.-C. Jetter, D. Cernea, H. Reiterer and T. J. Jankun-Kelly. "Fluid interaction for information visualization." Information Visualization 10(4) (2011), 10.1177/1473871611413180.

Elmqvist, N., J. Stasko and P. Tsigas. DataMeadow: A Visual Canvas for Analysis of Large-Scale Multivariate Data. Proceedings of the IEEE Symposium on Visual Analytics Science and Technology(2007): 187-194, 10.1057/palgrave.ivs.9500170.

eMarketer. "2 Billion Consumers Worldwide to Get Smart(phones) by 2016." Retrieved June. (2014), 2016, from http://www.emarketer.com/Article/2-Billion-Consumers-Worldwide-Smartphones-by-2016/1011694.

Harrison, C., R. Xiao, J. Schwarz and S. E. Hudson. TouchTools: leveraging familiarity and skill with physical tools to augment touch interaction. Proceedings of the ACM Conference on Human Factors in Computing Systems(2014): 2913-2916, 10.1145/2556288.2557012.

Hochheiser, H. and B. Shneiderman. "Dynamic query tools for time series data sets: timebox widgets for interactive exploration." Information Visualization 3(1) (2004): 1-18, 10.1057/palgrave.ivs.9500061.

Holz, C. and S. Feiner. Relaxed selection techniques for querying time-series graphs. Proceedings of the ACM Symposium on User Interface Software and Technology(2009): 213-222, 10.1145/1622176.1622217.

Inselberg, A. Multidimensional detective. Proceedings of the IEEE Symposium on Information Visualization(1997): 100-107, 10.1109/INFVIS.1997.636793.

Inselberg, A. and B. Dimsdale. Parallel Coordinates. Human-Machine Interactive Systems. A. Klinger. Boston, MA, Springer US(1991): 199-233 10.1007/978-1-4684-5883-1_9.

Jansen, Y. and P. Dragicevic. "An Interaction Model for Visualizations Beyond The Desktop." IEEE Transactions on Visualization and Computer Graphics 19(12) (2013): 2396-2405, 10.1109/TVCG.2013.134.

Javed, W. and N. Elmqvist. "ExPlates: Spatializing Interactive Analysis to Scaffold Visual Exploration." Computer Graphics Forum 32(3pt4) (2013): 441-450, 10.1111/cgf.12131.

Kondo, B. and C. Collins. "DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation." IEEE Transactions on Visualization and Computer Graphics 20(12) (2014): 2003-2012, 10.1109/TVCG.2014.2346250.

Kosara, R. Poster: Indirect Multi-Touch Interaction for Brushing in Parallel Coordinates. Poster Proceedings of the IEEE Conference on Information Visualization(2010), 10.1117/12.872645.

Kosara, R. Indirect multi-touch interaction for brushing in parallel coordinates. Proceedings of IS&T/SPIE Electronic Imaging(2011): 7, 10.1117/12.872645.

Kosara, R., F. Bendix and H. Hauser. "Parallel sets: Interactive exploration and visual analysis of categorical data." IEEE Transactions on Visualization and Computer Graphics 12(4) (2006): 558-568, 10.1109/TVCG.2006.76.

Lee, B., 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) (2012): 2689-2698, 10.1109/TVCG.2012.204.

Lee, B., R. H. Kazi and G. Smith. "SketchStory: Telling More Engaging Stories with Data through Freeform Sketching." Visualization and Computer Graphics, IEEE Transactions on 19(12) (2013): 2416-2425, 10.1109/TVCG.2013.191.

Lee, B., G. Smith, N. H. Riche, A. Karlson and S. Carpendale. SketchInsight: Natural data exploration on interactive whiteboards leveraging pen and touch interaction. Proceedings of the IEEE Pacific Visualization Symposium(2015): 199-206, 10.1109/PACIFICVIS.2015.7156378.

Nebeling, M. and M. Norrie. jQMultiTouch: lightweight toolkit and development framework for multi-touch/multi-device web interfaces. Proceedings of the ACM Symposium on Engineering Interactive Computing Systems(2012): 61-70, 10.1145/2305484.2305497.

Nguyen, H. and M. C. Bartha. "Shape Writing on Tablets: Better Performance or Better Experience?" Proceedings of the Human Factors and Ergonomics Society Annual Meeting 56(1) (2012): 1591-1593, 10.1177/1071181312561317.

Nielsen, M. and K. Grønbæk. PivotViz: Interactive Visual Analysis of Multidimensional Library Transaction Data. Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries. Knoxville, Tennessee, USA, ACM(2015): 139-142 10.1145/2756406.2756937.

Nielsen, M., M. B. Kjaergaard and K. Gronbaek. Exploring Interaction Techniques and Task Types for Direct-Touch as Input Modality. Poster Proceedings of the IEEE Conference on Information Visualization(2013).

Norman, D. A. and S. W. Draper. User Centered System Design; New Perspectives on Human-Computer Interaction, Lawrence Erlbaum Associates Inc. (1986).

Perin, C., P. Dragicevic and J.-D. Fekete. Crossets: Manipulating Multiple Sliders by Crossing. Proceedings of the Graphics Interface Conference(2015): 233-240.

Roberts, J. C. State of the art: Coordinated & multiple views in exploratory visualization. Proceedings of the International Conference on Coordinated and Multiple Views in Exploratory Visualization(2007): 61-71, 10.1109/CMV.2007.20.

Ryall, K., N. Lesh, T. Lanning, D. Leigh, H. Miyashita and S. Makino. QueryLines: approximate query for visual browsing. Proceedings of the ACM Conference on Human Factors in Computing Systems(2005): 1765-1768, 10.1145/1056808.1057017

Rzeszotarski, J. M. and A. Kittur. Kinetica: Naturalistic multi-touch data visualization. Proceedings of the ACM Conference on Human Factors in Computing Systems(2014): 897-906, 10.1145/2556288.2557231.

Sadana, R. and J. Stasko. Designing and Implementing an Interactive Scatterplot Visualization for a Tablet Computer. Proceedings of the ACM Conference on Advanced Visual Interfaces(2014): 265-272, 10.1145/2598153.2598163.

Schmidt, S., M. A. Nacenta, R. Dachselt and S. Carpendale. A set of multi-touch graph interaction techniques. Proceedings of the ACM Conference on Interactive Tabletops and Surfaces(2010): 113-116, 10.1145/1936652.1936673

Shneiderman, B. "Direct Manipulation: A Step Beyond Programming Languages." Computer 16(8) (1983): 57-69, 10.1109/MC.1983.1654471.

Theron, R. Visual Analytics of Paleoceanographic Conditions. Proceedings of the IEEE Symposium on Visual Analytics Science And Technology(2006): 19-26, 10.1109/VAST.2006.261452

Tsandilas, T., A. Bezerianos and T. Jacob. SketchSliders: Sketching Widgets for Visual Exploration on Wall Displays. Proceedings of the ACM Conference on Human Factors in Computing Systems(2015): 3255-3264, 10.1145/2702123.2702129.

Tufte, E. R. Envisioning information, Graphics Press (1991).

Vlaming, L., C. Collins, M. Hancock, M. Nacenta, T. Isenberg and S. Carpendale. Integrating 2D mouse emulation with 3D manipulation for visualizations on a multi-touch table. Proceedings of the ACM Conference on Interactive Tabletops and Surfaces(2010): 221-230, 10.1145/1936652.1936693

Wang, F. and X. Ren. Empirical evaluation for finger input properties in multi-touch interaction. Proceedings of the ACM Conference on Human Factors in Computing Systems(2009): 1063-1072, 10.1145/1518701.1518864

Wattenberg, M. Sketching a graph to query a time-series database. Extended Abstracts of the ACM Conference on Human Factors in Computing Systems(2001): 381-382, 10.1145/634067.634292